

Université Abou Bakr Belkaid– Tlemcen

Faculté des Sciences

Département d'Informatique



Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option : Génie logiciel (G.L)

Thème

Conception et réalisation d'une plateforme numérique pour le tourisme

Réalisé par :

- Ali Chaouch Sid Ahmed
- Benhammou Mohammed Rachid

Présenté le 04 Juillet 2022 devant le jury composé de :

- Mr. Mahfoud Houari (Président)
- Mr. Brikci Nigassa Amine (Examineur)
- Mr. Moussaoui Djilali (Encadrant)
- Mr. Tadlaoui Mohamed (Encadrant)

Année universitaire : 2021-2022

Remerciements

Au nom d'Allah le Miséricordieux le Très Miséricordieux. Certes, la louange est à Allah, De qui nous implorons aide et repentance.

Nous remercions Allah le tout puissant de nous avoir guider et aider à la réussite de ce projet avec courage et patience.

En préambule à ce mémoire, nous souhaitons adresser, nos remerciements les Sincères aux personnes qui nous ont apporté leur aide et qui ont attribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Nous tenons à remercier sincèrement Nos encadrants M. Moussaoui Djilali et M. Tadlaoui Mohamed, pour leurs contributions, pour le temps précieux qu'ils nous ont accordé, et pour leur soutien tout au long du projet.

Nous tenons à saluer les membres du jury Mr.Mahfoud Houari et Mr. Brikci Nigassa Amine qui nous ont honorés par leur présence et d'avoir accepté d'évaluer notre travail.

Nous n'oublierons surtout pas nos chers parents pour leurs contributions, leur soutien et leur patience.

À tous les enseignants du département informatique et surtout nos enseignants qui par leur engagement scientifique et éducatif, durant ces cinq années d'études, ont été pour nous une source d'inspiration.

Merci à tous et à toutes.

Dédicaces

Nous dédions ce modeste travail et notre profonde gratitude :

À Nos très chers parents qui nous ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, nous ne saurions exprimer notre gratitude seulement par des mots. Que dieu vous protège et vous garde pour nous.

À nos chers frères et nos précieuses sœurs, les mots ne peuvent résumer notre reconnaissance et notre amour à votre égard.

À tous les membres de nos familles.

À nos adorables amis, pour votre fidélité et votre soutien.

À tous nos amis avec lesquels nous avons partagé nos moments de joie et de bonheur.

À tous nos enseignants pour votre soutien, votre enseignement et vos conseils tout au long de notre parcours éducatif et professionnel.

Que toute personne nous ayant aidé de près ou de loin, trouve ici l'expression de notre reconnaissance.

TABLE DES MATIERES

Introduction générale	9
Contexte	9
Problématique	9
Solutions proposées	10
1. PLANIFICATION ET SPECIFICATION DES EXIGENCES	11
1.1. Introduction	12
1.2. Estimation Initiale (Retro Planning)	12
1.3. Spécification des exigences.....	14
1.3.1. Besoins Fonctionnels	14
1.3.2. Besoins non fonctionnels	15
1.3.3. Cas d'utilisations.....	16
1.3.4. Diagrammes de séquence.....	21
1.3.5. Maquettes	28
1.3.6. Animations.....	29
1.3.7. Palettes de couleurs.....	31
1.3.8. Logo	31
1.3.9. Police du texte.....	31
1.3.10. Utilisation.....	32
1.3.11. Boutons	32
1.4. Systèmes similaires	33
1.5. Enchaînement des écrans	36
1.6. Conclusion.....	37
2. CONCEPTION	38
2.1. Introduction	39
2.2. Architecture de l'application.....	39
2.3. Patrons de conceptions	40
2.3.1. Patron Modèle-vue-contrôleur	40
2.3.2. Les effets du MVC sur un projet	41
2.3.3. Projection	41
2.3.4. Authentification (exemple).....	42
2.3.5. Ajout d'une place touristique (exemple)	43

2.4.	Modèle de données.....	44
2.4.1.	Modèle entité relation (MCD)	45
2.4.2.	Modèles logiques des données (MLD)	45
2.5.	Diagramme de classe.....	46
2.6.	Génération de chemin	47
2.7.	Sécurité du système.....	50
2.7.1.	Sécuriser le code	50
2.7.2.	Utilisations d'APIs autorisées.....	50
2.7.3.	Le principe du moindre privilège.....	50
2.7.4.	Communications via les protocoles sécurisés.....	51
2.7.5.	Tester à plusieurs reprises	51
2.8.	Conclusion.....	51
3.	REALISATION ET DEVELOPPEMENT	52
3.1.	Introduction	53
3.2.	Interfaces finales	53
3.3.	Outils de conception UML.....	57
3.4.	Outils de conception graphique.....	57
3.5.	Environnement de développement.....	57
3.5.1.	Protocoles, marquages & langages de programmation.....	57
3.5.2.	Outils de développement	62
3.5.3.	Fournisseurs de serveurs et d'hébergement	64
3.5.4.	Modèles de pages web	65
3.6.	Packages de flutter	66
3.7.	Services web.....	68
3.8.	Déploiement d'application web sur Serveur	71
3.8.1.	Prérequis	71
3.8.2.	Etapes d'installation.....	71
3.9.	Conclusion.....	74
	Conclusion Générale.....	75
	Bibliographie	77
	Résumé.....	78
	ملخص.....	78
	Abstract.....	78

LISTE DE FIGURE

Figure 1: planification du mois Avril	12
Figure 2: Planification du mois Mai-Juin	13
Figure 3: Diagramme de cas d'utilisation général.....	19
Figure 4: Diagramme de séquence "Générer un chemin".....	23
Figure 5: Diagramme de cas d'utilisation "Gérer les sites touristiques".....	26
Figure 6: Diagramme de séquence "Consulter les sites touristiques".....	27
Figure 7: Maquettes d'application mobile.....	28
Figure 8: Spinkit affichage	29
Figure 9: Modal Progress Hud.....	30
Figure 10: Animated Text Kit.....	30
Figure 11: Palette couleur de l'interface	31
Figure 12: Logo d'application.....	31
Figure 13: Fonts.....	32
Figure 14: UI buttons.....	32
Figure 15 : Interface de l'application IDVizit.....	33
Figure 16: Interface de l'application Culture Trip.....	34
Figure 17: Enchaînement des écrans de l'application.....	36
Figure 18: Architecture de l'application	39
Figure 19: Schéma d'interactions MVC [13].....	41
Figure 20:Interface d'identification : front-office a gauche - back-office a droite	43
Figure 21:Schéma de l'action d'authentification	43
Figure 22:Vue de la liste des places et le formulaire d'ajout	44
Figure 23: Schéma de l'action d'ajout de places	44
Figure 24: Diagramme entité relation.....	45
Figure 25: Diagramme de classe.....	47
Figure 26:Organigramme de la méthode "Générer un chemin"	49
Figure 27: Prototypes des interfaces finales pour l'administrateur (Web).....	54
Figure 28: Prototype des interfaces finales pour les touristes (Web).....	55
Figure 29: Prototype des interfaces finales (Mobile).....	56
Figure 30: Graphes et statistique de Flutter et React Native	61
Figure 31 Graphe de contribution au git.....	64
Figure 32: Modèle WonderTour	65

Figure 33: Argon Dashboard	65
Figure 34:Google Maps API distance exemple	70

LISTE DE TABLEAU

Tableau 1 : Liste des besoins fonctionnels	14
Tableau 2: Cas d'utilisation associé au système.....	17
Tableau 3: Description du cas d'utilisation "Ajouter place touristique"	19
Tableau 4 : Description du cas d'utilisation Enregistrer un chemin	20
Tableau 5: Description du cas d'utilisation Consulter un site touristique.....	21
Tableau 6: Comparaison de la concurrence	35
Tableau 7:Différence entre Flutter & React Native[14]	59

LISTE D'ABREVIATION

API	Application Programming Interface.
APK	Android Package.
BDD	Base de données.
CSS	Cascading Style Sheets - Feuilles de style en cascade.
HTTP	Hyper Text Transfer Protocol.
HTTPS	Hyper Text Transfer Protocol Secure.
IDE	Integrated Development Environnement.
IHM	Interface Homme Machine.
IOS	Operating System.
JSON	(JavaScript Object Notation) est un format léger d'échange de données

MVC	Modele Vue Controlleur.
REST	Representational state transfer.
SDK	Software Developpment Kit.
UI	Interface de l'utilisateur.
UML	Unified Modeling Language.

Introduction générale

Contexte

Dans un monde moderne où la façon de penser et de vivre a été révolutionné par l'invention de l'internet, les tâches quotidiennes qu'une personne accomplit deviennent faciles et ne nécessitent pas beaucoup de temps ni de peine de mouvement. Après l'an 2000, les appareils mobiles connectés sont apparus capables d'envahir la vie humaine. La personne peut accéder à toutes les informations dont elle a besoin partout où il y a une couverture réseau. Ceci nous a poussé et motivé à développer une application mobile.

Sous la pression du travail et l'atmosphère de tension, toutes les personnes cherchent à profiter d'une période de leurs vacances dans le tourisme et l'aventure, vous les trouvez donc voyager dans de nouvelles régions à la recherche de repos et de loisirs.

La culture du voyage s'est généralisée dans toutes les couches de la société. Chaque année, de nombreuses familles et jeunes se rendent dans les villes et les zones touristiques pour découvrir et s'imprégner de leur histoire et de leur culture, ce qui les aide à élargir leur réflexion, à augmenter leur expérience et à leur donner la tranquillité d'esprit. Les zones touristiques peuvent être classées en fonction de leur histoire et des significations qu'elles symbolisent, et c'est ce qui aide les touristes à acquérir la destination qu'ils souhaitent visiter.

Problématique

Cependant, ces voyages imposent aux voyageurs certains obstacles, notamment la méconnaissance des lieux touristiques et l'incapacité de connaître leur histoire, le prix d'entrée, le temps nécessaire pour les visiter, le chemin qui y mène, la distance, la durée et d'autres informations. Aussi, il est difficile de choisir les zones que nous voulons visiter, car il y a ceux qui veulent visiter les monuments anciens, et il y a ceux qui veulent visiter les lieux de culte, les parcs d'attractions et beaucoup d'autres types de site touristique, donc comment pouvons-nous organiser notre temps pour visiter les endroits que nous voulons ?

Le service de tourisme a manifesté des difficultés dans la gestion des sites touristiques, et les touristes rencontrent des obstacles et problèmes divers.

L'informatique est devenue indispensable au niveau de tous les domaines dans notre vie courante, pour cela il est utile de réaliser une application pour la gestion des sites touristiques. Mais peut-on vraiment intégrer la technologie et l'informatique à ce domaine ? Existe-t-il un moyen afin de réaliser une application mobile (avec un site web) pour répondre à ces besoins en respectent les termes d'ergonomies et les aspects fonctionnels ? Si oui, quelles sont les méthodes et les processus de développement suivis pour accomplir un tel système ?

Solutions proposées

À travers les réunions dans le but d'expliquer le thème choisi et clarifier ses besoins, nous avons pu comprendre le fonctionnement de ce service, ce qui nous a permis après une analyse et étude préalable d'adopter la proposition suivante qui pourrait bien répondre aux questions posées précédemment comme solution :

- Conception et réalisation d'un système pour la gestion des sites touristiques qui se compose de deux applications, la première est une application mobile pour les touristes, et la deuxième est une application web qui est elle-même divisée en 2 espaces (un espace pour les touristes et un espace pour l'administrateur).

Dans le premier chapitre, nous allons présenter une introduction générale en faisant un aperçu rapide sur l'environnement (scope) de notre thème. Puis nous définissons la méthodologie à suivre afin de réaliser notre système.

Ensuite, dans le second, nous allons introduire la planification tout au long du projet, une analyse des besoins et une étude des systèmes similaires dont nous définissons la différence des critères entre eux, les points forts et les points faibles de chaque système et les critiques.

Le troisième chapitre, focalise sur les méthodes utilisées et les étapes suivis pour réaliser ce projet. Ce chapitre inclut les phases nécessaires tels que la conception du système schématisées par l'architecture logicielle, les patrons de conceptions et le diagramme de classe qui sont bien détaillés.

Enfin, le dernier chapitre, dédié à la présentation d'environnement de développement, les outils et langages de programmations utilisés dans notre travail. Et nous concluons par une conclusion générale et quelques prescriptives.

1.

**PLANIFICATION
ET SPECIFICATION DES
EXIGENCES**

1.1. Introduction

Nous commençons dans ce chapitre par la planification du processus de conception et la création de l'application. Puis, il est intéressant d'étudier le marché et d'analyser quelques applications qui ont un but similaire à celui de notre produit final. Ensuite, nous allons présenter l'enchaînement des écrans pour les interfaces web et mobiles de l'application. Par la suite, nous allons identifier les exigences fonctionnelles et non-fonctionnelles accompagnées d'une analyse conceptuelle et des diagrammes de séquence. Enfin, nous terminons par la charte graphique de la plateforme et une conclusion.

1.2. Estimation Initiale (Retro Planning)

Pour organiser notre temps de travail et mieux gérer notre projet, nous avons utilisé l'outil de diagramme de Gantt pour visualiser l'avancement des différentes tâches dans le temps afin d'éviter de dépasser les limites convenues au départ (mars 2022).

Tâche à accomplir	Responsable(s)	Etat	avr-22			
Création des interfaces de l'utilisateur			Semaine 1	Semaine 2	Semaine 3	Semaine 4
L'étude des besoins	Ali Chaouch, Benhammou	Terminé				
Etude du marché	Ali Chaouch, Benhammou	Terminé				
MockUp application	Ali Chaouch, Benhammou	Terminé				
Création des interfaces Web	Ali Chaouch	Terminé				
Réunion et changements	Ali Chaouch, Benhammou	Terminé				
Apprendre et se familiariser avec Flutter	Benhammou	En cours				
Mise en œuvre des prototypes finaux Mobile	Benhammou	Terminé				
Mise en œuvre des prototypes finaux Web	Ali Chaouch	Terminé				
Réunion et prise de décisions finales	Ali Chaouch, Benhammou	Terminé				
Conception			Semaine 1	Semaine 2	Semaine 3	Semaine 4
Conception des cas d'utilisation	Ali Chaouch, Benhammou	Terminé				
Conception des diagrammes (séquences, class)	Ali Chaouch, Benhammou	Terminé				
Réunion et correction	Ali Chaouch, Benhammou	Terminé				
Conception et correction modèle relationnel	Ali Chaouch, Benhammou	En cours				
Installation et maintenance des logiciels	Ali Chaouch, Benhammou	En cours				
Création des interfaces mobiles	Benhammou	A faire				
Création des interfaces Web	Ali Chaouch	A faire				
Mise en forme de la BDD et liens	Ali Chaouch	A faire				

Figure 1: planification du mois Avril

Tâche à accomplir	Responsable(s)	Etat	Mai - Juin 2022				
			Semaine 1	Semaine 2	Semaine 3	Semaine 4	Semaine 5
Réalisation							
Ajout des interfaces et fonctions pour l'administrateur sur Web (CRUD: catégories, villes, lieux)	Ali Chaouch	A faire	■				
Création des interfaces&API Login/Register	Ali Chaouch, Benhammou	A faire	■				
Création des interfaces&API des villes et places	Ali Chaouch	A faire	■				
Débogage et tests des fonctions ajoutées	Ali Chaouch, Benhammou	A faire		■			
Ajout de l'aspect multi-lingues à l'application Mobile	Benhammou	A faire		■			
Ajout de la fonctionnalité consulter site touristique/chemin sur carte/médias	Ali Chaouch, Benhammou	A faire		■			
Ajouter la fonctionnalité modifier le profil	Ali Chaouch	A faire		■			
Débogage et tests des fonctions ajoutées	Ali Chaouch, Benhammou	A faire		■	■		
Ajout de l'aspect multi-lingues à l'application Web	Ali Chaouch	A faire			■		
Ajouter text to speech dans l'application	Benhammou	A faire			■		
Ajouter la fonctionnalité&API créer circuit	Ali Chaouch, Benhammou	A faire			■		
Ajouter la fonctionnalité&API lieux à proximité	Ali Chaouch, Benhammou	A faire			■		
Ajouter la fonctionnalité&API liste des lieux préférés	Ali Chaouch, Benhammou	A faire			■		
Débogage et tests des fonctions ajoutées	Ali Chaouch, Benhammou	A faire			■	■	
Test user final et finitions	Ali Chaouch, Benhammou	A faire				■	■
Déploiement	Ali Chaouch, Benhammou	A faire					■

Figure 2: Planification du mois Mai-Juin

Ce planning s'est fait en Mars 2022 sur le logiciel Excel.

À la fin de chaque semaine, les membres de l'équipe sont appelés pour une réunion afin de fournir des démonstrations et des critiques constructives. Aussi, après avoir été satisfaits par les résultats attendus, d'autres tests sont effectués auprès des utilisateurs finaux autour de nous, et une intégration continue sera nécessaire pour éviter tout conflit de code.

1.3. Spécification des exigences

La spécification des exigences est la première phase du cycle de vie d'une application. Cette étape a pour but de décrire l'application à développer. Après avoir dressé une vague liste de besoins, nous avons identifié les besoins prioritaires et obligatoires dont nous avons besoin pour assurer son bon fonctionnement.

1.3.1. Besoins Fonctionnels

Tableau 1 : Liste des besoins fonctionnels

ID	Nom	Description	Priorité
ID1	Fonctionnalité d'inscription et Connexion	Nécessaire pour l'application afin de sauvegarder les circuits et les lieux préférés, et garder un maximum de sécurité, passe sous 2 formulaires (connecter, inscrire).	P1
ID2	Fonctionnalité d'ajouter, modifier ou supprimer une catégorie, une ville ou un site touristique	Indispensable pour l'application, sans elle, l'application n'aura pas de sens, propre à l'administrateur pour gérer la liste des villes et sites touristiques.	P1
ID3	Fonctionnalité de consulter les monuments	Un utilisateur doit avoir la possibilité de consulter un site touristique en affichant une description (texte, voix, images, vidéos).	P1
ID4	Fonctionnalité d'avoir le chemin vers le lieu à visiter	L'utilisateur doit avoir un chemin sur une carte (de sa position vers le lieu à visiter).	P1

ID5	Fonctionnalité d'avoir la liste des lieux à proximité	La liste des lieux à proximité (dans un rayon de 5 kilomètres) doit s'afficher pour l'utilisateur triées en critère de distance.	P2
ID6	Fonctionnalité de créer un circuit	Un chemin doit être proposé à l'utilisateur en respectant certains critères (catégories, durée, budget, distance).	P2
ID7	Fonctionnalité de créer une liste des lieux préférés	L'utilisateur peut avoir une liste de lieux favoris, à chaque fois qu'il aime un lieu, ce lieu est ajouté à la liste des favoris.	P3
ID8	Fonctionnalité de recherche d'un site touristique	Un utilisateur peut accéder à la recherche d'un lieu avec mots-clés qui pourrait se trouver dans le titre d'un lieu ou dans l'un de ses attributs (prix, durée).	P3
ID9	Fonctionnalité de modifier le profil	L'utilisateur peut avoir la possibilité de modifier ses coordonnées enregistrées.	P4
ID10	Fonctionnalité d'avoir une application multilingue	L'IHM des applications web et Mobile et les données doivent supporter l'aspect du multilingue.	P4

1.3.2. Besoins non fonctionnels

Ce sont les besoins qui concernent la performance de l'application, la technique respectée pour la réaliser et même les contraintes d'implémentation (langage de programmation, Serveur...). Dans ce travail, l'application devra impérativement respecter les contraintes suivantes :

- ❖ **Extensible et Evolutive** : C'est-à-dire qu'il est toujours possible d'exercer l'application dans de nouvelles régions en ajoutant ses villes et lieux touristiques.

Aussi, les fonctionnalités peuvent être modifiées et de nouvelles peuvent être ajoutées.

- ❖ **Authentification** : la partie la plus concernée par ce point est l'administrateur parce que c'est lui qui gère la base des lieux. Et si cette base n'est pas bien sécurisée alors l'application est vulnérable et ouverte aux attaquants.
- ❖ **Ergonomie** : l'interface de l'application doit être conviviale, explicite et facile à utiliser en respectant les règles des couleurs et de taille des éléments.
- ❖ **Le contrôle des champs** : l'application doit avoir un contrôleur des champs de saisie pour respecter les types des champs tels que mot de passe, email, date...
- ❖ Le site web est capable de gérer une forte charge de visiteurs sans perte de performance.
- ❖ L'application web s'exécute rapidement les différentes fonctions et ne consomme pas trop de mémoire.
- ❖ L'application web peut être exécutée sur un protocole https sécurisé sans aucune intervention.
- ❖ Enfin, l'application mobile doit avoir accès à la localisation, réseau internet pour assurer que les données sont reçues correctement à travers les APIs.

1.3.3. Cas d'utilisations

1.3.3.1. Identification des acteurs

Ci-dessous, nous énumérons les acteurs qui interagissent directement avec le système étudié avec une brève explication pour chacun :

- **Administrateur** : le rôle de l'administrateur est la gestion des villes, catégories et sites touristiques.
- **Utilisateur connecté** : toute personne s'étant inscrite à l'application, un utilisateur peut consulter les sites touristiques et les services qui leur sont liés.
- **Utilisateur non-connecté** : un individu qui accède à l'application et une partie de ses fonctionnalités sans disposer d'un compte.

1.3.3.2. Identification des cas d'utilisation

Les cas d'utilisation permettent d'exprimer les besoins des utilisateurs d'un système informatique, ils représentent un ensemble de séquences ou d'actions, et produisent des services rendus intéressants par un acteur particulier, sans toutefois préciser leurs modes de réalisation. En d'autres termes, un cas d'utilisation modélise une fonctionnalité en exprimant les interactions acteur/système et apporte une valeur ajoutée à l'acteur concerné.

Pour chaque acteur identifié précédemment, nous associons les cas d'utilisation qui lui correspondent.

Le tableau suivant répertorie les différents cas d'utilisation associés à notre système, en indiquant si ce cas est accessible via web & mobile ou web seulement :

Tableau 2: Cas d'utilisation associé au système

N°	Cas d'utilisation	Acteur
1	Gérer les villes (web)	Administrateur
2	Gérer les catégories (web)	Administrateur
3	Gérer les places touristiques (web)	Administrateur
4	Gérer la galerie d'un site touristique(web)	Administrateur
5	Gérer le profil (web)	Utilisateur connecté
6	S'inscrire (web & mobile)	Utilisateur non-connecté
7	Rechercher un lieu touristique (mobile)	Utilisateur connecté
		Utilisateur non-connecté
8	Générer les chemins (web & mobile)	Utilisateur connecté
		Utilisateur non-connecté

9	Enregistrer le chemin (web & mobile)	Utilisateur connecté
10	Consulter lieux préférés (web & mobile)	Utilisateur connecté
11	Consulter les chemins enregistrés (web & mobile)	Utilisateur connecté
12	Consulter un site touristique (web & mobile)	Tous les acteurs
13	Consulter la galerie du site touristique (web & mobile)	Tous les acteurs
14	Aimer un lieu (web & mobile)	Utilisateur connecté
15	S'authentifier (web & mobile)	Administrateur
		Utilisateur connecté

1.3.3.3. Diagrammes des cas d'utilisation et leurs descriptions

- **Diagramme de cas d'utilisation général**

Le diagramme suivant résume tous les cas d'utilisation associés à tous les acteurs de notre système :

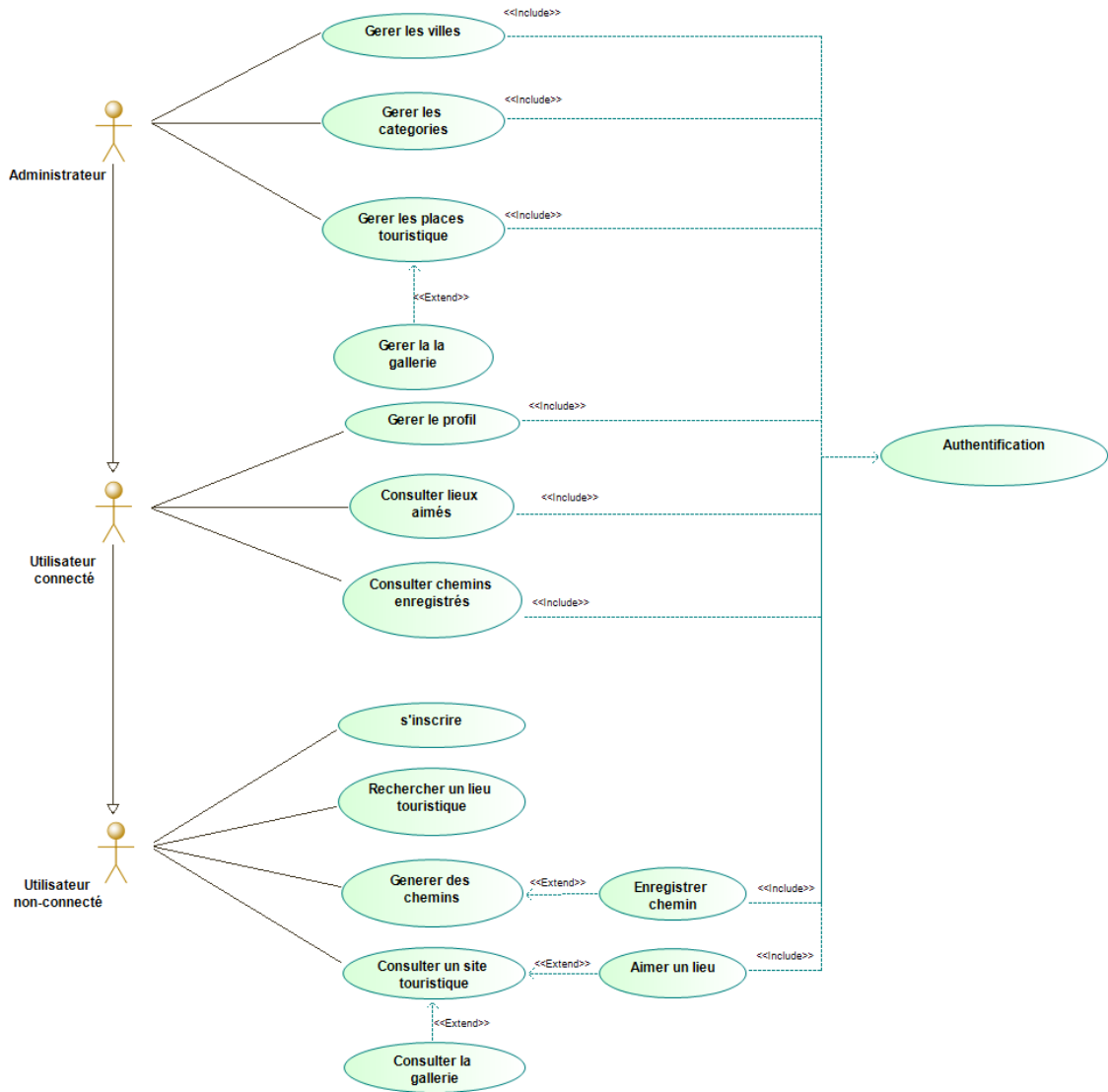


Figure 3: Diagramme de cas d'utilisation général

- Diagrammes des cas d'utilisation détaillés :

Côté administrateur : Le cas d'utilisation gérer les places touristiques :

Ci-dessous, nous décrivons le cas d'utilisation gérer les places touristiques, et plus précisément, ajouter une place touristique :

Tableau 3: Description du cas d'utilisation "Ajouter place touristique"

Cas d'utilisation	Gérer les places touristiques
Acteur	Administrateur

Objectif	Ajout/suppression et modification des sites touristiques et leurs galeries
Précondition	Authentification
Scénario nominal	<ul style="list-style-type: none"> - Le système affiche le Dashboard - L'administrateur demande l'interface des places - Le système affiche l'interface demandée - L'administrateur ajoute les données d'une place et valide - Le système enregistre les données
Alternative	Si les informations fournies sont incomplètes ou incorrectes (champ vide, incompatibilité avec le type de champ), le système réaffiche l'interface d'ajout d'article avec un message d'erreur.

Côté Utilisateur : Le cas d'utilisation Enregistrer un chemin :

Ci-dessous, nous décrivons le cas, d'utilisation Enregistrer un chemin :

Tableau 4 : Description du cas d'utilisation Enregistrer un chemin

Cas d'utilisation	Enregistrer un chemin
Acteur	Utilisateur connecté
Objectif	Enregistrer un chemin déjà créé
Précondition	Authentification, création du chemin
Scénario nominal	<ul style="list-style-type: none"> - L'utilisateur demande l'enregistrement du chemin en appuyant sur Save. - Le système stocke ce chemin à la liste des chemins sauvegardés privés pour cet utilisateur. - Le système affiche la liste des lieux de ce chemin

Alternative	Si l'opération de stockage échoue, un message d'erreur s'affiche.
--------------------	---

Côté Visiteur : Le cas d'utilisation Consulter un site touristique :

Ci-dessous, nous décrivons le cas, d'utilisation consulter un site touristique :

Tableau 5: Description du cas d'utilisation Consulter un site touristique

Cas d'utilisation	Consulter un site touristique
Acteur	Utilisateur non-connecté
Objectif	Consulter un site touristique, ses données et ses médias (photos, vidéos et sons)
Précondition	Aucune.
Scénario nominal	<ul style="list-style-type: none"> - Le visiteur demande de consulter les sites touristiques d'une ville. - Le système affiche l'interface demandée. - Le visiteur demande d'afficher les détails d'un site touristique. - Le système affiche les détails du site sélectionné.
Alternative	Aucune.

1.3.4. Diagrammes de séquence

Les diagrammes de séquence sont utilisés pour montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation. Par souci de simplification, nous représentons l'acteur principal à gauche du schéma, et les éventuels acteurs secondaires à droite du système.

Le schéma de séquence du système considère le système comme une boîte noire vue de l'extérieur (par les acteurs) sans précision sur son fonctionnement.

1.3.4.1. Diagramme de séquence du cas d'utilisation

« Générer un chemin »

Pour générer un chemin dans l'interface WEB, il faudra simplement naviguer vers la page découvrir, choisir une ville, puis un bouton de génération apparaîtra pour accéder aux paramètres du chemin, ainsi le serveur fera les calculs nécessaires et retournera les résultats. Si l'utilisateur est authentifié, il pourra alors sauvegarder le chemin généré en lui donnant un nom.

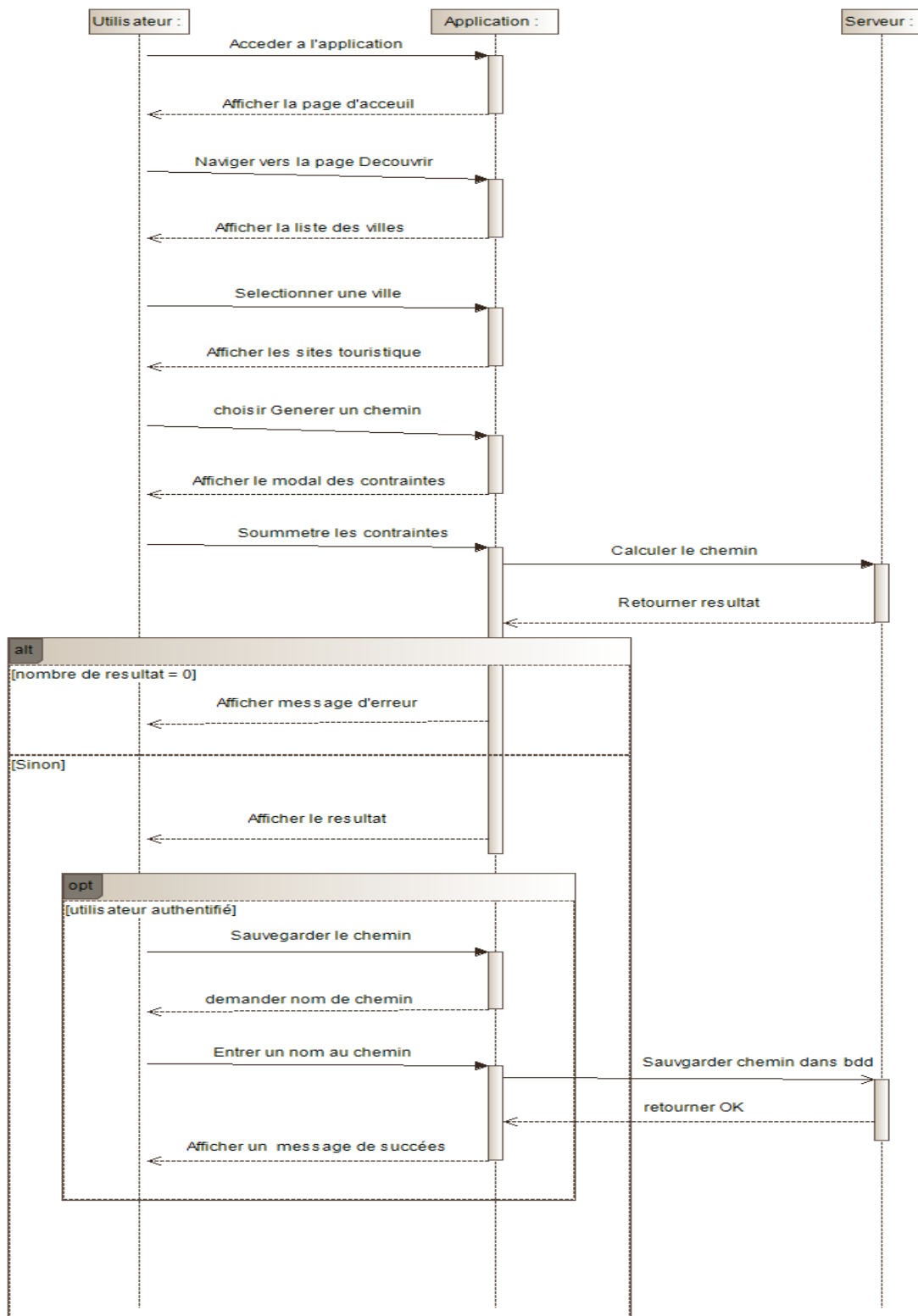
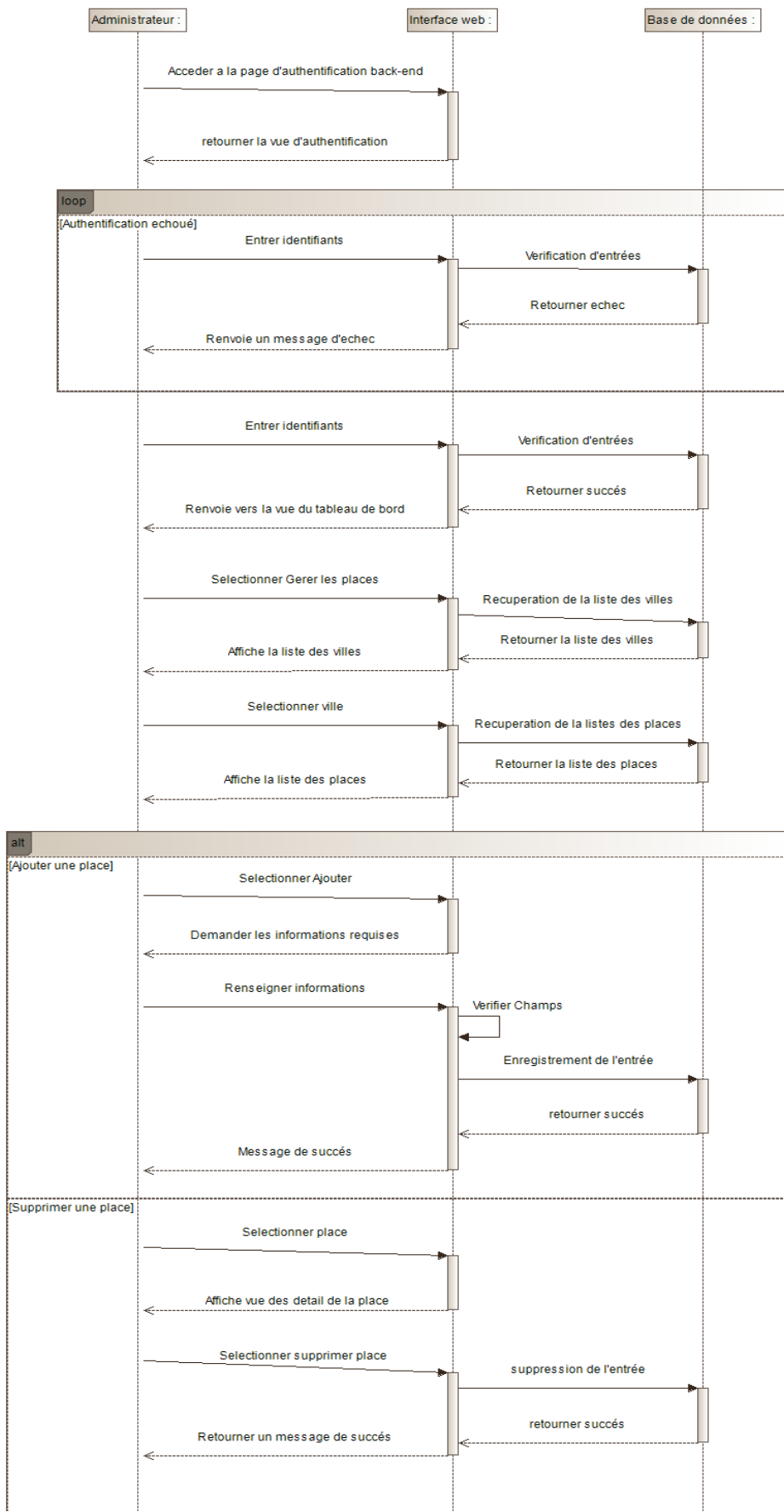


Figure 4: Diagramme de séquence "Générer un chemin"

1.3.4.2. Diagramme de séquence du cas d'utilisation « Gérer les sites touristique »

Pour gérer les sites touristiques, il faudra accéder à la page d'administration après l'authentification, une page contenant le tableau de bord sera présentée, l'administrateur devra choisir gérer les places depuis le menu. Puis, sélectionner la ville qui vas contenir les places correspondantes. Il pourra alors, ajouter une place en choisissant « Ajouter une nouvelle place ». Ensuite, un formulaire va être affiché pour entrer la catégorie et le nom de la place.

À la sélection d'une place touristique, une vue contenant un formulaire de toutes les information, l'administrateur pourra modifier et soumettre les changements, supprimer une place, ou ouvrir la page de gestion de la galerie où il peut ajouter un nouveau media, modifier le titre ou supprimer le media.



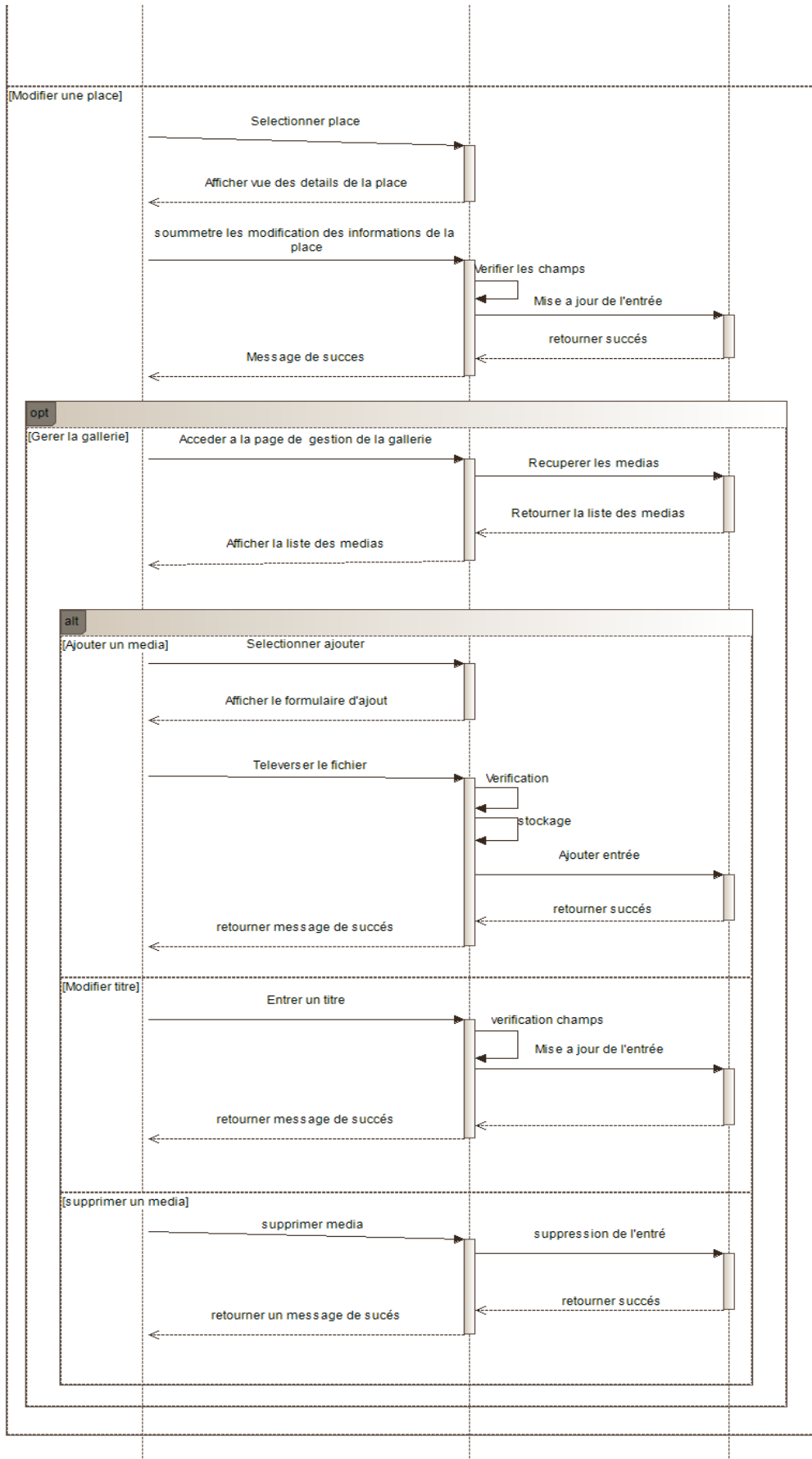


Figure 5: Diagramme de cas d'utilisation "Gérer les sites touristiques"

1.3.4.3. Diagramme de séquence du cas d'utilisation

« Consulter les sites touristiques »

Pour consulter un site touristique via l'application mobile, il faudra choisir une ville. Puis, le système va chercher les places touristiques, l'utilisateur aura l'option de trier ou chercher les sites touristiques. Dès la sélection d'une place, l'application fera appel à un service qui vas retourner les données météorologiques de ce site, puis appel au serveur pour récupérer les informations et puis les afficher à l'utilisateur, il aura la possibilité d'accéder à la galerie pour afficher les médias.

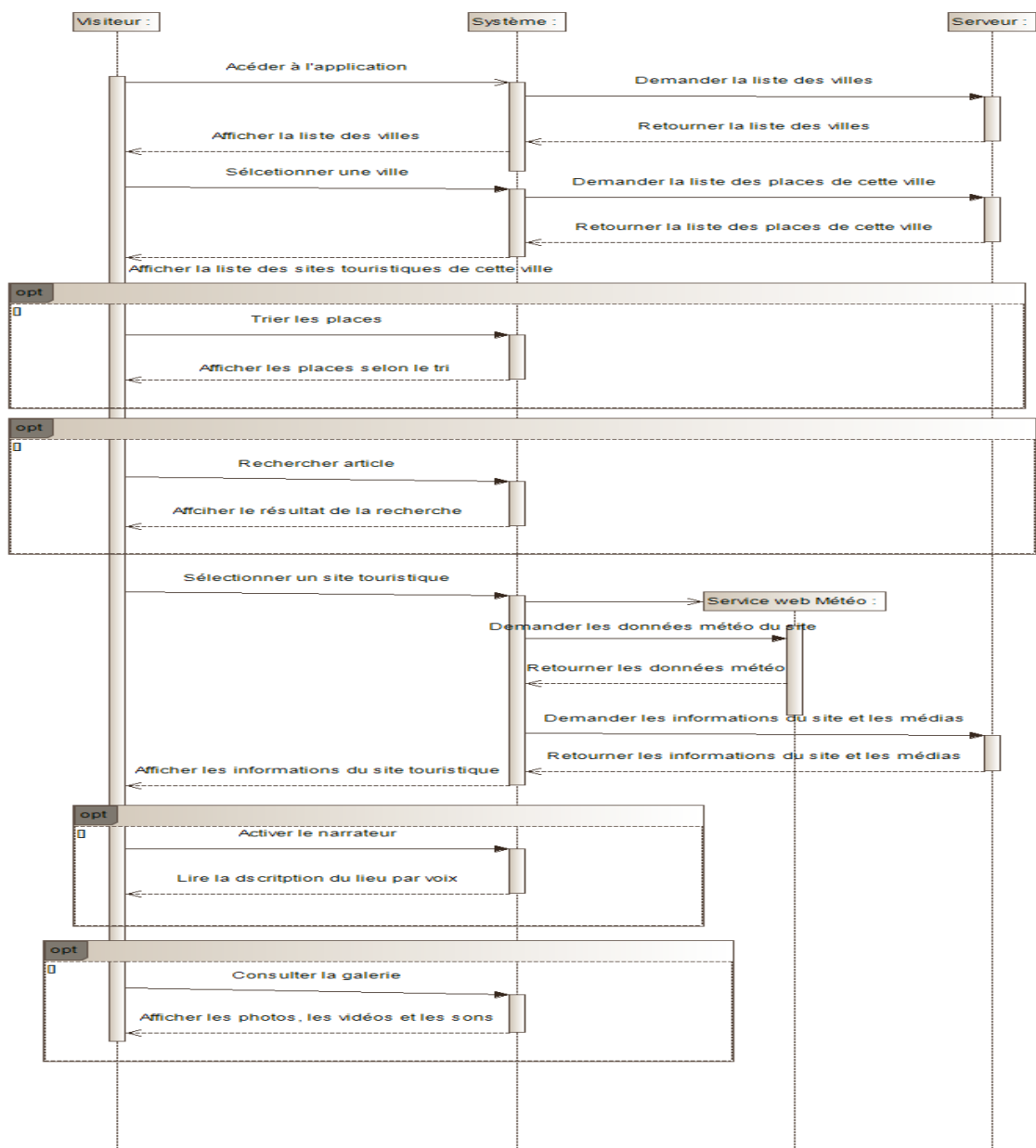


Figure 6: Diagramme de séquence "Consulter les sites touristiques"

1.3.5. Maquettes

Après avoir une bonne vision de l'application, il est possible de mettre en œuvre des interfaces. Et avec l'utilisation du Figma (éditeur des graphiques vectoriels en ligne), nous avons réalisé les interfaces suivantes, jugées les plus importantes en matière d'User Experience :

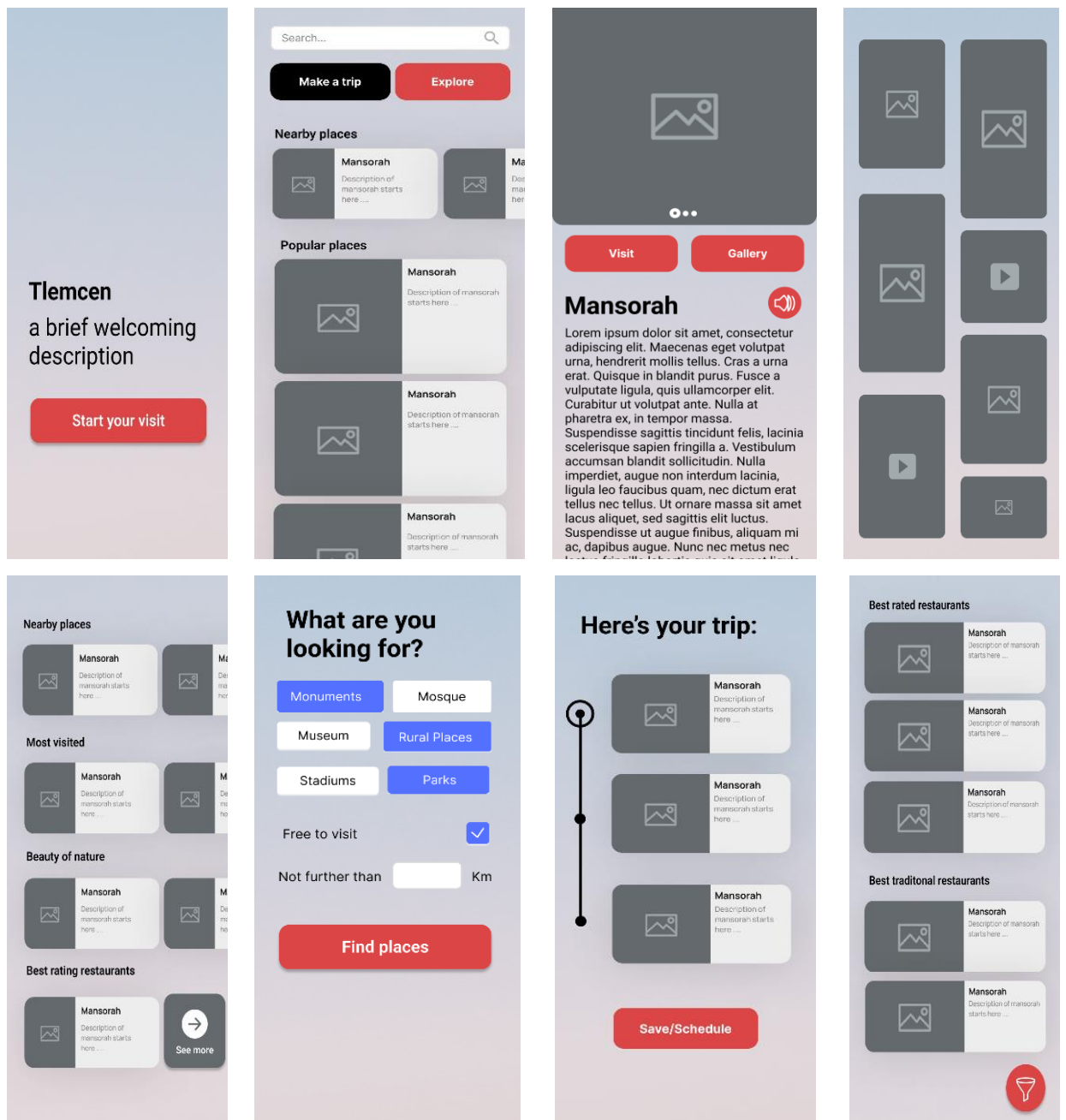


Figure 7: Maquettes d'application mobile

1.3.6. Animations

Afin de permettre à la page de charger entièrement les éléments d'écran et les données reçues des API REST sans mettre l'utilisateur dans un état d'impatience, il sera préférable d'utiliser des animations pour indiquer que l'interface utilisateur est bloquée ou pour augmenter la beauté du design de l'application.

- ❖ Pour indiquer que l'UI est bloquée :
 - Flutter Spinkit :

Une collection d'indicateurs de chargement animés de flutter.



Figure 8: Spinkit affichage

- Modal_progress_hud :

Un emballage de widget simple pour activer la progression modale HUD (un indicateur de progression modale, HUD = Heads Up Display).

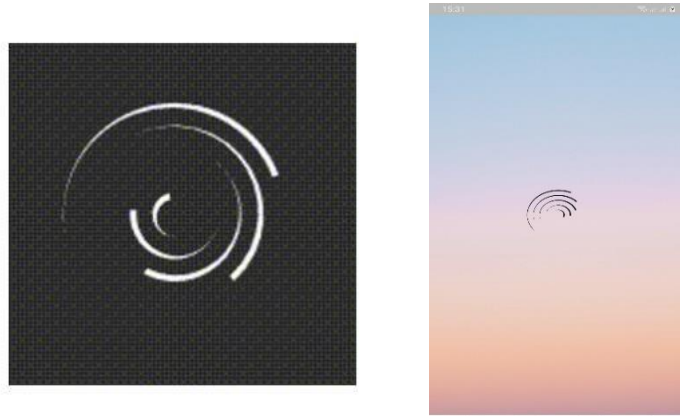


Figure 9: Modal Progress Hud

- ❖ Pour décorer le dessin :

- Animated_text_kit :

Un package flutter qui contient une collection d'animations de texte sympas et impressionnantes.



Figure 10: Animated Text Kit

1.3.7. Palettes de couleurs

Nous avons choisi une palette qui sert mieux à décorer notre interface, et à la relier au domaine du tourisme. Le choix de ces couleurs a été fait en prenant en considération le confort des yeux de l'utilisateur et pour ajouter une beauté et un sens aux lieux touristiques.

#303F9F DARK PRIMARY COLOR	#C5CAE9 LIGHT PRIMARY COLOR	#3F51B5 PRIMARY COLOR	#FFFFFF TEXT / ICONS
#FF4081 ACCENT COLOR	#212121 PRIMARY TEXT	#757575 SECONDARY TEXT	#BDBDBD DIVIDER COLOR

Figure 11: Palette couleur de l'interface

1.3.8. Logo

Pour mieux présenter notre application, nous avons créé le logo suivant qui symbolise le tourisme et l'aventure :



Figure 12: Logo d'application

1.3.9. Police du texte

Pour réaliser une application qui inspire le sérieux et le professionnalisme tout en donnant ce côté tourisme et voyage, nous avons utilisé la famille Font Fallbacks (Serif, Sans Serif, Monospace, Cursive, Fantasy) pour les paragraphes et les titres, accompagné

de la famille « Lobster » & « Permanent Marker » pour les labels dans la page détails du lieu et médias.

Tous les fonts utilisés sont sous la licence « Apache Licence Version 2.0 ».

Almost before we knew
it, we had left the
ground.

*Almost before we knew it, we
had left the ground.* **ALMOST BEFORE WE
KNEW IT, WE HAD LEFT
THE GROUND.**

Figure 13: Fonts

Les polices utilisées proviennent de google fonts (<https://fonts.google.com>).

1.3.10. Utilisation

Pour répondre aux besoins attendus et faciliter la compréhension du fonctionnement de l'application aux utilisateurs, Nous nous sommes appuyés sur un mode écran très simple d'utilisation qui ne nécessite aucune formation pour s'habituer à l'utiliser. C'est à la portée de toute personne à partir de 8 ans.

1.3.11. Boutons

Les boutons et éléments cliquables utilisés dans notre UI se distinguent des autres par l'aspect d'élévation ou d'ombre.

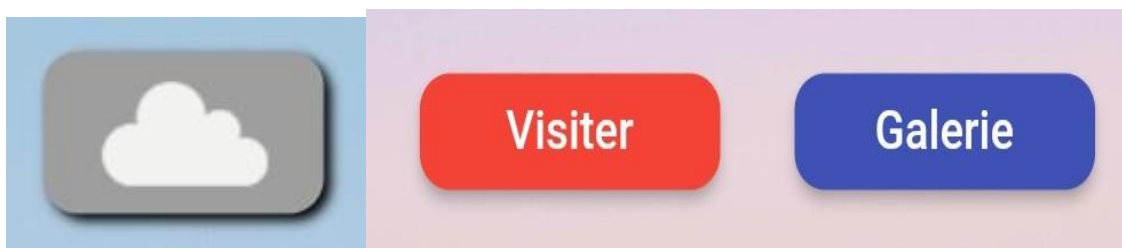


Figure 14: UI buttons

1.4. Systèmes similaires

Nous nous sommes concentrés sur deux applications différentes :

IDVizit

C'est une est une référence en matière d'itinéraires de visite. Elle offre aux utilisateurs la possibilité de personnaliser leurs profils (voyage seule ou en couple, leurs goûts en matière de tourisme : musée, shopping...) pour leurs proposer des circuits touristiques sur mesure. Aussi, l'application notifie l'utilisateur s'il s'approche d'un point d'intérêt.[1][2]

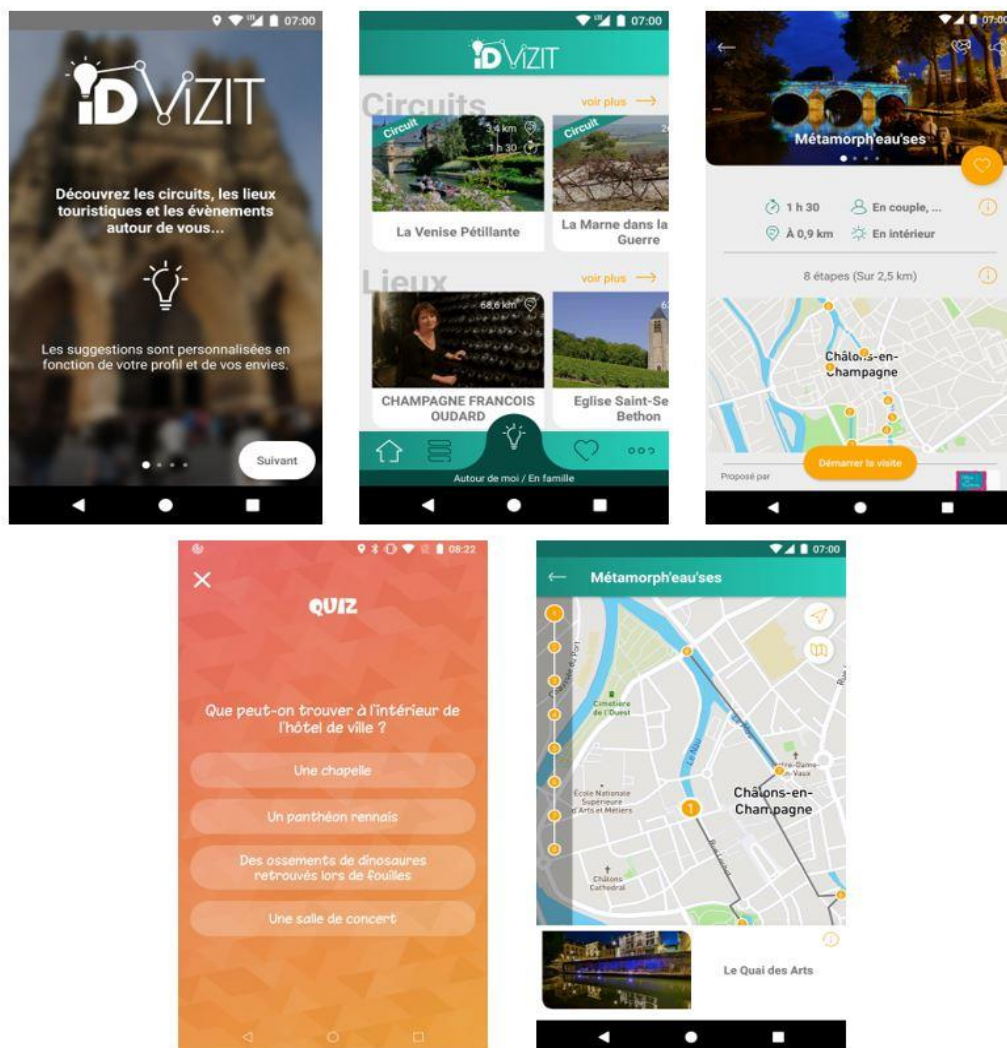


Figure 15 : Interface de l'application IDVizit

The Culture Trip : Book Travel

Originaire des États-Unis, Culture Trip est une application destinée à accompagner les touristes pour découvrir et réserver des voyages, ainsi que des séjours et des expériences sur place - organisés et triés sur le volet par une communauté mondiale d'experts du voyage et d'initiés locaux.[3][4]

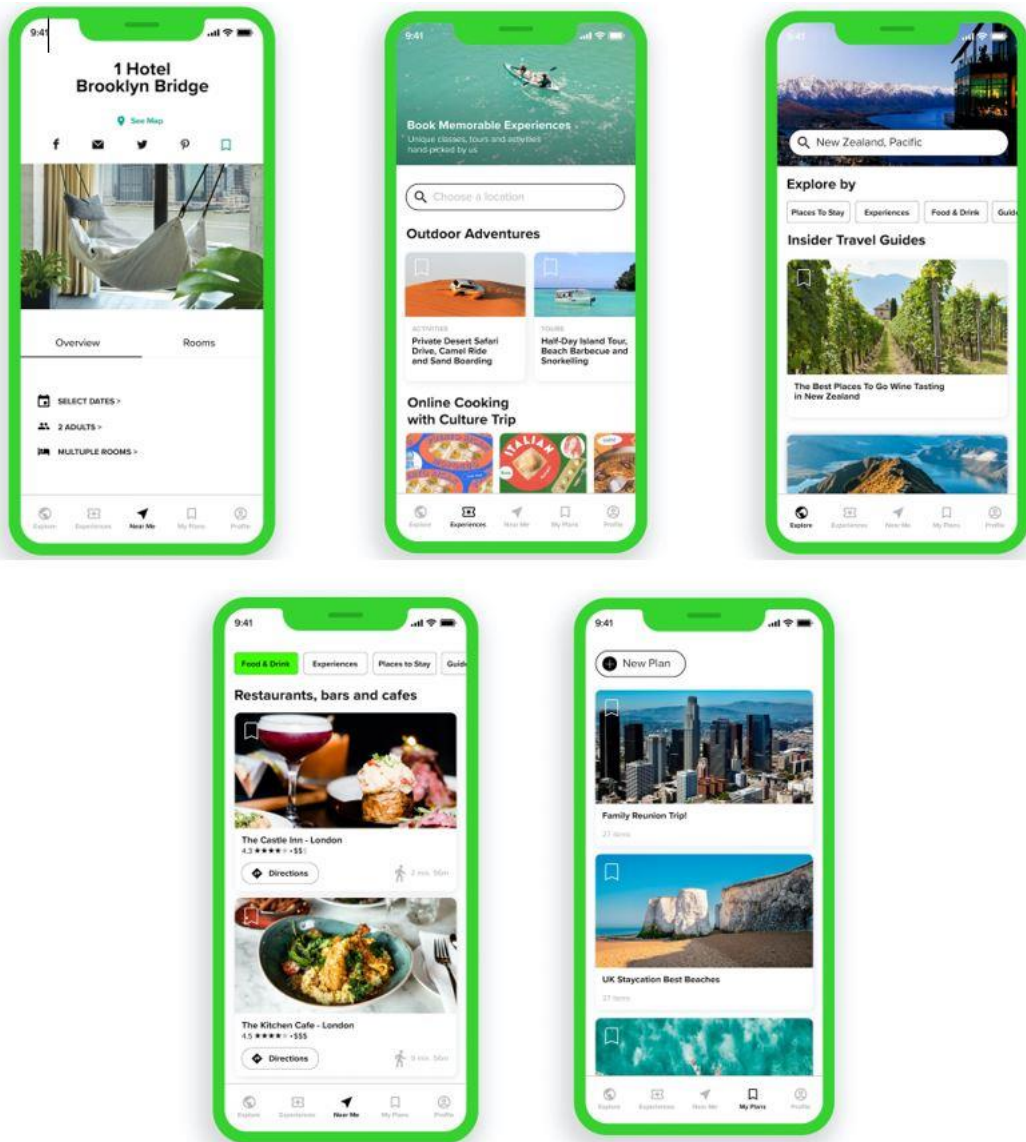


Figure 16: Interface de l'application Culture Trip

Le tableau ci-dessous résume notre étude concurrentielle des deux applications mobiles :

Tableau 6: Comparaison de la concurrence

	IDVizit	The Culture Trip: Book Travel
Site lié à l'application	https://idvizit.com	https://theculturetrip.com/
Créé en	2017	2011
Pays où il exerce	France	Etats-Unis, Grande Bretagne
Public visé	Tous	Tous + aventuriers
Cross-platform	Oui	Oui
Disponible sur	App Store + Google Play	App Store + Google Play
Création des circuits	Oui	Non (circuits proposés par l'application)
A l'échelle mondiale	Non	Non
Personnalisation du profil (places suggérées adaptées)	Oui	Oui
Notifier l'utilisateur à l'approche d'un point d'intérêt	Oui	Non
Géolocalisation	Oui	Oui
Lieux classés en catégories	Non	Non
Visualisation des Articles et Médias des lieux	Non	Oui
Enregistrement des articles et des vidéos hors ligne	Non	Oui
Notoriété	10K + Téléchargements	1 M+ Téléchargements

Pour la collecte de données sur les sites touristiques, certaines Applications dépendent de Wikitravel [11] et Wikivoyage [12] pour obtenir des informations. D'autres obtiennent ces données auprès des citoyens locaux, mais cela entraîne le risque de fausses

informations. Par conséquent, certains ont opté pour la saisie des informations par un administrateur comme solution aux problèmes de risques et de manque d'informations.

1.5. Enchaînement des écrans

Pour avoir une bonne compréhension de l'application à concevoir et bien départager les tâches, nous devons d'abord, avoir un enchaînement des écrans qui décrit le processus de notre prochaine application avant d'entamer la phase de conception. C'est comme suit :

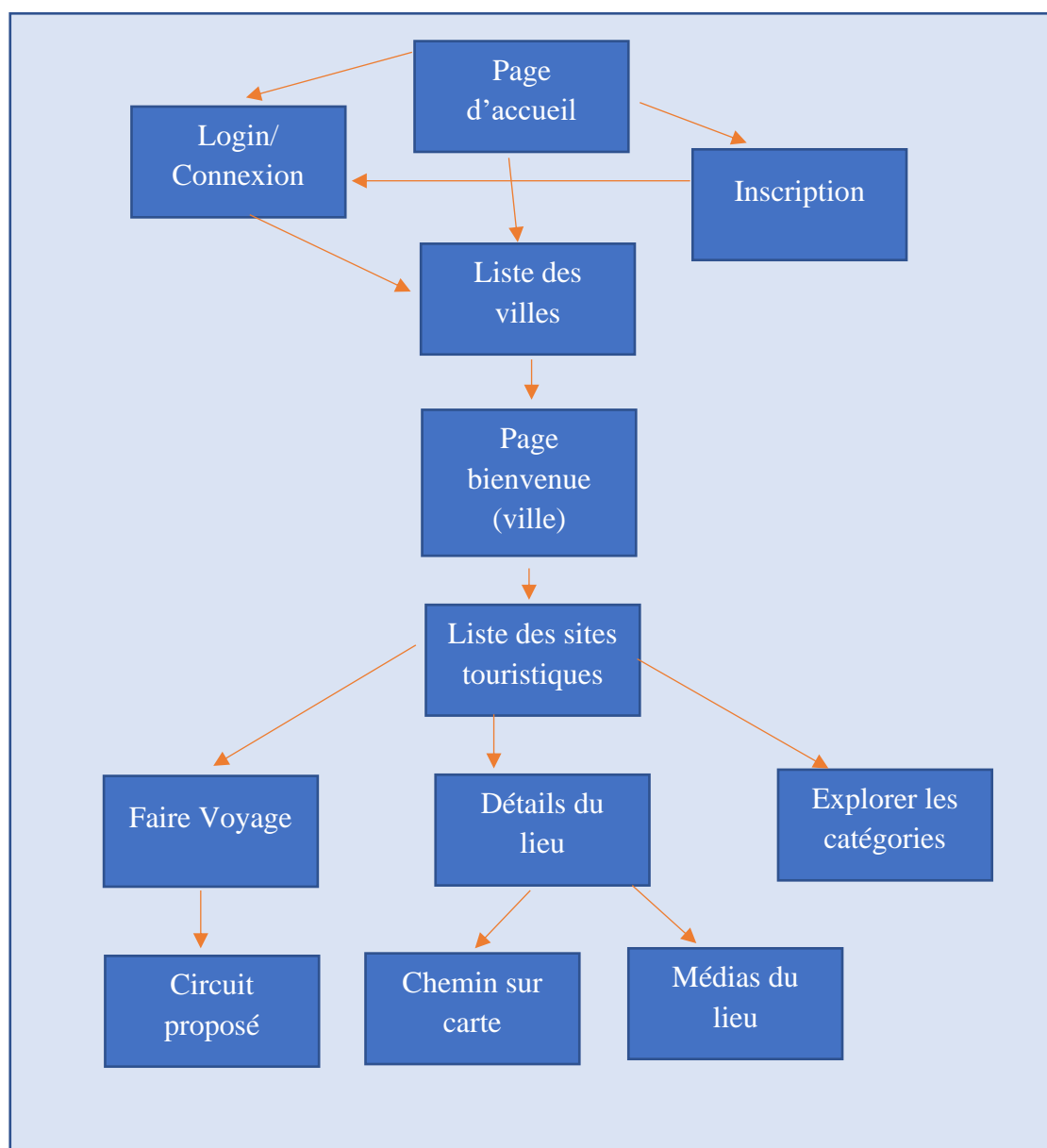


Figure 17: Enchaînement des écrans de l'application

1.6. Conclusion

Il est important d'avoir une bonne compréhension des besoins des utilisateurs, car c'est la clé pour avoir un projet bien organisé et ne pas sortir de la portée ou dépasser le temps. Il faudra ensuite définir la méthodologie de conception à suivre qui va nous amener au chapitre suivant.

2.

CONCEPTION

2.1. Introduction

Dans ce chapitre, nous allons présenter l'architecture de l'application et le patron de conception choisi, nous dévoilons aussi les différents modèles de données, le diagramme de classe, et la définition de la méthode de génération de chemin et enfin, les consignes de sécurité pour le système.

2.2. Architecture de l'application

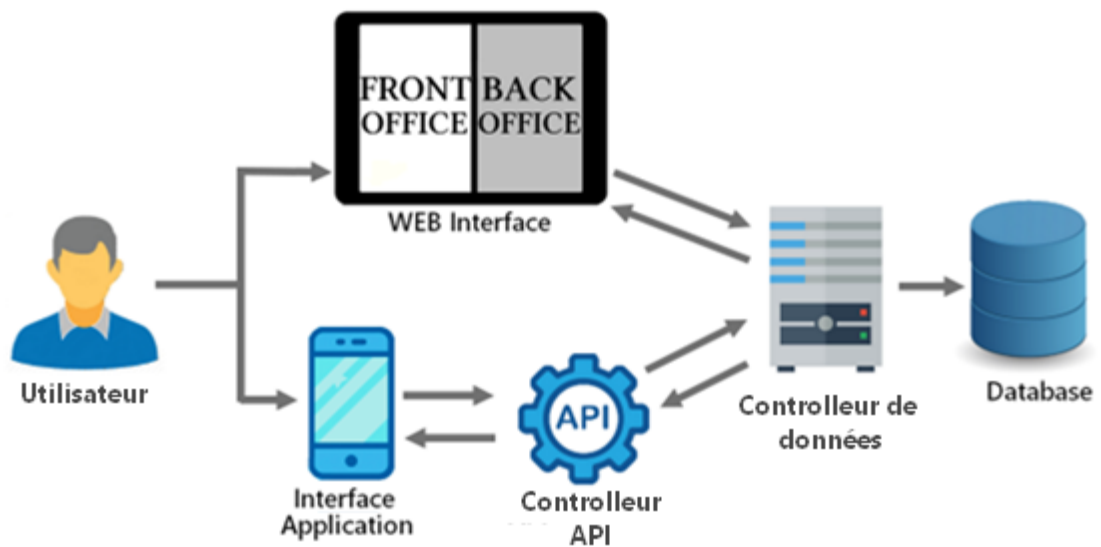


Figure 18: Architecture de l'application

L'application est accessible via deux plates-formes différentes, qui sont le web et le mobile, donc l'architecture de l'application est composée de six composants essentiels comme indiqué sur la figure ci-dessus :

- L'utilisateur : peut-être soit un touriste, soit l'administrateur.
- L'interface web : elle-même composée de deux piliers :
 - Front office : désigne la partie visible par les clients, la façade de l'application sur le web. Les visiteurs ont la possibilité de profiter des services offerts.
 - Back office : la partie responsable de la gestion des données, son accès est restreint aux administrateurs seulement.
- Application Mobile : qui présente la façade de l'application sur les plateformes mobiles (Android et iOS). L'administrateur ne peut pas gérer les sites touristiques

via cette interface, mais ce sont les visiteurs et les touristes qui peuvent profiter des services de l'application via leur mobile.

- **Contrôleur d'API** : désigne la partie de la communication avec le système d'information. Celui-ci envoie ensuite des requêtes d'accès aux données au serveur afin de répondre aux besoins avec les informations requises.
- **Contrôleur de données** : représente la partie responsable de l'exécution des requêtes et de la gestion de la base de données contenant la logique métier.
- **Base de données** : le composant dans lequel les données sont stockées, tels que les données des utilisateurs, des villes, des lieux et leurs médias (vidéos, images, sons).

2.3. Patrons de conceptions

En développement logiciel, un patron de conception est un agencement caractéristique de modules, reconnu comme une bonne pratique pour un problème de conception. Il décrit une solution type adaptable selon la conception du logiciel.

Le patron utilisé dans notre projet est le suivant :

2.3.1. Patron Modèle-vue-contrôleur

Le MVC (Model-View-Controller) est un motif utilisé dans l'architecture logiciel informatique, il impose une séparation entre les vues, les actions et les modèles des données, se compose de trois parties :

- **Model** : Élément contenant des données et une logique liée à la manipulation de ces derniers, comme la modification ou la suppression. Il peut imposer des contraintes ou contenir des méthodes d'affichage partiel ou total des données. Au cas de mise à jour de données, ce dernier va notifier la vue pour prendre en charge les changements. Il est indépendant des autres éléments.
- **Vue** : Elle représente l'interface graphique, son rôle est d'afficher selon une logique définie, les données récupérées depuis le model ainsi que prendre les entrées d'utilisateur (clique sur bouton, remplissage d'un formulaire...) et les envoyer au contrôleur.

- **Contrôleur** : Son rôle est de lier les évènements reçus par la vue et exécuter les actions appropriées, ça peut être une affectation aux données qui donc va nécessiter un appel au model, ou un changement de vue ou un enchaînement des deux [5][6].

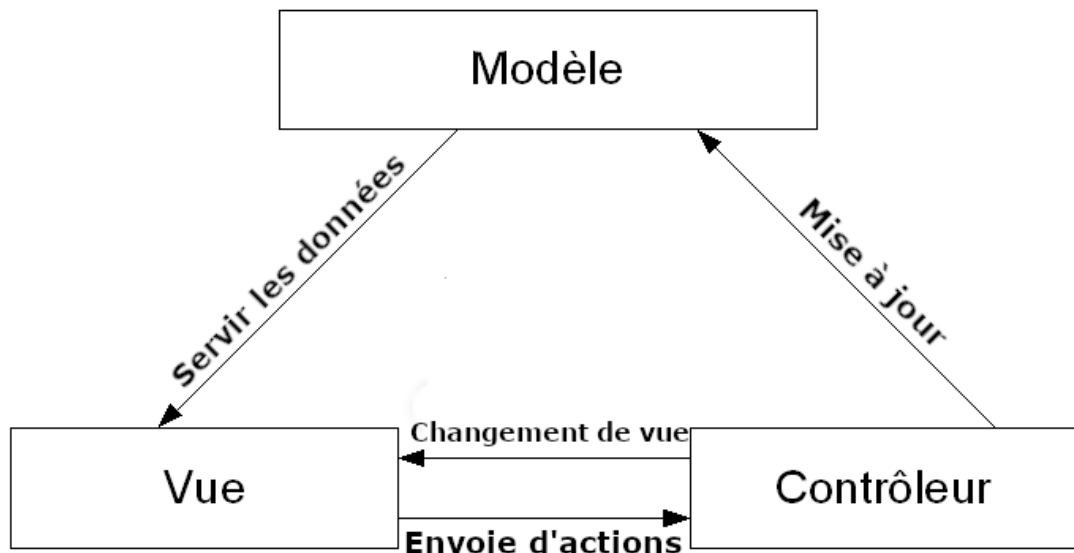


Figure 19: Schéma d'interactions MVC [13]

2.3.2. Les effets du MVC sur un projet

Tout au long du développement d'un projet, celui-ci va grandir et disposer de plusieurs interfaces et modèles de données. Par conséquent, une répartition des différents éléments sera nécessaire. C'est pourquoi l'utilisation du pattern d'architecture MVC sera utile, car il facilite la gestion d'un projet et lui apportera plusieurs effets tels que :

- Une collaboration facile entre équipe de développement.
- Définition facile de la logique-métier pour la manipulation de données sans avoir d'effets sur les autres éléments.
- Une facilité pour retrouver les différents modules ou fonctions de votre code.
- Un gain de temps à la modification ou ajout de nouvelles fonctionnalités.[7]

2.3.3. Projection

Le modèle MVC est souvent utilisé dans le développement web, de nombreux Framework l'utilisent à travers différents langages. Par exemple, en JavaScript, nous pouvons trouver : AngularJS ou VueJS, en Python : Django, Flask, et enfin en PHP : le Framework le plus populaire dans ce langage : Laravel.

Ainsi, notre application web est basée sur Laravel, donc basée sur une architecture MVC. Par conséquent, nous pouvons distinguer la séparation des différents éléments où :

- Les vues : sont des fichier `*.blade.php` localisé dans `~/resources/views/`
- Les contrôleurs : sont localisés dans `~/app/Http/Controllers/`, et peuvent être créé à partir de la console en utilisant l'interface de ligne de commande artisan :
`php artisan make:controller {{ Nom Contrôleur }}`
- Les modèles : sont localisés dans `~/app/Models /`, et peuvent être créés avec l'interface de ligne de commande artisan : `php artisan make:model {{ Nom Model }}`

Dans le cas de notre application web, nous avons utilisé plusieurs de ces éléments, pour les vues, les interfaces front-office et back-office sont distinctes mais qui incluent les mêmes données, par exemple le modèle User qui à travers le contrôleur UserController permet la création d'un utilisateur, la vérification des identifiants ou la modification des données personnelles.

2.3.4. Authentification (exemple)

En prenant l'exemple de l'identification, nous pouvons voir que nous avons deux vues distinctes : une pour le frontend, et une pour le backend. Pourtant, le mécanisme de vérification se fait par la même méthode via le contrôleur **UserContrôleur** qui appellera la méthode du model **User** qui à son tour vérifiera les identifiants puis mettre à jour la vue en envoyant les données des résultats.

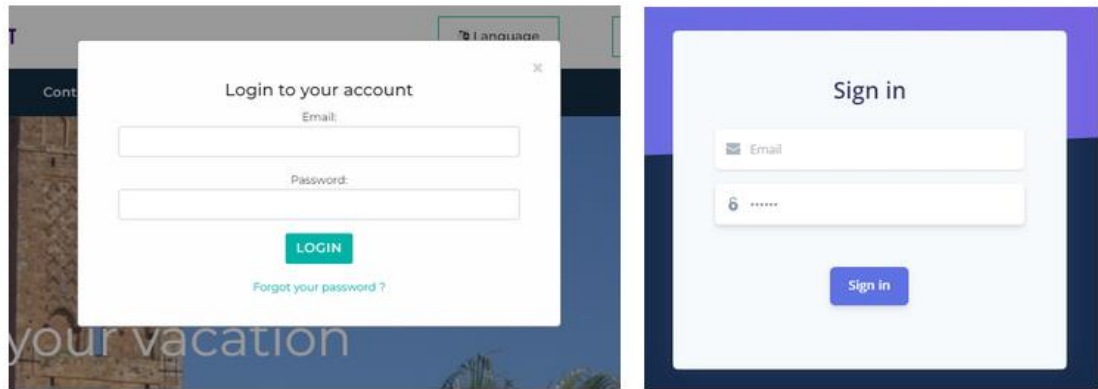


Figure 20: Interface d'identification : front-office a gauche - back-office a droite

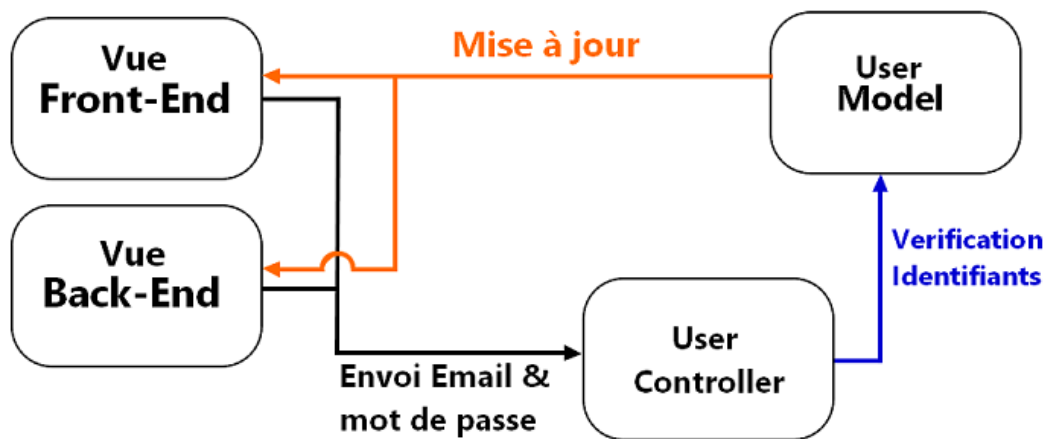


Figure 21: Schéma de l'action d'authentification

2.3.5. Ajout d'une place touristique (exemple)

Les lieux touristiques doivent être ajoutés par un administrateur via une vue dédiée dans le backend. Cette vue contient la liste des lieux déjà inscrits et un formulaire dont le rôle est d'ajouter une nouvelle entrée. En remplissant un formulaire, les données saisies seront soumises au contrôleur PlaceController qui les affectera à son tour au modèle. Ensuite, ce dernier va notifier la vue pour mettre à jour la liste avec la nouvelle ligne.

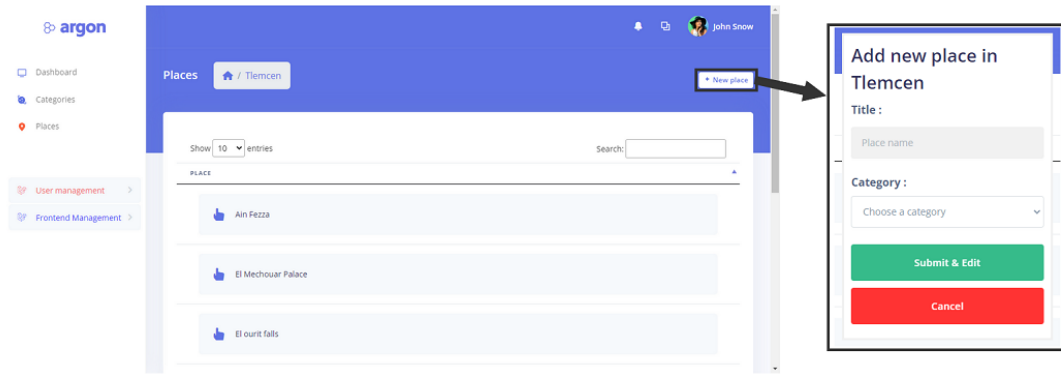


Figure 22: Vue de la liste des places et le formulaire d'ajout

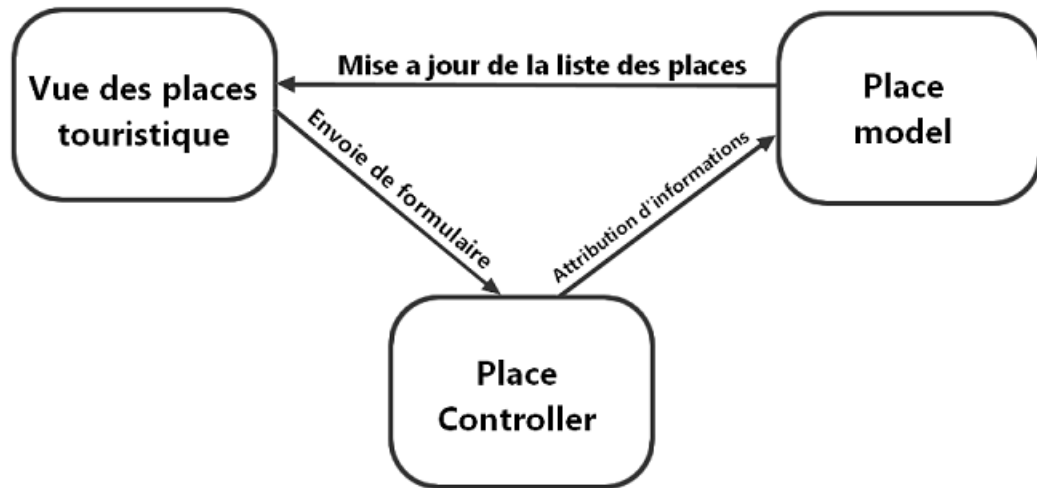


Figure 23: Schéma de l'action d'ajout de places

2.4. Modèle de données

Le modèle des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements.

Ainsi, le modèle logique est dépendant du type de base de données utilisé. Il consiste à donner la structure des données selon un marquage spécifique. Et ensuite, il peut être utilisé pour créer la base de données [7].

2.4.1. Modèle entité relation (MCD)

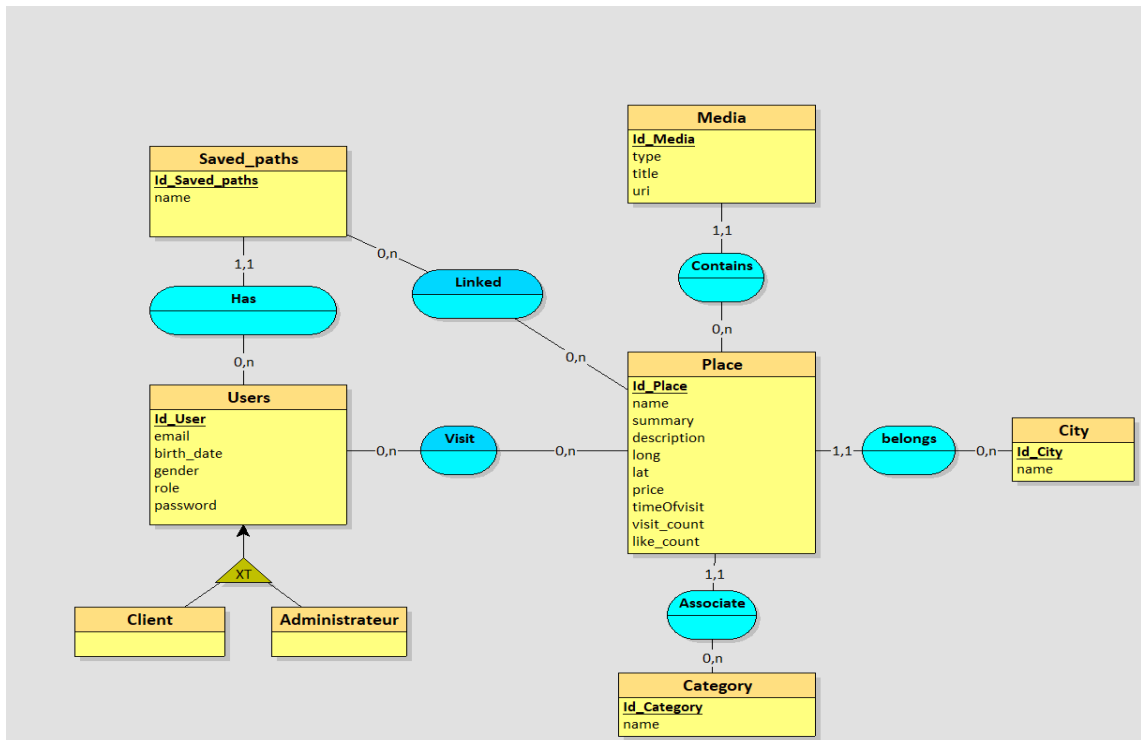


Figure 24: Diagramme entité relation

2.4.2. Modèles logiques des données (MLD)

- **Users** (id_user, email, password, birth_date, gender, role)
- **Category** (id_cat, name)
- **City** (id_city, name)
- **Places** (id_place, name, summary, description, long, lat, price, time, visit_count, like_count, id_city#, id_cat#, thumbnail_id#)
- **Media** (id_media, mime_type, uri, title, id_place#)
- **Saved_paths** (id_saved_path, name, id_user#)
- **Points** (id_saved_path#, id_place#)

Donc d'après ces schémas, nous pouvons décrire les données comme suit :

- Un utilisateur est défini par un identifiant comme clé primaire, ainsi que d'autres données pour son identification (e-mail et mot de passe), d'autres informations personnels (année de naissance et son sexe) et un rôle pour lui attribuer les privilèges requis.
- Deux tables pour les cités et les catégories, ayant un identifiant comme clé primaire et un nom.
- Une table Places qui contiendra les places touristiques, ayant comme clé primaire un identifiant, elle comprendra le nom de la place, une description et un résumé de description qui sera utilisé pour les cartes de miniature avec un identifiant d'une photo récupéré depuis la table Medias. Cette place sera liée à une ville par son identifiant obtenu depuis la table cities, elle aura d'autres informations additionnelles : ses données géographique, une durée de visite et un montant de visite ainsi que des mesures de statistiques seront enregistré comme le nombre de visite et le nombre de gens qui aiment cette place.
- La table media contiendra la liste des différents éléments des galeries des places, ayant comme clé primaire un identifiant, un fichier media aura un lien URI, un titre, son type sera obtenu systématiquement et stocké dans un champ mime_type pour l'aide à l'affichage dans l'interface utilisateur, et enfin, il sera lié à une place.
- La table Saved_paths contiendra les chemins générés sauvegardé pour une consultation ultérieure, elle aura comme clé primaire un identifiant, un nom pour faciliter sa recherche et sera lié avec l'utilisateur.
- La table points qui vas lier les chemins avec leur points correspondants, cette association sera prise comme clé primaire et donc n'aura pas besoin d'autres champs.

2.5. Diagramme de classe

Après l'études des fonctions requises, ainsi que les attributs des modèles des données et leurs associations, nous pouvons déduire un diagramme de classe représenté

comme suit :

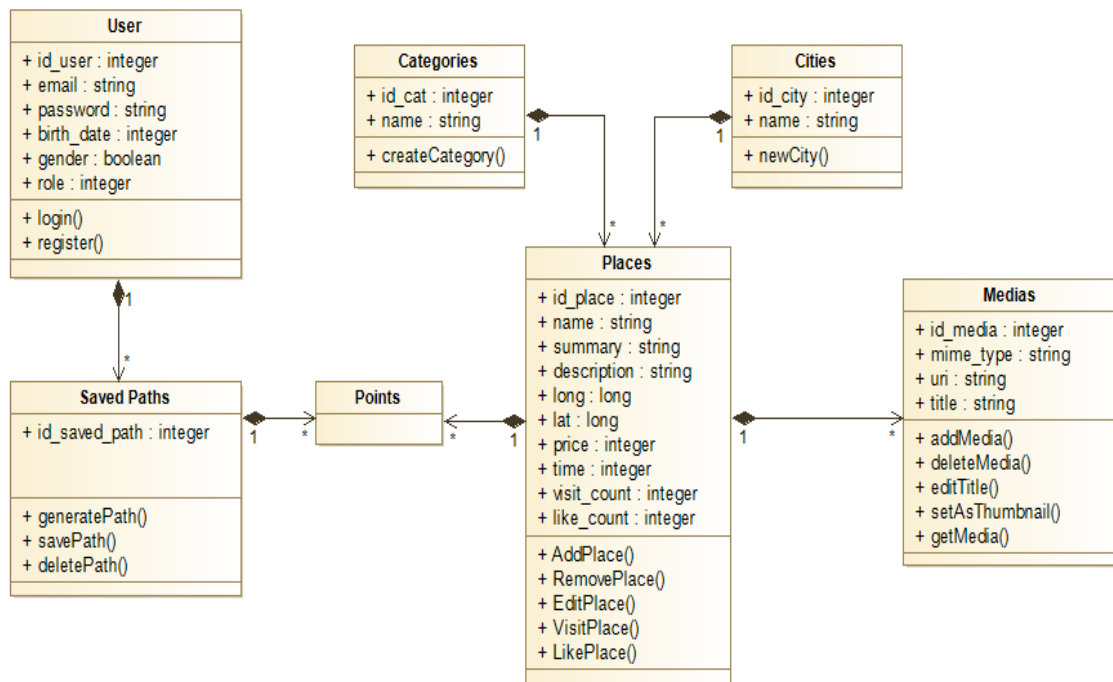


Figure 25: Diagramme de classe

2.6. Génération de chemin

La génération de chemin est une fonctionnalité importante et très utile pour le client, cette option lui permettra de créer une sélection de sites touristiques personnalisée, il pourra choisir les types des places qu'il veut visiter, limiter le prix de la visite, limiter le temps du trajet ainsi que la distance maximale à partir de sa localisation actuelle ou d'un point choisi.

Après la définition des contraintes, le système aura alors, un processus assez complexe pour la recherche des points. La méthode aura comme entrée les critères suivants :

- Categories [] : La liste contenant les catégories choisies
- LimitePrix : La valeur de la limite de prix
- LimitTemp : la valeur du temps de trajet
- LimitDist : la valeur de la distance du trajet

Au début, toutes les places de la ville et les catégories choisies qui respectent le montant défini sont récupérées. Puis, deux boucles seront lancées sur le résultat trouvé : une qui vas pointer sur la position de la liste et une pour naviguer sur la liste. Ensuite,

l'algorithme de recherche va calculer le point le plus court, sur la première position avec le point de départ, et puis, la distance entre la place de la position courante et position-1, et puis stocker la position du point le plus court dans une variable minDistPos. À chaque itération, deux attributs seront ajoutés : distance et temps qui va être calculé en passant par une méthode qui va faire appel au service de cartographie google par un API, une vérification est nécessaire pour s'assurer que le temp limite est respecté diminuant le nouvel attribut (temp) de la variable d'entrée LimiteTemp, et le nouvel attribut (distance) de la variable (LimitDist). Si les deux contraintes sont respectées, alors les points de la position courante et celui de la distance minimale seront permutés. Sinon une des deux opérations ont un résultat inférieur à 0, alors la queue de la liste sera supprimée, et le résultat sera retourné à l'utilisateur.

A la fin, les résultats du calcul sont affichés à l'utilisateur, il pourra consulter le texte de résumé de chaque place, ouvrir le chemin dans Google Maps, ou sauvegarder le chemin pour une consultation ultérieure, un nom sera demandé pour faciliter le repérage de ce dernier.

2.7. Sécurité du système

La sécurité des applications n'est pas un besoin non-fonctionnelle ou un avantage de plus, c'est une nécessité absolue. Elle doit être prise en considération à partir de l'écriture du premier code jusqu'à la dernière ligne de code.

2.7.1. Sécuriser le code

Quelques recherches montrent que les codes malveillants affectent plus de 11 millions d'appareils mobile. En déduction de cela, les bugs et les vulnérabilités d'un code sont les portes ouvertes pour les attaquants afin de pénétrer une application ou même pour un échec total d'un projet informatique. Cependant, il est nécessaire d'exécuter plusieurs tests sur des différentes plateformes avec des différentes données afin dans le but de corriger ces bugs au fur e à mesure. Il vaut mieux d'ajouter même des commentaires pour expliquer chaque partie du code source.

2.7.2. Utilisations d'APIs autorisées

Les APIs non-autorisées peuvent accorder involontairement des privilèges malveillants à des pirates. Donc, il est indispensable d'utiliser des clés API valides fournies par une source fiable (nous pouvons citer deux exemples depuis ce projet : la clé fournie par OpenWeatherMap, et la clé fournie par google cloud console, qui sont des clés privées à ce projets).

2.7.3. Le principe du moindre privilège

Le principe du moindre privilège signifie que les applications s'exécutent avec uniquement les permissions/autorisations dont ils ont besoin et pas plus.

2.7.4. Communications via les protocoles sécurisés

L'utilisation des protocoles chiffrés est importante pour assurer la sécurité de la communication, pour la navigation il faut prendre en compte l'usage du protocole HTTPS qui est basé sur le protocole HTTP, ceci permet de vérifier l'authenticité d'un site web grâce à des certificats d'authentification afin d'éviter toute fraude liée aux informations personnels.

Ainsi pour le développement, l'accès aux machines virtuels distantes se fait à travers le protocole SSH (Secure Shell) qui vas grâce à des clés de chiffrements, coder la session de communication pour éviter toute lecture des commandes par une partie tierce.

2.7.5. Tester à plusieurs reprises

Le phase des tests est très importantes, il faut tester chaque module en isolation, et ensuite en interaction avec les autres modules. Ces étapes améliorent la sécurité et la confidentialité du projet et garantit le bon fonctionnement et l'obtention des résultats correctes.

2.8. Conclusion

Au cours de ce chapitre, nous avons présenté l'étape d'analyse et la phase de conception, constituées de l'architecture logicielle pour mieux décrire le fonctionnement de notre système et faciliter la réalisation et la maintenance. Par la suite, nous avons défini le MVC et l'avons projeté sur notre projet en donnant un exemple. Enfin, nous avons conçu le diagramme de classe afin d'illustrer globalement la structure des éléments constituant la base de données associée à notre application. Dans le chapitre suivant, nous entamerons l'étape de réalisation.

3.

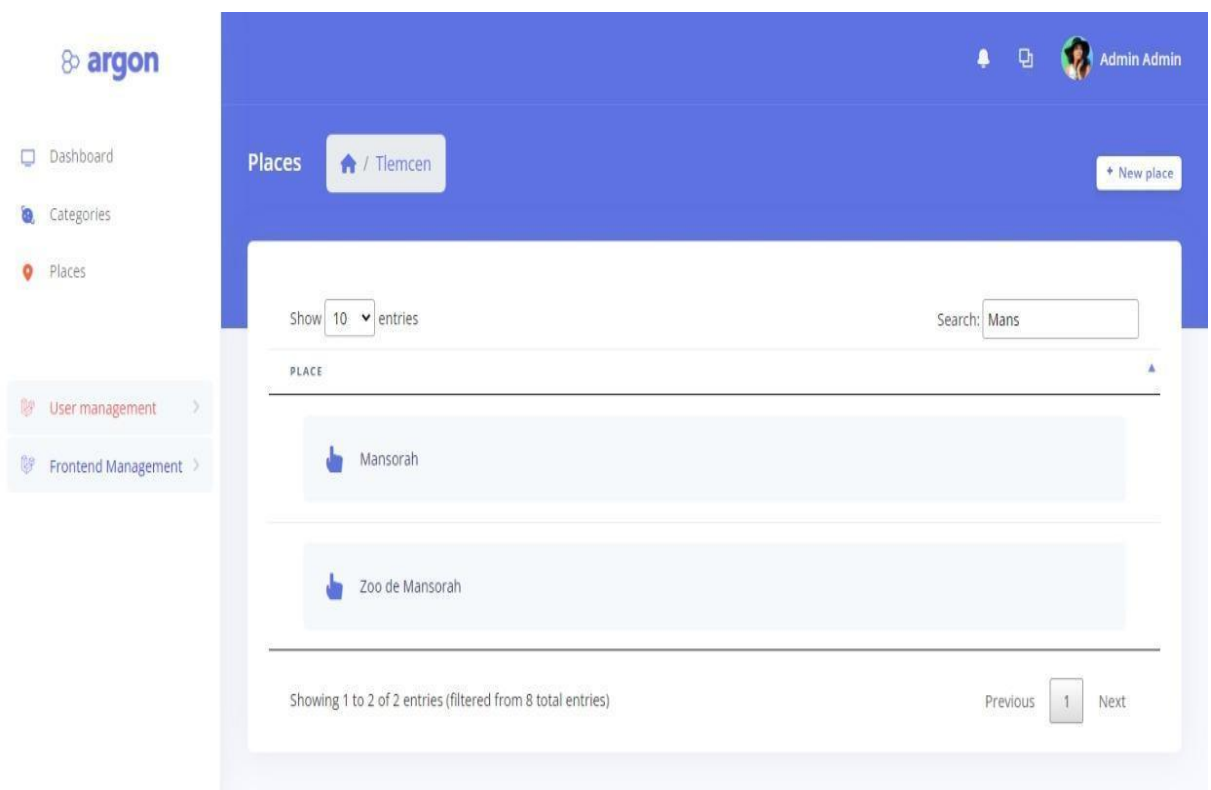
**REALISATION ET
DEVELOPPEMENT**

3.1. Introduction

Dans ce chapitre, nous allons présenter les interfaces finales (pour le web et mobile) tous les outils nécessaires à la création et à la mise en place de notre application. Tout au long du projet, nous avons eu besoin de différents types d'instruments pour différentes tâches, tels que des outils de conception de logo, différents diagrammes et maquettes, ainsi qu'un environnement de développement (IDE), des langages de programmation, des packages liés au flutter, des services web et un serveur utilisé dans notre application. Ensuite, nous mentionnerons lequel des outils est libre d'utiliser ou nécessite une autorisation, et nous terminons par une conclusion.

3.2. Interfaces finales

❖ Pour le web :



Places [/](#) [Tlemcen](#) / [Mansourah](#)

General info

Title

Category

Summary

Description

Mansourah Mosque is a ruined historic mosque in the city of Mansourah, Algeria, dates back to the Marinid dynasty. It consists of a part of the Mansourah Castle. The excavation has revealed regarding the design of the mosque that it was built by the Sultan Abu Yakub in 1303, and the decoration of the main gate was done by Abu al-Hassan in 1336. Inside the yard there are 13 doors which surrounded by the wall, and a sahn with a fountain in the middle. The sahn is surrounded by three corridors, of which employ similar tilework with the prayer hall. The prayer hall is connected directly to the sahn and contains tilework similar to the qibla wall, the design which can be observed as well on the 13th-century Bab el Bhar in Fez. The main entrance to the mosque leads to the main prayer hall, which is square-shaped, covered by the

Metadata

Geographical coordinates

Longitude Latitude

Price for the visit Time to visit

[Open gallery](#) [Submit changes](#)

Figure 27: Prototypes des interfaces finales pour l'administrateur (Web)

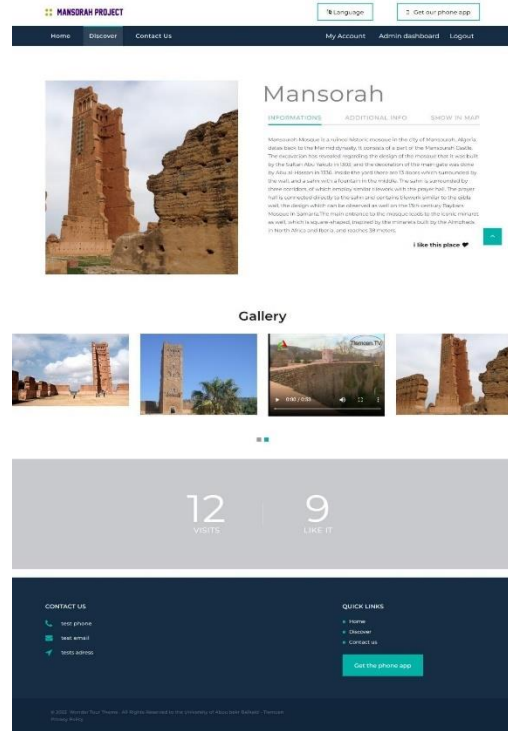
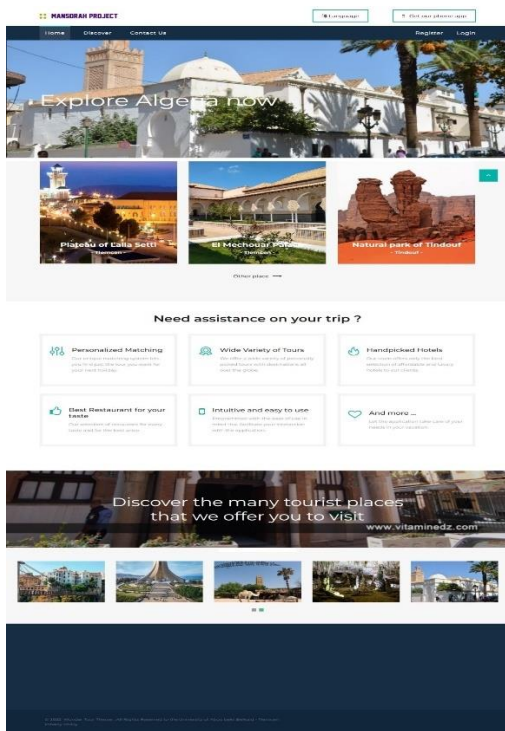
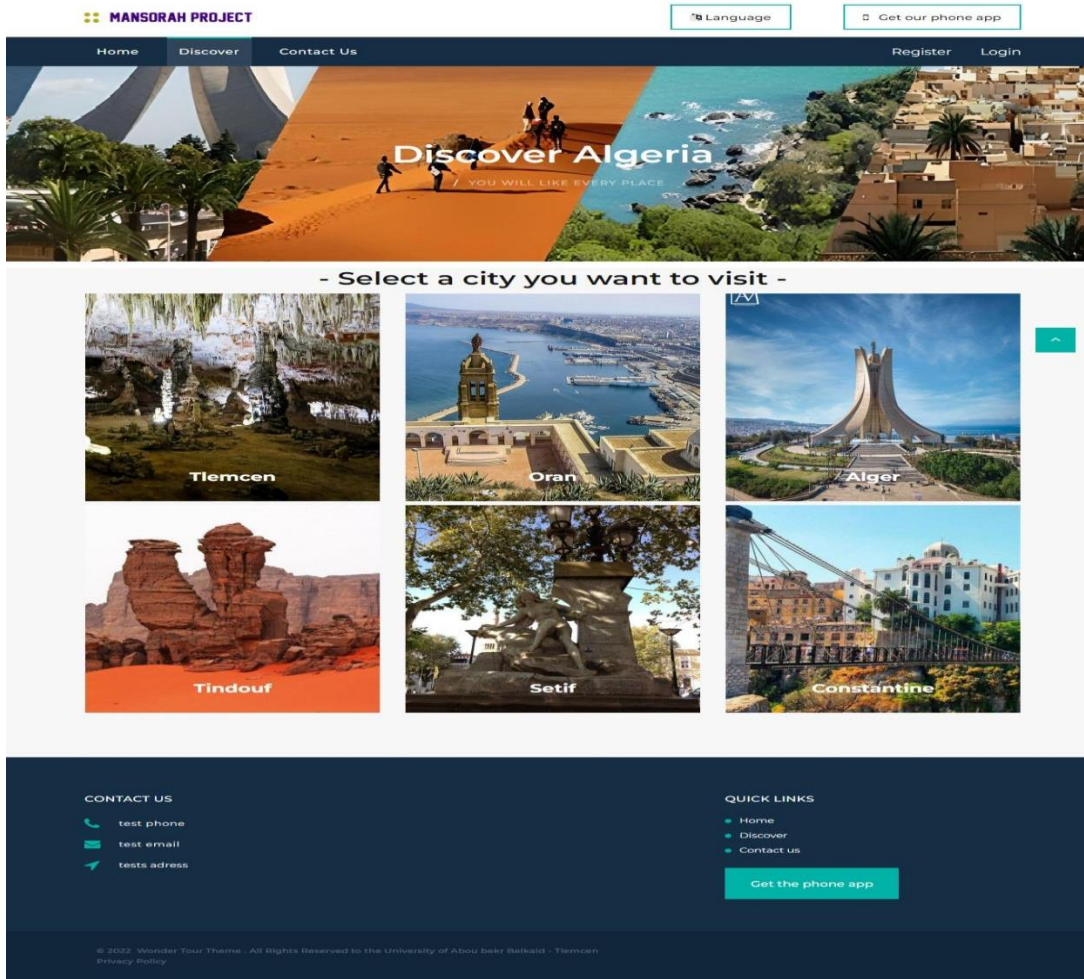


Figure 28: Prototype des interfaces finales pour les touristes (Web)

❖ Pour le mobile

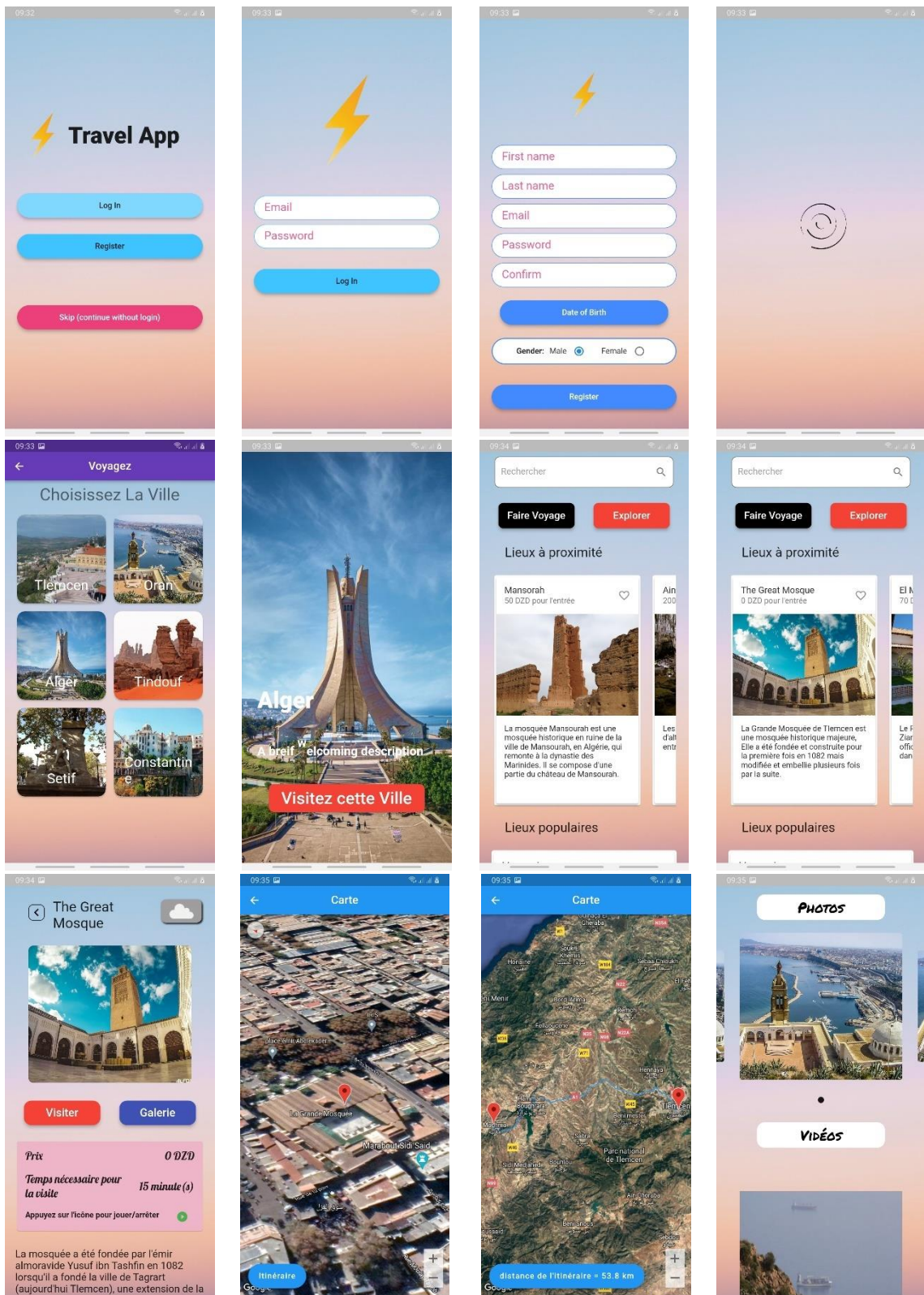


Figure 29: Prototype des interfaces finales (Mobile)

3.3. Outils de conception UML

- **Modelio :**

Modelio est avant tout un environnement de modélisation, prenant en charge une large gamme de modèles et de diagrammes, et fournissant des fonctionnalités d'assistance aux modèles et de vérification de la cohérence. Aussi, il combine la prise en charge BPMN (Business process model and notation) et la prise en charge UML dans un seul outil, avec des diagrammes dédiés pour prendre en charge la modélisation des processus métier.

- **Lucid Chart**

Lucid chart est une plateforme qui offre un service de création des schémas et diagrammes en ligne, l'interface utilisateur est simple et facile à manipuler et offre des style prédéfinis et créer différentes vus pour un seul diagramme.

3.4. Outils de conception graphique

- **GIMP**

Un outil d'édition et de retouche d'image, adaptés à la création de logo. Il en existe des versions pour la plupart des systèmes d'exploitation dont GNU/Linux, macOS et Microsoft Windows.[8]

- **Figma**

Figma est un outil graphique de prototypage. Il est basé sur le web, il permet de concevoir, de collaborer, de créer des prototypes et des maquettes pour différents type de systèmes.

3.5. Environnement de développement

3.5.1. Protocoles, marquages & langages de programmation

- **SSH**

Le SSH est un protocole de communication informatique sécurisé, et un programme qui permet l'accès au terminal d'une machine à distance via un réseau. Cette communication se fait après avoir fait un échange de clés chiffrées au début de la connexion pour assurer une connexion fiable.

- **Blade**

Blade est un moteur de création de modèle. Il comprend un marquage atypique mais simple, il est utilisé par le Framework Laravel pour générer les vues dynamiques.

- **HTML**

HyperText Markup Language est un langage de marquage de pages web, développé par le World Wide Web Consortium il permet de placer les différents éléments d'une page web et il est généralement accompagné par une feuille de style CSS et des actions en forme de script en Javascript.

- **Bootstrap**

Bootstrap est un Framework CSS qui se compose de plusieurs outils pour styliser des pages web. Son but est de faciliter la responsivité d'un site web en divisant une page sur des colonnes et spécifier une taille d'affichage pour chaque type d'écran.

- **JavaScript (JS)**

Javascript est un langage de programmation utilisé dans les pages web dans le but d'exécuter des scripts, il ne dépend d'aucune plateforme tant qu'elle soit disposée d'un navigateur web compatible.

- **PHP**

Le PHP (Hypertext PreProcessor) est un langage de programmation orienté objet utilisé généralement dans le développement web, il est le langage le plus utilisé dans ce domaine. Il est compatible avec multiple plateformes tel que Windows et Linux.

- **JSON**

JSON est un format de fichier standard ouvert et un format léger d'échange de données qui utilise du texte facile à lire par l'homme pour stocker et transmettre des objets de données constitués de paires attribut-valeur et de tableaux. Il s'agit d'un format de données commun avec diverses utilisations dans l'échange de données électroniques, y compris celui des applications web avec des serveurs. Il est aisément analysable ou généré par les machines.

- **Flutter & Dart**

Flutter est un SDK d'interface utilisateur (UI toolkit) Open Source développé par Google. Il permet le développement des applications multiplateformes pour iOS/Android et utilise Dart comme langage de programmation. Dart est un langage de programmation Open Source côté client. Il est facile à apprendre, stable et crée des applications performantes. Flutter met à disposition ses propres Widgets qui représentent des éléments graphiques, ce qui rend la conception des interfaces graphiques considérablement simplifié. De plus, grâce aux fonctionnalités Hot Reload & Hot Restart, la création d'applications est très rapide et le temps de compilation est fortement réduit. Récemment, nouvelles règles ajoutées à Flutter 2.0 telles que Sound Null Safety qui aide à mieux gérer les variables au cas où elles seraient nulles.[9]

Le choix de cette technologie s'est effectué après une étude comparative entre deux technologies de développement mobile, proposées par le client (besoin non fonctionnel), et qui sont ReactNative et Flutter. L'étude s'est déroulée comme suit : [10][11]

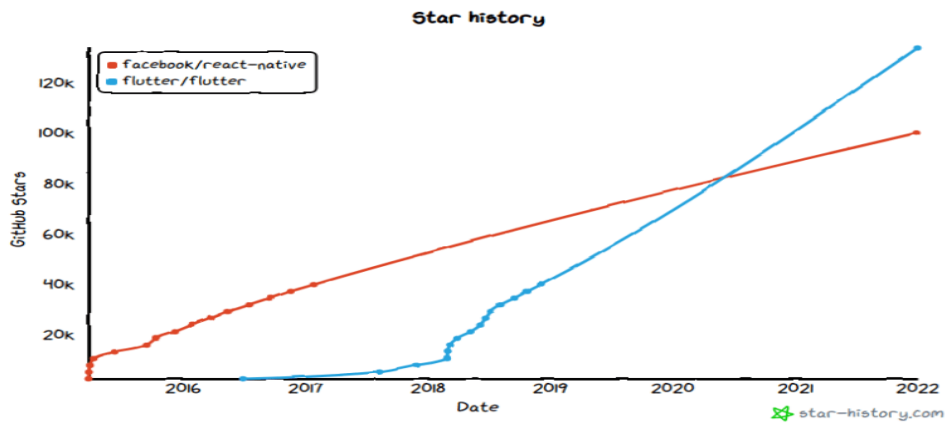
Différence entre Flutter et React Native :

Tableau 7:Différence entre Flutter & React Native[14]

Caractéristiques	Flutter	React Native
Développé par	Google (2017)	Facebook (2015)
Utilisé pour développer	Application mobile, web et de bureau	Application mobile
Nature	UI toolkit/ mobile Framework	Framework

Langage de programmation	Dart	JS
Open source et gratuit	Oui	Oui
Multiplateforme	Oui	Oui
UI (responsive)	Elevé	Très élevé
Hot reloading	Oui	Oui
Support des bibliothèques tierces	Oui + APIs des plateformes & SDKs	Oui
Performance	Important et plus rapide	Elevé
Installation	Moyenne	Facile
Documentation	Bonne documentation	Documentation moyenne
Test	Offre un ensemble important pour les tests unitaires, d'intégration et de widgets	Dépend d'outils tiers
Avantages	-Offre des Widgets -Utilisé pour les applications complètes (stockage, appareil photo, géolocalisation...)	

Flutter et React Native ont tous deux une grande communauté et une popularité avec une petite différence. Flutter est plus populaire mais React Native a une plus grande communauté. Un autre aspect important est que la plupart des développeurs prennent en charge React Native en raison de l'utilisation de ReactJS pour le web et non pour les applications mobiles. Aussi, les développeurs de flutter sont plus actifs sur GitHub, Facebook ou twitter et Flutter augmente à grande vitesse.



STATISTICS ON FLUTTER AND REACT NATIVE IN 2021		
	Flutter	React Native
StackOverflow VOICES	68.8% ↑	57.9%
Github CONTRIBUTORS	118k ↑	94.6k
Google Trends QUERIES PER DAY	86 ↑	58

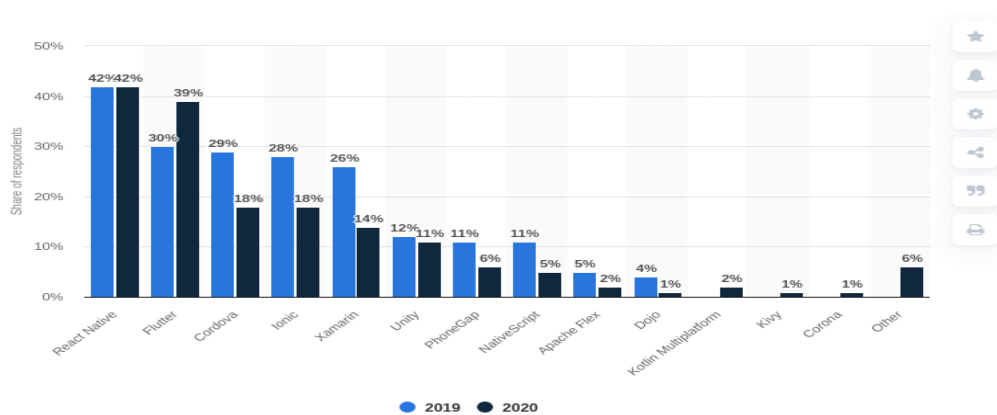


Figure 30: Graphes et statistique de Flutter et React Native

Installation et configuration

Pour installer et exécuter Flutter, l'environnement de développement doit satisfaire les conditions suivantes :

- Système d'exploitation : Windows 10 ou version ultérieure (64 bits), basé sur x86-64.
- Espace disque disponible : 1,64 Go (n'inclut pas l'espace disque pour l'IDE/les outils)
- Outils :
 - Windows PowerShell 5.0 ou plus récent (préinstallé avec Windows 10)

- Git pour Windows 2.x, avec l'option Utiliser Git à partir de l'invite de commande Windows.
- Un émulateur ou périphérique physique : pour tester durant l'implémentation.

Si ces conditions sont vérifiées, pour se servir de cette technologie, il faut d'abord télécharger le SDK Flutter et l'extraire dans un dossier. Puis, ajouter le chemin vers Flutter dans l'environnement des variables PATH afin de pouvoir utiliser les commandes Flutter dans la console Windows. Vérifier par la suite en utilisant la commande « flutter doctor » qui affiche un rapport sur l'état d'installation de Flutter. Ensuite, ajouter l'emplacement du SDK Android à l'environnement de variable ANDROIDHOME et PATH. Enfin installer le plugin flutter en acceptant Dart et créer un projet.

3.5.2. Outils de développement

- **Android Studio**

Android Studio est l'environnement de développement intégré officiel du système d'exploitation Android de Google, basé sur le logiciel IntelliJ IDEA de JetBrains et conçu spécifiquement pour le développement Android. Il permet de visualiser les changements en temps réels, et offre la possibilité de créer des émulateurs avec des différentes distributions.[12]

Pour utiliser Flutter sur Android, il faut d'abord ajouter le plugin flutter qui offre aussi d'autres avantages tels que :

- Dart analysis : fournit une bibliothèque qui effectue une analyse statique du code Dart. Il est utile pour l'intégration d'outils et l'incorporation.
- Dart Format : remplace les espaces blancs du programme par un formatage conforme aux directives de Dart. Il est appelé automatiquement lors de l'enregistrement.
- Flutter Outline : à chaque fois qu'un fichier change, un nouvel événement "Flutter Outline" est envoyé à Android Studio avec des informations structurées sur le contenu d'un fichier. Ces informations incluent les classes, les méthodes, les champs, ainsi que des informations sur les widgets Flutter créés dans le fichier.

- Flutter Inspector : c'est un outil puissant pour visualiser et explorer les arborescences de widgets Flutter. Le Framework Flutter utilise des widgets comme élément de base pour tout, des contrôles (tels que le texte, les boutons et les bascules) à la mise en page (telle que le centrage, le remplissage, les lignes et les colonnes).

- **Apache http Server**

Apache HTTP server est un logiciel de serveur HTTP, il permet d'interpréter plusieurs langages dont le PHP, Python, Perl. Il est le logiciel de serveur le plus populaire dans le World Wide Web.

- **Composer**

Composer est un logiciel de gestion des dépendances, il permet d'installer les bibliothèques nécessaires déjà répertorié dans un serveur tier.

- **Laravel**

Laravel est un Framework web reposant sur le PHP. Il suit le patron d'architecture MVC. Il utilise un marquage pour les vues nommé Blade. La gestion des bases de données se font à travers les migrations Ceci facilite la tâche pour un développement par équipe.

- **PHP runtime**

PHP runtime est un programme dont le rôle est d'exécuter un script PHP contenant les calculs et fonctions nécessaires, et ensuite, de générer une page au format HTML qui sera affichée dans un navigateur.

- **Visual Studio Code (VS Code)**

Visual Studio Code est un éditeur de code développé par Microsoft, il a le caractère d'être entièrement personnalisable par des extensions, il offre plusieurs options tel que la saisie automatique, configurer un débogueur ou choisir plusieurs environnements d'exécution. Il prend en charge la quasi-totalité des langages et marquages disponible.

- **Ubuntu Server 20.04**

Ubuntu Server est un système d'opération dédié pour les serveurs, il est caractérisé par sa légèreté et sa grande compatibilité par n'importe quelle configuration matérielle,

sa principale utilisation est de servir des sites web, partage de fichier, fournir un service cloud, etc. Ce système d'opération n'inclut que les logiciels nécessaires pour son fonctionnement, l'interaction se fait généralement en Terminal en utilisant le protocole SSH.

3.5.3. Fournisseurs de serveurs et d'hébergement

- **DigitalOcean**

DigitalOcean est un fournisseur de serveurs dédiés, il permet d'obtenir une infrastructure complète d'un ou plusieurs services, il se démarque de sa concurrence par sa facilité d'extensibilité par rapport au ressources demandé, qu'il s'agit de base de données, espace de stockage, ou puissance de calculs.

- **GitHub**

GitHub est un service d'hébergement de développement des logiciels qui appartient à Microsoft, il utilise le programme open source GIT pour la gestion des versions, il offre deux types de comptes : Gratuit pour le développement des logiciels libres, Payant pour des comptes professionnels. Il facilite la coopération entre une équipe des développeurs, Son caractéristique clé : il offre un pack étudiant qui comprend plusieurs services offerts par d'autre sociétés informatiques. Nous étions intéressés par l'offre qui nous propose un serveur privé par DigitalOcean.com. Ça a été l'un des outils les plus importants à l'aide au développement,

Apr 3, 2022 – Jun 24, 2022

Contributions: Commits ▾

Contributions to main, excluding merge commits and bot accounts

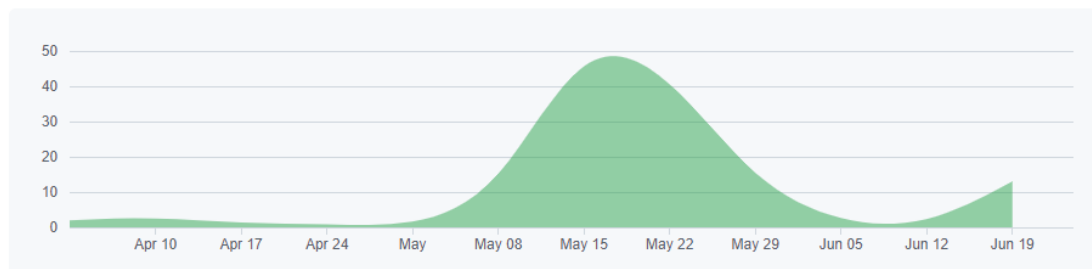


Figure 31 Graphe de contribution au git

3.5.4. Modèles de pages web

- **WonderTour**

Modèle de site web d'agence de voyage propre et simple, il fournit plusieurs outils et page prédéfinis. Et il est responsif.

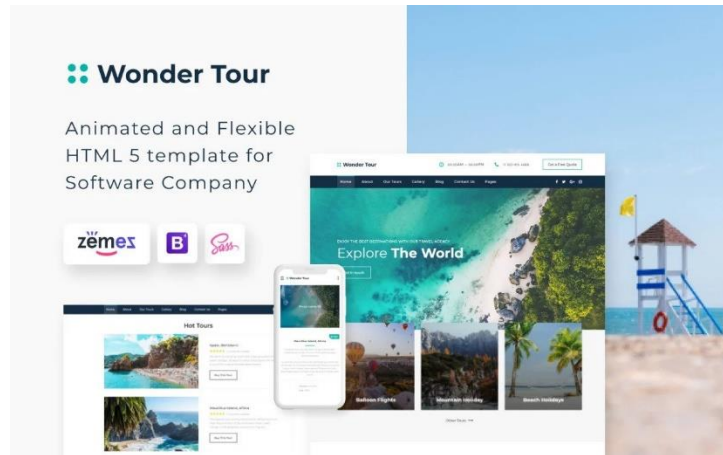


Figure 32: Modèle WonderTour

- **Argon Dashboard**

Argon Dashboard est un modèle de site web d'administration, il est compatible avec plusieurs environnements de travail tel que Laravel, il offre une version gratuite qui est fourni avec 70 composant individuel pour permettre une combinaison à la demande de l'utilisateur.

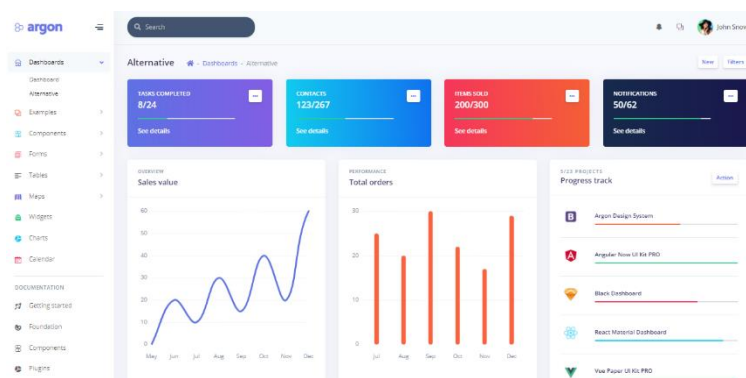


Figure 33: Argon Dashboard

3.6. Packages de flutter

Flutter prend en charge l'utilisation de packages partagés apportés par d'autres développeurs aux écosystèmes Flutter et Dart. Cela permet de créer rapidement une application sans avoir à tout développer à partir de zéro. Tout au long de ce projet, nous avons utilisé les packages suivants :

➤ **Animated Text Kit**

Un package flutter qui contient une collection d'animations de texte sympas et impressionnantes.

➤ **Carousel Slider**

Un widget de type Carousel, fournit les fonctionnalités suivantes : défilement infini, widgets enfants personnalisés et lecture automatique dans différents types de vue (avec ou sans indicateur, plein écran, etc.).

➤ **Chewie**

Le plugin `video_player` fournit un accès de bas niveau à la lecture vidéo. Chewie utilise le `video_player` sous le capot et l'enveloppe dans une interface utilisateur Material ou Cupertino conviviale.

➤ **Flutter Spinkit**

Une collection d'indicateurs de chargement animés de flutter.

➤ **Flutter TTs (text to speech)**

Plugin de synthèse vocale développé avec Swift, et offre les fonctionnalités suivantes : parler, arrêter, obtenir des langues, définir la langue, définir le volume, définir la hauteur de la parole, obtenir la voix, définir la voix.

➤ **Flutter polylines points**

Un plug-in flutter qui décode la chaîne de poly-lignes Google encodée en une liste de coordonnées géographiques adaptées à l'affichage d'itinéraires/poly-lignes sur des cartes.

➤ **Google Maps flutter**

Un plugin Flutter qui fournit un widget Google Maps avec ses caractéristiques.

➤ **Location**

Ce plugin pour Flutter gère l'obtention d'une position sur Android et iOS. Il fournit également des rappels lorsque l'emplacement est modifié.

➤ **Model Progress hud nsn**

Un widget parent simple pour activer la progression modale HUD (un indicateur de progression modale, HUD = Heads Up Display).

➤ **HTTP**

Une bibliothèque composable basée sur Future pour effectuer des requêtes HTTP. Ce package contient un ensemble de fonctions et de classes de haut niveau qui facilitent l'utilisation des ressources HTTP. Il est multiplateforme et prend en charge les mobiles, les ordinateurs de bureau et le navigateur.

➤ **RFlutter alert**

RFlutter Alert est une boîte de dialogue d'alerte/popup super personnalisable et facile à utiliser pour Flutter. Elle permet de créer facilement des styles d'alerte réutilisables ou d'ajouter autant de boutons que le développeur le souhaite.

➤ **Video Player**

Un plugin Flutter pour iOS, Android et web pour la lecture de vidéo sur une surface Widget.

Pour ajouter un package à une application, il faut suivre les étapes suivantes :

- Flutter pub add \$NAME (dans le terminal).
- Flutter pub get (dans le terminal).

- Importer le package.
- Utiliser le package.

Certains packages ont besoin d'autorisations depuis le téléphone pour fonctionner correctement et pour demander depuis l'application ou l'autorisation, certaines lignes doivent être ajoutées pour `manifest.xml` pour Android et `info.plist` pour iOS. Par exemple, pour ajouter une permission d'utiliser l'internet sur Android, il suffit d'ajouter à `manifest`:

```
<uses-permission android:name="android.permission.INTERNET"/>.
```

3.7. Services web

Selon la définition du W3C (World Wide Web Consortium), un web service (ou service web) est une application appelable via Internet par une autre application d'un autre site Internet permettant l'échange de données (de manière textuelle) afin que l'application appelante puisse intégrer le résultat de l'échange à ses propres analyses. Les requêtes et les réponses sont soumises à des standards (SOAP, WSDL et UDDI) et normalisées à chacun de leurs échanges.

Un service web complet est tout service qui est disponible sur Internet ou sur des réseaux privés (intranet), Autodescriptif via une grammaire XML ou JSON commune, Découvrable via un mécanisme de recherche simple, N'est lié à aucun système d'exploitation ou langage de programmation, et Utilise un système de messagerie XML/JSON standardisé.[12]

Tout au long du notre projet, nous avons utilisé les services web suivants :

Concernent google Map Android et distance, il faut d'abord avoir un compte sur google cloud. Ensuite, générer une clé privée au projet, activer les APIs nécessaire (directions API, places API, etc.), et enfin utiliser cette clé dans les requêtes HTTP.

➤ **Google Map distance Matrix API**

L'API Distance Matrix fournit la distance et le temps de trajet pour une matrice d'origines et de destinations, et se compose de lignes contenant des valeurs de durée et de distance pour chaque paire. Distance Matrix est disponible sous plusieurs formes :

- ❖ En tant qu'API autonome.
- ❖ Dans le cadre de l'API Maps JavaScript côté client.
- ❖ Pour une utilisation côté serveur dans le cadre des bibliothèques clientes pour les services web Google Maps.

L'exemple suivant illustre une requête de données de distance entre Tlemcen et Maghnia :

https://maps.googleapis.com/maps/api/directions/json?origin=Maghnia&destination=Tlemcen&key=*****

La clé API a été masquée pour des raisons de sécurité (google cloud console nécessite une carte de crédit, bien qu'il soit possible de profiter de certains services gratuitement).

L'API renvoie des informations basées sur l'itinéraire recommandé entre les points de départ et d'arrivée. Aussi, il est possible de demander des données de distance pour différents modes de déplacement, demander des données de distance dans différentes unités telles que des kilomètres ou des miles, et estimer le temps de trajet dans le trafic.[13]

Exemple d'une partie de réponse au format JSON est fourni ci-dessous :

```
{
  "distance": {
    "text": "0,2 km",
    "value": 161
  },
  "duration": {
    "text": "1 minute",
    "value": 18
  },
  "end_location": {
    "lat": 34.9081917,
    "lng": -1.6418337
  },
  "html_instructions": "Au rond-point, prendre la <b>2e</b> sortie",
  "maneuver": "roundabout-right",
  "polyline": {
    "points": "c~`tEfm_IMMSGK@OFMJCBCN?LBPDNLZJTD^A\Fr@"
  },
  "start_location": {
    "lat": 34.9080209,
    "lng": -1.6406835
  },
  "travel_mode": "DRIVING"
},
```

Figure 34: Google Maps API distance exemple

➤ Google Maps API

Avec le SDK Maps pour Android et iOS, il est possible d'ajouter des cartes basées sur les données Google Maps à l'application. Le SDK gère automatiquement l'accès aux serveurs Google Maps, l'affichage de la carte et la réponse aux gestes de l'utilisateur tels que les clics et les glissements. Aussi, il permet également d'ajouter des marqueurs, des poly-lignes, des superpositions au sol et des fenêtres d'informations à la carte. Ces objets fournissent des informations supplémentaires sur les emplacements de la carte et permettent une interaction de l'utilisateur avec la carte.

➤ OpenWeatherMap

OpenWeatherMap est un service en ligne, détenu par OpenWeather Ltd, qui fournit des données météorologiques mondiales via une API, y compris des données météorologiques actuelles, des prévisions, des prévisions immédiates et des données météorologiques historiques pour n'importe quel emplacement géographique. La société fournit des prévisions de précipitations hyperlocales minute par minute pour n'importe quel endroit.[14] Pour faire un appel API, il faut utiliser un URL similaire à l'exemple suivant :

<https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}&units=metric>.

3.8. Déploiement d'application web sur Serveur

3.8.1. Prérequis

- Une machine connectée à un réseau internet tournant sous Ubuntu server.
- Apache Web Server.
- PHP Runtime 7.4.
- Composer.
- Git.
- MySQL Server.
- Client SSH.

3.8.2. Etapes d'installation

- Accéder ou connecter via SSH au terminal du serveur.
- Installez les dépendances nécessaires :

```
> sudo apt update
> sudo apt install apache2 mysql-server wget php libapache2-mod-
php php-mbstring php-xmlrpc php-soap php-gd php-xml php-cli php-
zip php-bcmath php-tokenizer php-json php-pear php-curl
> sudo wget -O composer-setup.php
https://getcomposer.org/installer
> sudo php composer-setup.php --install-dir=/usr/local/bin --
filename=composer
```

- Configurez le pare-feu pour autoriser l'accès à Apache :

```
> sudo ufw allow "Apache Full"
```

- Créer un répertoire pour l'installation de l'application via la commande :

```
> mkdir /var/www/html/{{NOM REPERTOIRE}}
```

- Cloner le programme depuis le git en utilisant les commandes :

```
> cd /var/www/html/{{NOM REPERTOIRE}}
> sudo git remote add origin {{LIEN GIT}}
> git branch -M main
> git pull origin main
```

- Installer les composants requis avec composer via :

```
> composer install
```

- Créer et éditer le fichier environnement de Laravel :

```
> sudo nano .env
```

- Copier le modèle en ajustant les paramètres selon votre environnement :

```
APP_NAME={{NOM D'APPLICATION}}

APP_ENV=production

APP_KEY=

APP_URL={{NOM DE DOMAINE DE L'APPLICATION}}

DB_CONNECTION=mysql

DB_HOST= {{ADRESSE BASE DE DONNES}}

DB_PORT= {{PORT DE LA BASE DE DONNEES}}

DB_DATABASE= {{NOM DE LA BASE DE DONNEE}}

DB_USERNAME= {{NOM D'UTILISATEUR DE LA BDD}}

DB_PASSWORD= {{MOT DE PASSE DE LA BDD}}
```

- Générer une clé pour votre projet :

```
> php artisan key:generate
```

- Exécuter les migrations pour les données et la structure de votre base de données :

```
> php artisan migrate
> php artisan db:seed
```

- Créer un nouveau fichier de configuration pour Apache via les commandes :

```
> cd /etc/apache2/sites-available
> sudo nano laravel_project.conf
```

- Ajouter et éditer le modèle ci-dessous pour le fichier de configuration en remplaçant :
 - {{ADRESSE_SERVEUR}} par l'adresse IP du serveur
 - {{EMAIL_WEBMASTER}} par l'adresse email du webmaster.
 - {{NOM_REPERTOIRE}} par le nom du répertoire choisi.
 - {{DOMAINE_SERVEUR}} par le nom de domaine du serveur.

```
<VirtualHost *:80>
    ServerName {{ADRESSE DU SERVEUR}}
    ServerAdmin {{EMAIL WEBMASTER}}
    DocumentRoot /var/www/html/{{NOM REPERTOIRE}}/public

    <Directory /var/www/html/{{NOM REPERTOIRE}}/public>
        AllowOverride All
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
        Require all granted

    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    RewriteEngine on
    RewriteCond %{SERVER_NAME} ={{NOM DE DOMAINE DU SERVEUR}} [OR]
    RewriteCond %{SERVER_NAME} =www.{{NOM DE DOMAINE DU SERVEUR}}
    [OR]
    RewriteCond %{SERVER_NAME} ={{ADRESSE_SEREVEUR}}
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
    [END,NE,R=permanent]
```

```
</VirtualHost>
```

- Activer la configuration de l'application tout en désactivant la configuration par défaut

```
> sudo a2dissite 000-default.conf  
  
> sudo a2ensite laravel_project
```

- Donner les autorisations requises a l'application

```
> sudo chown -R www-data:www-data /var/www/{{NOM REPERTOIRE}}/  
  
> sudo chmod -R 755 /var/www/{{NOM REPERTOIRE}} /  
  
> sudo a2enmod rewrite
```

- Redémarrer le service apache pour prendre en charges toutes les actions requises

```
> sudo systemctl restart apache2.service
```

- Vous pourriez accéder à votre application web en navigant vers l'adresse du serveur ou le nom de domaine attribué.

Le site web est accessible sur le lien <https://pfeweb.ml/>.

3.9. Conclusion

Ce dernier chapitre représente la conclusion de notre projet. Dans ce chapitre nous avons présenté brièvement le processus de réalisation de nos applications en spécifiant l'environnement, les outils, les protocoles, les marquages, les langages de programmations, les packages, le serveur et les services web associé à notre système En effet, nous avons achevé l'implémentation tout en respectant la conception élaborée.

Conclusion Générale

Pour conclure, ce travail, réalisé dans le cadre du projet de fin d'études, en vue de l'obtention d'un master en Génie Logiciel, est une mise à disposition d'un outil de tourisme web et mobile qui permet aux visiteurs, tout au long de leur visite, de faciliter et accélérer la consultation et la localisation des lieux touristiques.

Au cours de ce projet, nous avons présenté les différentes étapes du processus de développement. Nous avons d'abord exposé le contexte, le problème et ensuite proposé une solution intelligente. Ensuite, une analyse concurrentielle a été mise en place entre deux applications similaires à notre système. Ensuite, nous avons développé une charte graphique et décrit certains aspects de l'UI.

L'étape suivante a été l'identification des besoins que nous avons ensuite modélisés sous forme de diagrammes de cas d'utilisation, diagrammes de séquence, diagramme de classes et terminé par la réalisation de notre application.

La création d'un chemin respectant certains critères était la caractéristique la plus complexe de la conception, la complexité étant de décider lequel de ces critères serait pris en compte en premier puis le second, etc., une méthode appelée « Voyageur de commerce » a été utilisée pour améliorer cette fonctionnalité.

Lors de ce travail, nous avons pu mettre en pratique nos connaissances théoriques acquises lors de notre formation.

Ce projet a été une expérience intéressante. Il nous a permis de comprendre que le projet de création d'une application mobile et web nécessite un ensemble de plusieurs actions planifiées et dépendantes les unes des autres.

Nous retenons également l'utilité des méthodes et des outils de conception modernes pour faciliter et accélérer le travail et organiser efficacement les groupes, afin de fournir un produit professionnel et moderne. D'un autre côté, nous avons appris que nous pouvons déduire le temps et les ressources nécessaires à la réalisation d'un projet informatique comme celui-ci.

Une amélioration qui nous semblerait appropriée à ce projet serait peut-être en termes de design (ergonomie), et certaines fonctionnalités telles que les notifications à

l'approche d'un site touristique, la synthèse vocale pour effectuer des actions, l'ajout d'une section pour les commentaires sur un lieu et même déployer l'application mobile au niveau du Play store et de l'App store afin de pouvoir l'exploiter.

Bibliographie

- [1] IDVizit - idvizit.com -consulté le 05/03/2022
- [2] ID-Vizit, Play Store,
<https://play.google.com/store/apps/details?id=fr.xpdigit.apptourism>,consulté le 05/03/2022
- [3] The Culture Trip - theculturetrip.com/our-app/- consulté le 05/03/2022
- [4] Culture Trip: Book Travel, Play Store,
<https://play.google.com/store/apps/details?id=culturetrip.com> consulté le 05/03/2022
- [5] Architecture Modèle/View/Contrôleur, Institut de recherche en informatique fondamentale,
www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html,
consulté le 10/06/2022
- [6] MVC - Mozilla Developpers Docs - developer.mozilla.org/fr/docs/Glossary/MVC -
consulté le 10/06/2022
- [7] Rayed Benbrahim, MVC : le pattern populaire pour une application web ?,
PracticalProgramming, practicalprogramming.fr/mvc/, consulté le 10/06/2022
- [8] GIMP, Wikipedia, fr.wikipedia.org/wiki/GIMP, consulté le 13/06/2022
- [9] Flutter (Software), Wikipedia [en.wikipedia.org/wiki/Flutter_\(software\)](http://en.wikipedia.org/wiki/Flutter_(software)) , consulté le 14/05/2022
- [10] Aphiniya Dechalert, Flutter vs React Native, Tabnine,
www.tabnine.com/blog/flutter-vs-react-native/ , consulté le 14/05/2022
- [11] WikiTravel, Wikipedia en.wikipedia.org/wiki/Wikitavel, consulté le 13/06/2022
- [12] WikiVoyage, Wikipedia en.wikipedia.org/wiki/WikiVoyage, consulté le 13/06/2022
- [13] Architecture MVC : qu'est ce que c'est? son fonctionnement, Rose Dieng Lab
rosedienglab.defarsci.org/a-quoi-sert-une-architecture-mvc-son-fonctionnement/
consulté le 10/06/2022
- [14] Difference between Flutter and React native, JavaAtPoint, javatpoint.com/flutter-vs-react-native , consulté le 01/03/2022

Résumé

Le but de ce travail est de créer une application web et mobile fiable, ergonomique qui offre les services nécessaires et adéquats aux besoins des touristes avec une utilisation simple. L'utilisateur peut profiter des informations fournies telles que des données sur les lieux touristiques accompagnées d'une galerie et du chemin y menant. Aussi, il peut générer des itinéraires selon certains critères (distance, durée, prix, catégorie) et les enregistrer. De plus, côté administrateur, c'est lui qui a le rôle de gestion (ajout/modification, suppression) des villes, catégories, sites touristiques et de leurs galeries. L'application a d'abord été conçue graphiquement avec l'outil Figma qui propose des interfaces interactives selon les standards UX/UI. Sa conception UML a été réalisée avec Modelio pour les diagrammes de cas d'utilisation, de séquence et de classes. Enfin, l'implémentation du code source a été réalisée avec le Flutter Framework de google accompagné du langage Dart pour le mobile et du Laravel Framework pour le web. Et ça c'est avec l'IDE Android Studio, Visual Studio Code, le serveur apache, le déploiement sous DigitalOcean avec SSH, sans oublier les packages Flutter et quelques services web.

Mots-clés : Gestion touristique, génération de parcours touristiques, application mobile, mobile tourisme.

ملخص

يتمثل الهدف من هذا العمل في خلق تطبيق صالح للاستعمال على شبكة الويب وفي نفس الوقت على الهاتف المحمول، حيث يكون هذا التطبيق موثوق ومريح في استعماله، ويقدم الخدمات الضرورية والملائمة التي تتماشى وحاجات السائح، باستخدام سهل. حيث يستطيع المستخدم الاستفادة من المعلومات المقدمة، كاليانات المتعلقة بالمواقع السياحية والمرفوقة بمعرض للصور إضافة إلى المسار المؤدي إليها. ناهيك عن قدرته على تشكيل مسارات وفقاً لمعايير معينة (المسافة، المدة، السعر، الفئة) مع إمكانية حفظها. ومن ناحية أخرى؛ يتولى المسؤول على التطبيق دور إدارة (إضافة، تعديل، حذف) المدن، الفئات، المواقع السياحية، ومعارض الصور الخاصة بهم.

والتي تقدم واجهات تفاعلية وفقاً لمعايير الـ UX/UI.

كلمات مفتاحية:

إدارة السياحة، وتوليد الطرق السياحية، وتطبيقات الهاتف المحمول، والسياحة المتنقلة

Abstract

The purpose of this work is to create a reliable platform that offers the necessary and adequate services to the needs of tourists with simple use. The user can enjoy the information provided such as data on the tourist places accompanied by a gallery and the path leading there. Also, it can generate routes according to certain criteria (distance, duration, price, category) and save them. In addition, on the administrator side, it is he who has the role of managing (adding/modifying, deleting) cities, categories, tourist sites and their galleries. The application was first designed graphically with the Figma tool, which offers interactive interfaces according to UX/UI standards. Its UML design was done with Modelio case, sequence and class diagrams. Finally, the implementation of the source code was carried out with the Flutter Framework from google accompanied by the Dart language for the mobile and the Laravel Framework for the web. And that's with the Android Studio IDE, Visual Studio Code, the Apache server, the deployment under DigitalOcean with SSH, not to mention the Flutter packages and some web services

Keywords: Tourism management, generation of tourist routes, mobile application, mobile tourism.