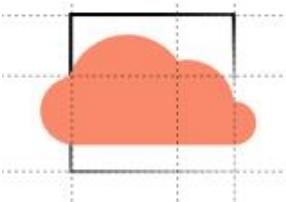


REPUBLIQUE DU CAMEROUN

Paix-Travail-Patrie

INSTITUT AFRICAIN D'INFORMATIQUE



Clouds Architects

Email : contact@cloudsarchitects.com

Tel (237) 651542854

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

AFRICAN INSTITUTE OF COMPUTER SCIENCES



Etablissement inter-états d'enseignement supérieur

Email : contact@iaicameroun.com

Tel : (237) 22729928

RAPPORT DE STAGE ACADEMIQUE

THEME : Mise en place d'une plateforme web et mobile de réservation de billets de transport

Stage effectué du 27 Juin au 29 Septembre 2023 en vue de l'obtention du Diplôme de Technicien Supérieur (DTS)

Rédigé et soutenus par :

FOGUE Tiagho Arsene Nelson

Élève ingénieur en travaux informatiques option Génie Logiciel

SOUS L'ENCADREMENT

Académique de :

Fenkam Jules Blaise

(Administrateur et intégrateur des SI)

Professionnel de :

Johan Kenneth Nken

(Architecte Salesfoce)

+237 694 30 17 30

Année Académique : 2022-2023

ANNÉE ACADEMIQUE : 2022-2023

DÉDICACE



À mes parents
Tiagho Norbert
Et
Kengni Madeleine

REMERCIEMENTS

Le présent document, est le fruit d multiples efforts; mais également, du soutien et des conseils de nombreuses personnes à qui nous souhaitons adresser notre gratitude:

- A Dieu Tout puissant,
- A monsieur **Armand Claude Abanda**, représentant résident de l'IAI-Cameroun, qui se bat quotidiennement pour que nous ayons une formation de qualité.
- A **Monsieur Serge Ymelé** pour l'aide précieuse qu'il n'a cessé de nous apporter durant notre parcours académique au sein de l'IAI-Cameroun centre de Bafoussam.
- A notre Encadreur académique **Monsieur Fenkam Jules Blaise** pour sa disponibilité, ses conseils et son savoir apportés dans la réalisation de notre projet.
- A **Monsieur Johan Kenneth Nken**, Directeur général de Clouds Architects et également notre encadreur professionnel, pour son accueil dans sa structure, et ainsi disponibilité et son suivie de notre projet.
- A **tous les enseignants et à tout le corps administratif** de l'IAI Bafoussam pour leur encadrement durant notre parcours.
- A **toute la famille TIAGHO** pour son accompagnement, ses et son soutien inconditionnel et sans faille.
- A **toute l'équipe de Clouds Architects** pour son soutien et ses conseils.
- A nos camarades de promotions qui ont participé à notre épanouissement au sein de l'IAI- Cameroun.
- A tous ceux et celles ayant participé de près ou de loin à la réalisation de ce projet.

SOMMAIRE

DÉDICACE	2
REMERCIEMENTS	i
SOMMAIRE	iii
LISTE DES FIGURES	iv
LISTE DES TABLEAUX.....	v
GLOSSAIRE.....	vi
RÉSUME	vii
ABSTRACT	viii
INTRODUCTION GÉNÉRALE	1
PARTIE 1 :PHASE D'INSERTION	2
CHAPITRE 1 : ACCEUIL EN ENTREPRISE	3
CHAPITRE 2 : PRÉSENTATION DE L'ENTREPRISE.....	6
PARTIE 2 : PHASE TECHNIQUE.....	8
DOSSIER 1 : L'EXISTANT	9
DOSSIER 2 : CAHIER DES CHARGES.....	12
DOSSIER 3 : DOSSIER D'ANALYSE.....	18
DOSSIER 4 : DOSSIER DE CONCEPTION	33
DOSSIER 5 : DOSSIER DE REALISATION	37
DOSSIER 6 : TESTS DE FONCTIONALITÉS.....	51
DOSSIER 7 : GUIDE D'INSTALLATION ET GUIDE UTILISATEUR	52
CONCLUSION GÉNÉRALE.....	56
ANNEXE 1 : BIBLIOGRAPHIE	57
ANNEXE 2 : WEBOGRAPHIE	58
ANNEXE 3 : TABLE DES MATIERES.....	59

LISTE DES TABLEAUX ET FIGURES

❖ LISTE DES TABLEAUX

Tableau 0 : fiche d'identification de l'entreprise	3
Tableau 01 : équipe de travaille	15
Tableau 02 : ressource matérielle	16
Tableau 03 : ressources humaines.....	16
Tableau 04 : Total	16
Tableau 05 : planifications de taches	15
Tableau 06 : Contrainte du projet	17
Tableau 07 : formalisme digramme de cas d'utilisations	Erreur ! Signet non défini.
Tableau 08 : liste des cas d'utilisations.....	Erreur ! Signet non défini.
Tableau 09 : cas d'utilisation du changement de statut d'un utilisateur	28
Tableau 10 : cas d'utilisation de création d'un utilisateur	29
Tableau 11 : cas d'utilisation d'authentification d'un utilisateur	29
Tableau 12 : cas d'utilisation effectuer une collecte d'un utilisateur	30
Tableau 13 : cas d'utilisation de consulter le solde d'un utilisateur	31
Tableau 14 : liste des attributs	38
Tableau 15 : formalisme du diagramme de séquences	Erreur ! Signet non défini.
Tableau 16 : les cardinalités.....	40
Tableau 17 : Matériels de base.....	Erreur ! Signet non défini.

❖ LISTE DES FIGURES

Figure 0 : schéma de la méthode 2TUP	24
Figure 01 : cas d'utilisation du client.....	27
Figure 02 : cas d'utilisation du collecteur	27

Figure 03 : cas d'utilisation de l'administrateur	28
Figure 04 : cas d'utilisation Gobal.....	Erreur ! Signet non défini.
Figure 05 : diagramme de séquence d'une collecte	Erreur ! Signet non défini.
Figure 06 : diagramme de séquence de la création d'un utilisateur par l'administrateur	Erreur ! Signet non défini.
Figure 07 : diagramme de séquence d'authentification	Erreur ! Signet non défini.
Figure 08 : diagramme de Classe.....	41
Figure 09 : logo XML	Erreur ! Signet non défini.
Figure 10 : logo Java.....	Erreur ! Signet non défini.
Figure 11 : logo Firebase	Erreur ! Signet non défini.
Figure 12 : logo React JS	Erreur ! Signet non défini.
Figure 13 : logo JavaScript	Erreur ! Signet non défini.
Figure 14 : logo HTML.....	Erreur ! Signet non défini.
Figure 15 : logo Sass.....	Erreur ! Signet non défini.
Figure 16 : logo GitHub	Erreur ! Signet non défini.
Figure 17 : logo node js	Erreur ! Signet non défini.
Figure 18 : logo PhotoShop	Erreur ! Signet non défini.
Figure 19 : logo Microsoft Word.....	Erreur ! Signet non défini.
Figure 20 : logo Microsoft PowerPoint	Erreur ! Signet non défini.
Figure 21 : représentation de la collection des utilisateur.....	Erreur ! Signet non défini.
Figure 22 : représentation de la collection des Comptes	Erreur ! Signet non défini.
Figure 23 : représentation de la collection des Transaction.....	Erreur ! Signet non défini.
Figure 24 : règle de création d'un document	Erreur ! Signet non défini.
Figure 25 : Diagramme de déploiement.....	Erreur ! Signet non défini.

GLOSSAIRE

- **SGBD** : Système de Gestion des Bases de Données
- **SQL**: Structured Query Language
- **IDE** : integrated development enviroment (environnement de développement intégré).
- **IAI** : Institut africain d'informatique.
- **DTS** : Brevet de Technicien Supérieur
- **UML**: Unified Modeling Language
- **FAQ** : foire aux questions
- **UI** : User interface (présentation visuel d'une application souvent assimilé au plaisir de regarder l'application);
- **UX** : User expérience (Sentiment de l'utilisateur lors de l'utilisation du produit ; plaisir, simplicité lors de l'utilisation d'un produit).
- **UP** : Unified Process
- **USDP** : Unified software development Process
- **PU** : Processus unifiés
- **POO** : Programmation orientée objet

RÉSUMÉ

Ce rapport présente le travail réalisé dans le cadre de notre stage académique en vue de l'obtention d'un DTS à l'IAI Cameroun. L'objectif principal était de fournir aux utilisateurs une solution pratique et conviviale pour réserver leurs billets de transport en ligne. La problématique abordée concernait la complexité et l'inefficacité du processus de réservation traditionnel, qui nécessitait souvent des déplacements physiques ou par appel, et des paiements en espèce. L'approche choisie pour résoudre cette problématique a consisté à développer une plateforme web et mobile permettant aux utilisateurs de réserver et payer leurs billets de manière sécurisée et pratique. Le travail abattu comprenait plusieurs étapes; notamment, l'analyse des besoins utilisateurs, la conception de l'architecture du système, le développement de l'interface utilisateur, l'intégration avec des services de paiement en ligne et la mise en place d'une base de données pour gérer les réservations. Des tests approfondis ont également été effectués pour garantir la qualité et la fiabilité du système. Les résultats obtenus ont été satisfaisants. La plateforme a été déployée avec succès, offrant aux utilisateurs la possibilité de rechercher, réserver et payer leurs billets de transport de manière simple. Les retours d'utilisateurs ont été positifs, mettant en évidence la simplicité d'utilisation et la commodité de la plateforme. En conclusion, la mise en place d'une plateforme web et mobile de réservation a permis de résoudre la problématique du processus de réservation traditionnel et contraignant. Le système développé offre une solution conviviale en améliorant l'expérience des utilisateurs et en facilitant la gestion des réservations. Il ouvre également des perspectives de développement futures; telles que l'intégration de fonctionnalités supplémentaires (un système de fidélité, un plus large éventail de moyens de paiement etc.) ; ainsi qu'une expansion à d'autres moyens et voies de transport.

ABSTRACT

This research project focuses on the implementation of a web and mobile platform for interurban ticket reservations. In the context of increasing demand for interurban transportation services, there is a need for an efficient and convenient ticketing booking system. The central problem addressed by this research is the development of a user-friendly platform that allows customers to easily book interurban transport tickets through a web or a mobile interface. Previous research has emphasized the importance of accessible ticket reservation systems in enhancing customer satisfaction and streamlining the ticketing progress. The goal of this research is to fill the existing gap in the market by creating a comprehensive platform specifically designed for interurban ticket reservations, improving convenience, accessibility for both customers and service providers; To address this question, we have employed a comprehensive methodology; Market research to understand the current landscape of the context and identify user preferences and pain points; software development methodologies and user-centered design principles, resulting with a seamless interface, secure payment integration, fast ticket availability updates, and efficient customer support. The main finding has been the successful implementation of the platform, demonstrating its effectiveness and usability through user testing and feedback. Overall, this research addresses the pressing need for an optimized reservation system in the interurban transportation sector, enhancing the overall travel experience.

INTRODUCTION GÉNÉRALE

Dans notre deuxième année de cursus à l'IAI, nous, étudiants sommes envoyés en stage académique afin de mettre en pratique des compétences professionnelles et de réaliser un projet résolvant un problème précis de la société. Le nôtre fut effectué au sein de l'entreprise Clouds Architects avec comme ligne directrice la "Conception et réalisation d'une plateforme web et mobile d'achat de billets de transport interurbain". Ce projet s'inscrit dans un contexte où les besoins en matière de mobilité sont en constante évolution créant ainsi une certaine latence. La digitalisation des services de transport s'impose donc comme un enjeu majeur. L'objectif étant de créer une interface conviviale, intuitive et sécurisée, permettant aux utilisateurs de rechercher, réserver et acheter des billets de transport de manière simple. Pour mener à bien ce projet, nous suivrons un processus structuré comprenant plusieurs étapes. Dans un premier temps, nous procéderons à une analyse approfondie des besoins et des attentes des utilisateurs ainsi que des contraintes techniques et opérationnelles. Ensuite, nous élaborerons une stratégie de conception en définissant les fonctionnalités clés, l'architecture de la plateforme et les technologies à utiliser. Par la suite, nous entamerons le développement de la plateforme en mettant en œuvre les différentes fonctionnalités et en assurant leur intégration harmonieuse. Enfin, nous réaliserons des tests approfondis pour garantir la fiabilité, la performance de la plateforme, suivis d'une phase de déploiement et de suivi post-lancement. Ce rapport de stage présentera en détail l'ensemble de ces étapes, en mettant l'accent sur les choix de conception, les défis rencontrés, les solutions développées, ainsi que les résultats obtenus.

PARTIE 1 : RAPPORT D'INSERTION

INTRODUCTION

Arrivé en entreprise le 27 Juin 2023, avons été accueillie par le directeur général Johan Kenneth Nken, en qualité de stagiaire. Dans les lignes qui suivent nous présenterons dans un premier temps la structure, et ensuite le déroulement du stage.

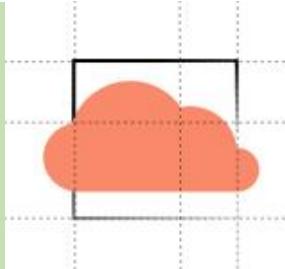
CHAPITRE 1 : PRESENTATION DE L'ENTREPRISE

I- PRESENTATION GENERALE DE LA STRUCTURE

I.1) Historique

Clouds architects est une entreprise existant depuis 2022 mais, officiellement créée le 22 février 2023 par Monsieur JOHAN KENNETH Nken, elle exerce principalement dans la formation des jeunes aux métiers du web et du Cloud, mais également à la digitalisation des processus métier. Son but premier est de créer une communauté de professionnels ayant des compétences sur les métiers du Cloud.

Tableau 1 : fiche d'identification de l'entreprise

Raison sociale	Clouds Architects
Logo	
Contact	652 54 28 54
Date de création	22 Février 2023
Siège social	Yaoundé Cameroun
Responsable	JOHAN KENNETH NKEN
Capital social	100 000 FCFA
Page Facebook	CloudArchitects
Coordonnées géographiques	3.854869, 11.479732
Site web	http://www.cloudsarchitects.com

I.2) Situation géographique

L'entreprise Clouds architects est située dans la ville de Yaoundé Cameroun, au lieu-dit « dépôt de boisson, centre) dans le quartier d'ETOUG EBE.

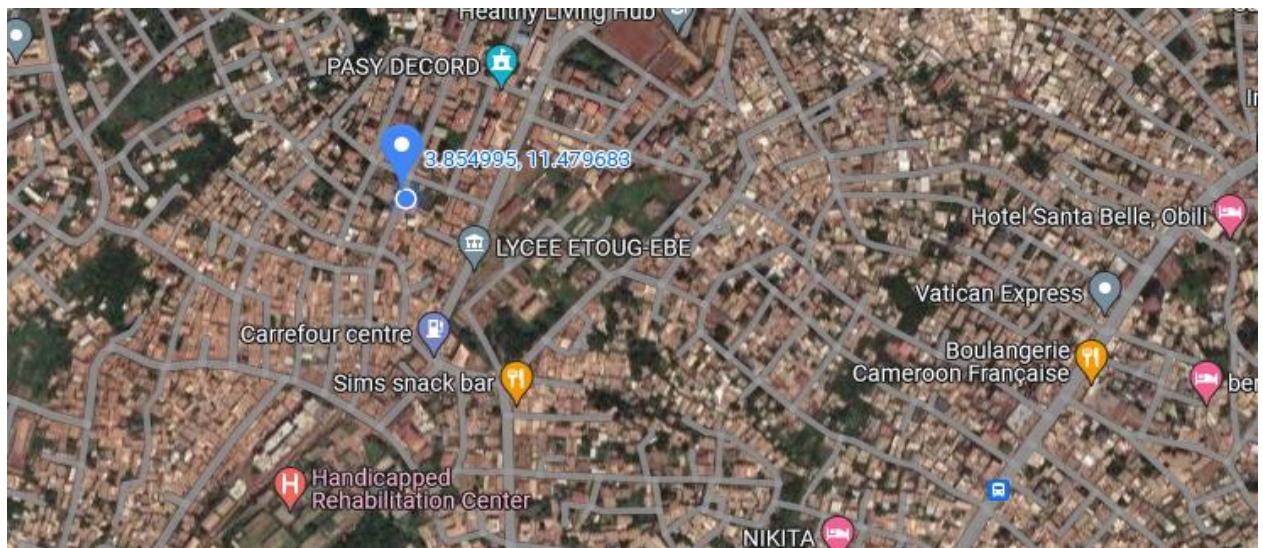


Figure 1: Situation géographique de Clouds architects

I.3) Valeurs

L'entreprise Clouds architects prône plusieurs valeurs au rang desquels :

- ✓ Respect
- ✓ Confiance
- ✓ Travail
- ✓ Collaboration
- ✓ Succès Client

II- MISSIONS ET ORGANES DE GESTION

II.1) Services

Les services offerts par celle-ci sont nombreux à savoir :

- ❖ Création de sites web.
- ❖ Création d'applications mobiles.
- ❖ Création d'instances salesforce pour la gestion de relations clients.

Pour mieux servir ces clients l'entreprise est ouverte de lundi à vendredi de 8h00 à 17H00.

II.2) Missions

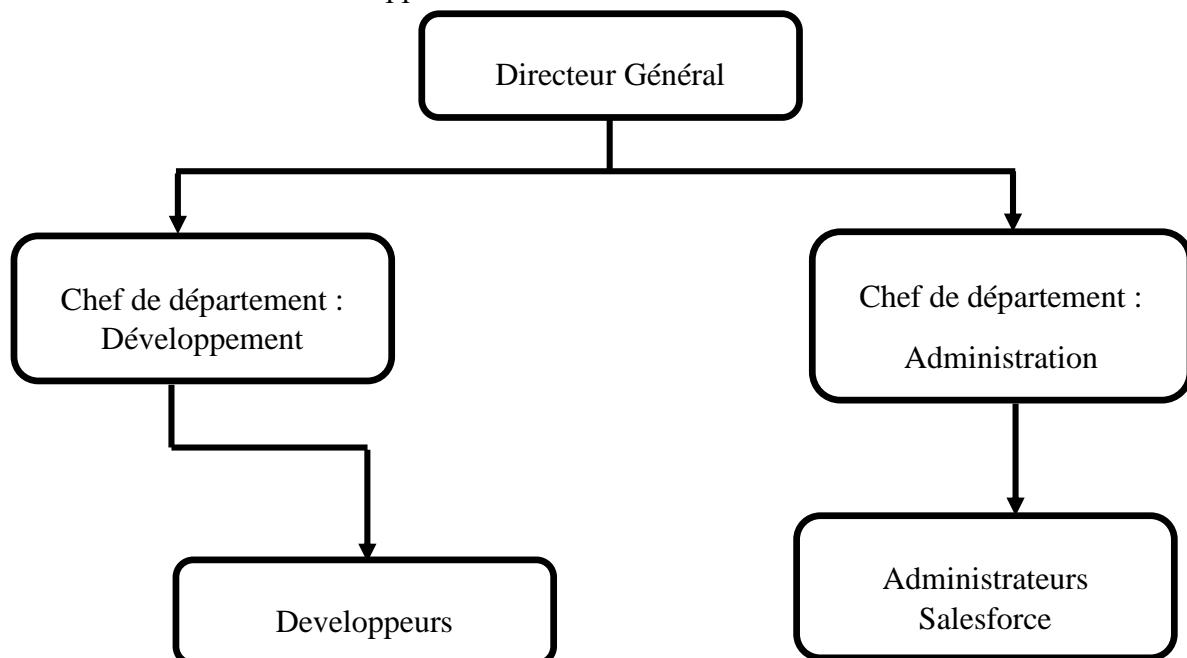
Vision : Devenir le meilleur fournisseur de solutions basées sur le Cloud et le meilleur dans l'accompagnement des entreprises dans leur transition digitale dans la sous-région Afrique centrale :

- ❖ créer une communauté de professionnels ayant des compétences sur les métiers du Cloud
- ❖ Former au moins de 500 personnes aux technologies du web et Cloud dans les 5 premières années.
- ❖ créer un marché à fort potentiel de professionnels du web et du Cloud dans la sous-région Afrique Centrale

II.3) Organes de gestion

Pour le bon fonctionnement de l'entreprise, elle possède comme organe de gestion :

- ❖ Le directeur général : qui est chargé de coordonner les opérations sur les différents projets.
- ❖ Chefs de département : administration
- ❖ Chef de département : Développeur
- ❖ Développeurs : apprennent et travaillent sur du développement Salesforce.
- ❖ Administration : Apprennent et travaillent sur de l'administration Salesforce.



CHAPITRE 2 : DÉROULEMENT DU STAGE

I- ACCUEIL

Mardi le 27 juin 2023 marque le début de notre stage à la startup Clouds Architects devant se poursuivre jusqu'au 30 Septembre 2023. Les premières semaines ont été marquées par plusieurs étapes parmi lesquelles :

- ✓ la prise en contact avec les membres du personnel
- ✓ La présentation du fonctionnement interne de l'entreprise et des technologies utilisées !
- ✓ La présentation des différentes contraintes de l'entreprise notamment les heures d'arrivée, de départ, de pause, des meetings en début et en fin de journée.
- ✓ L'inscription sur la plateforme Trailhead, dédiée à l'apprentissage des outils et technologies Salesforce.

L'on nous a également exhortés à nous mettre au travail dès le début afin de garantir un stage instructif et constructif !

II- TÂCHES EFFECTUÉES

Nous avons au sein de Clouds Architects effectué plusieurs tâches durant notre stage, aux rangs desquelles :

- ❖ Réalisation de défis et résolution de problèmes sur la plateforme Trailhead.
- ❖ Installation d'utilitaire sur des postes défectueux,
- ❖ Participation à la conception de sites web,
- ❖ Aménagement de notre espace de travail

III- ÉVALUATION

III.1) Difficultés

Le plus difficile lors de notre stage était les soucis rencontrés avec les coupures récurrentes d'électricité incluant une perte de connexion à internet. Nous avons également fait face à un pc parfois défaillant nécessitant des réinitialisation.

III.2) Bilan de l'expérience

Tout d'abord, il est important de noter que notre stage s'est déroulé dans de bonnes conditions. Grâce à une équipe dynamique, compétente et surtout sociable, notre intégration dans le groupe s'est faite non pas sans difficultés mais sans problème majeur.

Pour mener à bien notre stage, il a fallu que nous développions certaines attitudes notamment

- ❖ La sociabilité pour pouvoir créer et entretenir une bonne relation de travail avec les employés de la structure.
- ❖ Le développement de l'écoute; capitale en entreprise afin de produire un résultat fidèle aux souhaits des clients, et aux préoccupations de nos collègues.
- ❖ L'humilité de demander de l'aide quand cela est nécessaire et la facilité de porter assistance au mieux de nos compétences à tous les utilisateurs dans le besoin.

Conclusion

L'entreprise clouds architects exerçant dans la digitalisation des processus métiers, et dirigée par monsieur Johan Kenneth nken nous a accueillis dans ses locaux pour notre stage le 27 Juillet 2023; Elle nous a également accompagné dans les différentes étapes de la conception à la réalisation de notre projet. Etape que nous présenterons dans les prochains dossiers.

PARTIE 2 : PHASE TECHNIQUE

INTRODUCTION

La phase technique de notre rapport servira de repère, pour les étapes théoriques et concrètes dans la réalisation de notre projet, elle permettra de mettre l'accès sur les différents modèles, les différents diagrammes, et les différentes ressources mises en œuvre sur les pour réaliser notre projet.

DOSSIER 1 : L'EXISTANT

Introduction

Cette partie vise à présenter notre thème de manière générale mais également contextuelle, d'en faire une étude situationnelle, et enfin d'en faire une critique constructive, afin de proposer une solution optimale à la problématique posée.

I- PRÉSENTATION DU THÈME

Notre thème, mise en place d'une plateforme web et mobile de réservation de billets de transport urbain, a pour but de rendre plus autonome le système de transport interurbain actuel et le digitaliser d'avantage, en mettant sur pieds une plateforme qui regroupe les différents voyages créés par les agences de transport accessible aux utilisateurs, et leur permettant de réserver de manière simple et agréable leurs tickets de bus.

II- ÉTUDE DE L'EXISTANT

On peut procéder de plusieurs façons pour décrire le système actuel. Mais, que ce soit en interrogeant les clients, les fournisseurs ou même en nous rendant sur place, on retrouve à peu près les mêmes réponses.

Dans l'existant, les agences sont dissociées les unes des autres et utilisent des systèmes isolés, et centralisés associés à des générateur de facture impliquant donc une assez grande dépendance au papier, Les différentes agences possèdent généralement des boîtes à suggestion.

Du côté du client, nous pouvons dire que le système n'est pas vraiment à l'avantage de ce dernier:

- Pour toute information ou renseignements, qu'il concerne la grille de prix ou les horaires, le client doit se rendre à l'agence qui l'intéresse.

- La réservation se fait sur place ou au téléphone par une interaction avec un agent de l'agence concernée.

-Chaque agence possède son propre système ; le client doit donc faire le tour s'il souhaite faire des comparaisons de services.

III- CRITIQUE DE L'EXISTANT

Cette critique de l'existant qui découle de l'analyse préalablement menée a pour but de fournir un état de situation actuelle et notamment de faire apparaître les qualités et les défauts de ce qui existe déjà ; il s'agit cependant d'être objectif sans chercher à tout construire. Pour la faire, nous ferons à ce point une critique constructive.

Suite à notre étude, on se rend compte que le système de transport interurbain présent dans notre société est fonctionnel, et on peut y observer des points forts mais aussi des points faibles.

III.2) Points forts

- ❖ La présence d'une traçabilité des données et des différentes transactions.
- ❖ La collaboration et la présence de la confiance entre les différents acteurs (clients, agence) dues à la proximité.

III.3) Point faible

- ❖ Absence presque systématique d'une base de données relationnelle.
- ❖ Risque de Pertes des données : En général, les données ne sont pas stockées sur le web.
- ❖ Le client doit forcément se déplacer ou appeler pour réserver et éventuellement arriver plus tôt pour s'assurer d'une place ou s'enquérir des prix de chaque agence.
- ❖ Des problèmes récurrents de monnaie.
- ❖ Le client ne peut pas comparer les prix en fonction des destinations sans se rendre dans les différents sièges des différentes entreprises.

Nous observons donc un système fonctionnel mais avec des inconvénients. Notre système viendra résoudre les points faibles évoqués et rendre le système de réservation de billets plus performant ainsi que celui de gestion de billets.

IV- PROBLEMATIQUE

La problématique du sujet "Conception et réalisation d'une application web et mobile de réservation de billets de bus de transport interurbain" réside dans la nécessité de répondre aux besoins croissants des utilisateurs en matière de mobilité et de faciliter l'accès aux services de transport interurbain. Dans un contexte où les déplacements entre les villes sont de plus en plus

fréquents, il est essentiel de proposer une solution pratique et conviviale pour la réservation de billets de bus.

En effet, les systèmes de réservation traditionnels peuvent présenter des limitations, tels que la nécessité de se rendre physiquement à une agence ou de réserver par téléphone, ce qui peut être contraignant et chronophage pour les utilisateurs. De plus, l'absence d'une plateforme centralisée et facilement accessible peut rendre difficile la comparaison des horaires, des tarifs et des disponibilités entre les différents transporteurs.

En conclusion, la problématique de ce projet est d'apporter une solution innovante qui réponde aux besoins des utilisateurs en matière de réservation de billets de bus interurbains, tout en facilitant l'accès à l'information, en améliorant l'efficacité des processus et en offrant une expérience utilisateur optimale.

V- PROPOSITION DE SOLUTION

Pour pallier aux problèmes cités plus haut, notre système sera mis en avant garantissant une amélioration des services déjà proposés. Elle pourra aussi assurer une plus grande autonomie au système, permettra au client de réserver de n'importe où, avec son téléphone mobile Android ou un ordinateur en lui donnant accès aux voyages prévus par ses agences favorites ; assurera à l'administration des outils pour une meilleure organisation de leurs différentes ressources et donc des différents voyages. Le tout, accessible en temps réel et de manière sécurisée. En résolvant cette problématique, le projet vise à améliorer l'expérience des utilisateurs en leur offrant une solution pratique et efficace pour la réservation de billets de bus, tout en favorisant la digitalisation du secteur des transports.

Conclusion

Ainsi, notre étude révèle un système de transport interurbain fonctionnel avec ses points forts, mais aussi ses points faibles qui sont négatifs, surtout pour les clients; ces résultats nous confortent dans l'idée que notre thème est plus que justifié.

DOSSIER 2 : CAHIER DES CHARGES

Introduction

Le cahier des charges (CDC) est un document contractuel à respecter lors d'un projet. Le cahier des charges permet au maître d'ouvrage de faire savoir au maître d'œuvre ce qu'il attend de lui lors de la réalisation du projet, entraînant des pénalités en cas de non-respect. Ainsi dans notre cahier de charge nous justifierons le projet et présenterons ses objectifs; ensuite, nous exprimerons les besoins utilisateurs au niveau des différents acteurs et nous ferons une estimation du coût du projet, puis nous terminerons avec les contraintes du projet et les livrables.

I- CONTEXTE ET JUSTIFICATION DU PROJET

Dans notre quotidien, que ce soit pour nous rendre à un évènement particulier, pour des vacances ou même pour du tourisme, nous sommes appelés à nous déplacer d'une ville à l'autre. Au Cameroun, pour se faire, l'on se sert principalement d'agences de transport interurbain. Toutefois, le système mis en place nous semble avoir un certain nombre de lacunes qui impliquent une perte de temps, et éventuellement d'argent. Afin de rendre plus autonome le système, nous avons donc jugé nécessaire et surtout optimal de traiter notre thème : mise en place d'une plateforme web et mobile de réservation de billets de transport urbain. Une plateforme qui regroupe les différents voyages créés par les agences de voyages ; accessible aux clients qui après une inscription peuvent rechercher, réserver et acheter un billet pour un voyage.

II- LES OBJECTIFS DU PROJET

II.1) Objectifs pour l'administration

Le projet pourra aider les administrations des différentes agences de voyage à optimiser la surveillance des différents voyages, de leurs différentes ressources (chauffeurs, bus etc).

II.2) Objectifs pour le client

Le projet a pour objectif principal de permettre au clients de visualiser les différents voyages disponibles, de comparer les différentes agences correspondantes, et réserver son billets de n'importe où.

III- EXPRESSIONS DES BESOINS DE L'UTILISATEUR

III.1) Besoin fonctionnel du projet

Les besoins fonctionnels sont les besoins qui aboutissent à une fonctionnalité dans le système développé, elle représente une action ou un ensemble d'actions à exécuter pour aboutir à un résultat donné. Les besoins fonctionnels du système sont groupés en catégories comme suit :

III.1.1) Au niveau du client

- ❖ S'inscrire
- ❖ Se connecter
- ❖ Se déconnecter
- ❖ Voir le listing des dernières réservations / derniers voyages.
- ❖ Écrire au support d'une agence ou de la plateforme.
- ❖ Effectuer une réservation en se basant sur les voyages disponibles.
- ❖ Annuler temporairement un ticket.

III.1.2) Au niveau de l'administrateur

- ❖ Se connecter
- ❖ Se déconnecter
- ❖ Gérer (ajout, modification; suppression) les ressources de l'agence (chauffeurs, bus, motoboys) et les voyages.
- ❖ Faire une réservation pour un client.
- ❖ Consulter les messages reçus.
- ❖ Consulter l'historique des voyages et les statistiques relatifs à son agence.

III.1.3) Au niveau du super-administrateur

- ❖ Gérer (ajout, modification, suppression) les administrateurs, les agences, les dispositions.
- ❖ Consulter les messages reçus.
- ❖ Voir toutes les transactions du système.
- ❖ Consulter les statistiques du système.

III.2) Besoin non fonctionnel du projet

Ce sont des besoins qui devront rendre le système plus accessible et plus convivial. Permettant ainsi l'utilisateur d'être satisfait du système.

III.2.1) Fiabilité

L'application doit fonctionner de façon cohérente sans erreurs en étant satisfaisante.

III.2.2) Ergonomie et utilisabilité

Les efforts fournis par l'utilisateur pour l'utilisation satisfaisante de l'application doivent être minimes.

III.2.3) Sécurité

Notre solution doit respecter surtout la confidentialité des données personnelles des utilisateurs qui reste l'une des contraintes les plus importantes dans un système.

III.2.4) Aptitude à la maintenance et la réutilisation

III.2.5) L'intégrité

Les données enregistrées ne doivent pas pouvoir être modifiées par une tierce personne sans autorisation ni être perdues.

III.2.6) Disponibilité : 24h/24 et 7j/7

IV- PLANIFICATION DU PROJET

Comme tout projet de conception, un planning des étapes doit être établis et connus. Pour notre projet les étapes sont les suivantes.

L'étude préalable : Le résultat de cette phase est la détermination des objectifs à atteindre dans notre future application en partant de l'existant.

Conception : Il s'agit de détailler les spécifications des fonctions ainsi que la structure des données, et des contrôles et les interfaces.

Réalisation : Il s'agit de l'implémentation des différentes fonctionnalités du programmes et effectuer les tests unitaires.

Test et Validation : Il s'agit de tester notre plateforme en lui faisant passer des audits de validation qui consisteront à détecter tout comportement inapproprié (faille) du système et le corriger.

Rédaction du rapport : Description détaillée de notre travail.

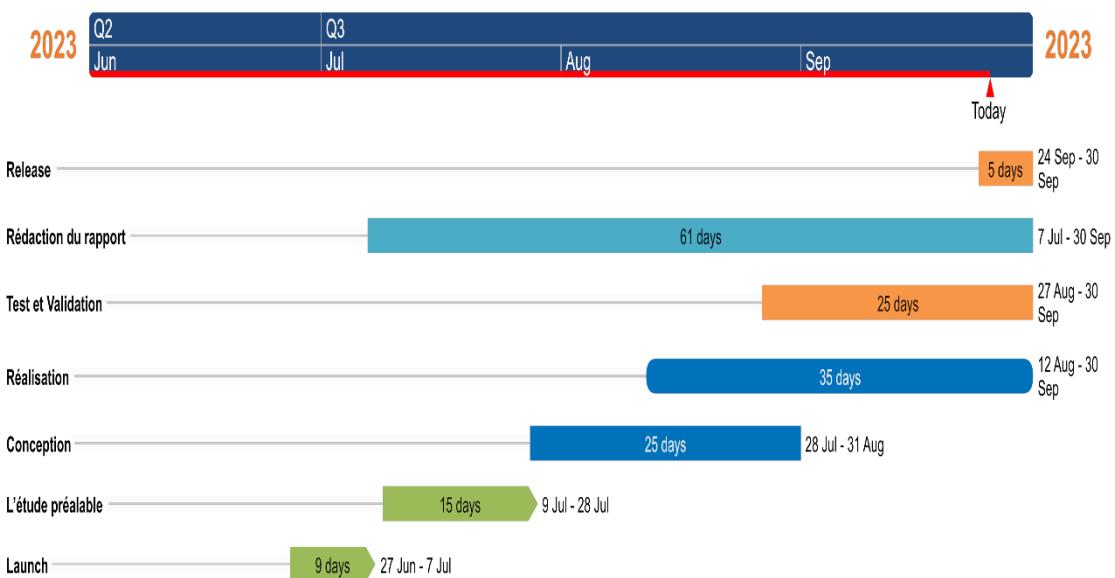


Tableau 05 : planifications de tâches

V- ESTIMATION DU COÛT DU PROJET ET EQUIPE DE TRAVAIL

Pour la réalisation de ce projet, nous devons avoir à notre disposition de ressources matérielles, logicielles, humaines et également financières, que nous avons listés comme suite :

V.1) Équipe de travail

Tableau 01 : équipe de travail

Noms et Prénom	Fonction
Fogue tiagho arsene nelson	Modélisation, Conception, et Réalisation

M. FENKAM Jules Blaise	Encadreur académique et chargé de communication à l'IAI Cameroun Centre technologique d'excellence de Paul Biya
M. JOHAN KENNETH NKEN	Chef de projet. Encadreur professionnel. Directeur de CLOUDS ARCHITECTS.

V.2) Resources matérielles

Tableau 02 : ressources matérielles

Matériels	Fonctions	Prix unitaire	Quantité	Prix total (FCFA)
Ordinateur : i7 2^e Gen 6Go Ram	Équipement de développement	150 000	03	450 000
Téléphone Android : 9, 2Go ram	Équipement de test	60 000	02	120 000
Modem Huawei	Accès à internet	150 000	01	150 000
Total	720 000 F CFA			

V.3) Ressources humaines

Tableau 03 : ressources humaines

Main d'œuvre	Nombres	Salaires
Analyste & concepteur	02	400 000
Designer	01	100 000
Programmeur	1	200 000
Tester	15	300 000
Total	15	1 000 000

V.4) Grand total

Tableau 04 : Total de prix des ressources du projet

Types	Prix (F CFA)

Matérielles	720 000
Humaines	1 000 000
Imprévues	1 720 000 * 15 %
Total de coût	1 978 000

VI- LES CONTRAINTES DU PROJET

Une contrainte est un élément qui limite un projet de par son potentiel par rapport à ses objectifs. Pour notre projet, nous devrons faire face à trois contraintes qui sont les suivantes ; Contrainte de temps, contrainte de coûts, contrainte de qualité et contrainte de ressources.

Tableau 06 : Contrainte du projet

Contraintes	Définitions
Contraintes de temps	il s'agit du calendrier de livraison du projet, notamment les dates de livraison de chaque phase du projet, ainsi que la date de livraison du livrable final. Ne pas respecter les deadlines des différentes phases peut entraîner un retard du projet global. Étant limité dans la réalisation de notre projet (03) mois; une planification optimale est nécessaire.
Contraintes de coûts	Les contraintes de coûts comprennent le budget du projet dans son ensemble et tout élément de valeur financière nécessaire au votre projet.
Contraintes de ressources	Les ressources sont étroitement liées aux contraintes de coûts de votre projet, car ces dernières représentent un certain coût. Une mauvaise allocation des ressources peut entraîner une baisse de qualité du projet, une augmentation du budget et des retards dans le calendrier.
Contrainte de qualité	Au départ du projet et tout au long de celui-ci, il est essentiel de garder à l'esprit que le produit final devra être de qualité, aussi bien sur le plan visuel (UI, UX) que sur le plan fonctionnel (résolution du problème des utilisateurs).

VII- LES LIVRABLES

A la fin du délai fixé pour le développement de cette solution, les éléments qui constituent les livrables sont :

- ❖ Le code source de l'application
- ❖ Le dossier d'analyse
- ❖ Le dossier de conception
- ❖ Le dossier de réalisation
- ❖ Le dossier de déploiement
- ❖ Un manuel d'utilisation

Conclusion

L'élaboration de notre cahier de charges nous a ainsi permis d'avoir une idée précise des exigences nécessaires à la réalisation de notre plateforme. Nous allons aborder avec beaucoup de lucidité la prochaine partie qui est le dossier d'analyse et de conception.

DOSSIER 3 : DOSSIER D'ANALYSE

Introduction

La démarche d'analyse est une étape fondamentale dans le processus de développement, puisqu'elle marque le début de la phase pratique en elle-même; liant les données du cahier de charge aux premiers diagrammes (Cas d'utilisation, séquence, activité), Cette partie sera dédiée aux premiers choix techniques, mais également, la modélisation des premiers diagrammes.

I- METHODOLOGIE

I.1) ETUDE COMPARATIVE UML ET MERISE

MERISE ET UML sont des méthodes/langages très utilisés dans les projets de développement logiciel, Décider de celui à utiliser est une étape cruciale de notre projet, il est donc important d'analyser les contours de notre projet et de connaître les différences entre les deux méthodes avant de faire le choix. Les différences entre UML et MERISE peuvent s'observer à plusieurs niveaux.

a. Introduction :

- UML : L'**unified modeling language** est un langage graphique standard utilisé pour modéliser des systèmes logiciels. Il fournit une notation visuelle pour représenter des concepts et des relations dans un système.
- Merise : Merise (**Méthode d'étude et de réalisation informatique pour les systèmes d'entreprise**) est une méthode de modélisation conceptuelle utilisée pour concevoir des systèmes d'information. Elle se concentre sur la modélisation des données, des processus métier et des interactions entre eux.

b. Domaines d'application :

- UML : UML est principalement utilisé dans le domaine du développement logiciel. Il permet de modéliser des systèmes logiciels complexes, tels que des applications, des systèmes embarqués ou des architectures orientées services.

- Merise : Merise est principalement utilisé dans le domaine de la gestion des systèmes d'information. Il est utilisé pour concevoir et modéliser des bases de données, des schémas de flux de données et des processus métier.

c. Niveaux de modélisation :

- UML : UML offre une variété de diagrammes qui permettent de modéliser différents aspects d'un système. Par exemple, le diagramme de cas d'utilisation représente les interactions entre les acteurs et le système, tandis que le diagramme de classes montre la structure des classes et leurs relations.
- Merise : Merise propose des modèles conceptuels, logiques et physiques pour la modélisation des données, des processus et des interactions. Le modèle conceptuel des données (MCD) représente les entités, les relations et les contraintes, tandis que le modèle logique des données (MLD) définit la structure détaillée de la base de données.

d. Approche de modélisation :

- UML : UML adopte une approche orientée objet, mettant l'accent sur la représentation des classes, des objets, des relations et des comportements. Il permet de modéliser des systèmes en identifiant des abstractions et en décrivant comment ces abstractions interagissent.
- Merise : Merise adopte une approche entité-association, mettant l'accent sur la représentation des entités, des attributs, des relations et des processus métier. Il se concentre sur la modélisation des données et des flux d'informations dans un système.

e. Notations et diagrammes :

- UML : UML utilise différentes notations graphiques pour représenter les concepts et les relations. Par exemple, les classes sont représentées par des rectangles avec des attributs et des méthodes, les relations sont représentées par des flèches entre les classes, et les interactions sont représentées par des séquences de messages.

- Merise : Merise utilise des notations graphiques spécifiques pour représenter les entités, les relations, les processus et les flux de données. Par exemple, les entités sont représentées par des rectangles, les relations sont représentées par des flèches, les processus sont représentés par des ovales, et les flux de données sont représentés par des flèches entre les processus.

f. Focus :

- UML : UML met l'accent sur la modélisation des aspects structurels et comportementaux des systèmes logiciels. Il permet de représenter les classes, les objets, les relations, les états, les événements et les interactions entre les composants d'un système.
- Merise : Merise met l'accent sur la modélisation des aspects structurels et fonctionnels des systèmes d'information. Il permet de représenter les données, les processus métier, les interactions entre les acteurs et les flux d'informations.

g. Utilisation complémentaire :

- UML : UML est souvent utilisé en conjonction avec des méthodologies de développement logiciel telles que les méthodes agiles ou en cascade. Il permet de documenter et de communiquer les différentes phases du cycle de vie d'un projet logiciel.
- Merise : Merise est souvent utilisé avec d'autres méthodes de gestion des systèmes d'information, telles que ITIL ou COBIT. Il permet de concevoir et de modéliser les systèmes d'information en se concentrant sur les données, les processus et les interactions.

h. Popularité et adoption :

- UML : UML est largement adopté dans l'industrie du développement logiciel. Il est soutenu par de nombreux outils de modélisation et est enseigné dans de nombreux programmes d'informatique.
- Merise : Merise est principalement populaire en France et dans les pays francophones, où il est largement utilisé dans le domaine de la gestion des

systèmes d'information. Il bénéficie d'une adoption significative dans les organisations qui suivent des normes de gestion spécifiques.

i. Flexibilité et extensibilité :

- UML : UML offre une grande flexibilité et la possibilité d'étendre le langage en définissant des profils spécifiques à un domaine ou à une entreprise. Cela permet d'adapter UML aux besoins particuliers d'un projet ou d'une organisation.
- Merise : Merise est moins flexible et moins extensible qu'UML, car il est plus spécifiquement axé sur la modélisation des systèmes d'information. Il offre cependant des concepts et des méthodes éprouvés pour la conception des systèmes d'information.

j. Choix de méthode :

- UML : UML est préféré pour la modélisation des systèmes logiciels complexes ou orientés objet. Il est largement utilisé dans le développement d'applications, de systèmes embarqués et de systèmes distribués.
- Merise : Merise est préféré pour la modélisation des systèmes d'information, en particulier dans les projets où la gestion des données et des processus est primordiale. Il est couramment utilisé pour concevoir des bases de données et des systèmes d'information dans divers secteurs d'activité.

Au final, la méthode la plus adaptée est UML, son approche orientée objet se concentrant sur les comportements est optimale pour nos outils de base adaptés à la POO (php, java). De plus, notre but, n'est pas de concevoir une application spécifique à un système d'information mais englobe un système plus grand qui s'élargit de jour en jour ; Merise n'est donc pas la solution.

I.2) ETUDE / PRESENTATION DE LA METHODE 2TUP

UP encore abrégé USDP ou PU en français est une famille de méthodes de développement de logiciels orientés objets. Elle se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML. Elle définit un processus intégrant toutes les activités de conception et de réalisation au sein de cycles de développement. Ses dérivées sont nombreuses, on peut distinguer **RUP, XUP, 2TUP**.

Devant le nombre considérable de méthodes disponibles, le choix devient difficile. Beaucoup de questions peuvent être posées à un chef de projet lors du lancement d'un projet : comment vais-je organiser l'équipe de développement ? Comment faire participer le client au développement afin de capter les besoins de celui-ci ? Pour y répondre, notre choix s'est porté sur la méthode 2TUP car c'est une méthode guidée par les besoins de l'utilisateur. 2TUP est un processus uniifié qui a pour but d'apporter une réponse aux contraintes de changement fonctionnelle et techniques qui s'imposent aux systèmes d'information. Il propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels. Il part du constat que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement suivant un axe fonctionnel et un axe technique. Il distingue ainsi deux branches (fonctionnelle et technique) dont les résultats sont fusionnés pour réaliser la solution finale. Il faut noter que le processus 2TUP commence d'abord par une étude préliminaire. Dans cette dernière, il s'agit d'identifier les acteurs qui vont interagir avec le système, les messages qu'échangent les acteurs et le système, puis à produire un cahier de charges et enfin à modéliser le contexte. A l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner le résultat des deux branches à savoir la branche fonctionnelle et la branche technique. Cette fusion conduit à l'obtention d'un processus de développement en forme Y, comme l'illustre le schéma suivant :

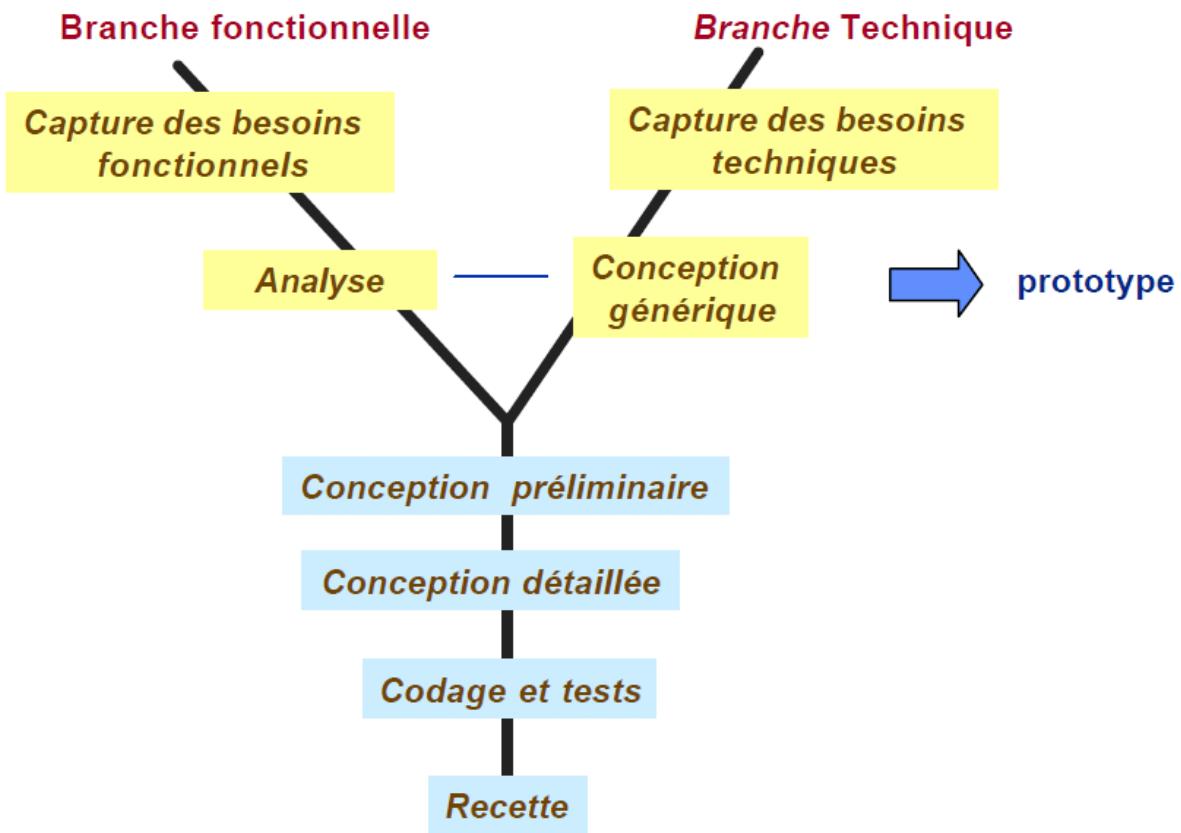


Figure 0 : schéma de la méthode 2TUP

II- MODELISATION

II.1) DIAGRAMME DE CAS D'UTILISATIONS

Le diagramme de cas d'utilisation (Use Case Diagram) constitue la première phase de l'analyse UML ; elle se concentre sur un certain nombre de tâches telles que :

- ❖ La modélisation les besoins des utilisateurs.
- ❖ L'identification les grandes fonctionnalités et les limites du système.
- ❖ La représentation des interactions entre le système et ses utilisateurs.
- ❖ La composition de cas d'utilisation

Tableau 07 : formalisme diagramme de cas d'utilisations

Composant	Description	Représentation graphique
-----------	-------------	--------------------------

Acteur	Un acteur est une entité extérieure au système modélisé, et qui interagit directement avec lui. Il existe 2 types d'acteurs : Primaire et secondaire.	
Cas d'utilisation	Un « use case » (cas d'utilisation) représente une fonctionnalité du système. Cette fonctionnalité est définie par une action déclenchant un ou plusieurs déroulements possibles.	
Association	Les acteurs et les cas d'utilisation sont reliés par des associations. De même que les acteurs du système entre eux.	

Notre système est constitué de trois groupes d'utilisateurs qui interagissent avec ce dernier en réalisant un ou plusieurs cas d'utilisations. Comme acteur, nous avons :

- ❖ **Le client** : C'est un utilisateur qui achète son ticket.
- ❖ **L'administrateur**: Il utilise le système pour administrer les ressources de son agence et effectue une réservation pour un client.
- ❖ **Le super-administrateur** : Celui-ci a le contrôle du système entier, c'est-à-dire qu'il a accès à l'intégralité du système sauf sur restriction.

Les interactions entre les acteurs et le système seront représentés grâce au diagramme de cas d'utilisation du langage de modélisation UML 2.0. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs-système. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera.

Tableau 08 : liste des cas d'utilisations

CAS D'UTILISATION	ACTEURS IMPLIQUES
Acheter un ticket	Client
Consulter la de ses tickets achetés	Client
Gérer les bus, les voyages, les chauffeurs, les réservations, les motoboys.	Administrateur

Consulter l'historique des voyages de son agence.	Administrateur
Gérer les admins, les agences, les dispositions.	Super-Administrateur
Faire un listing des voyages disponibles	Client, administrateur

II.1.1) Diagramme de cas d'utilisation du client

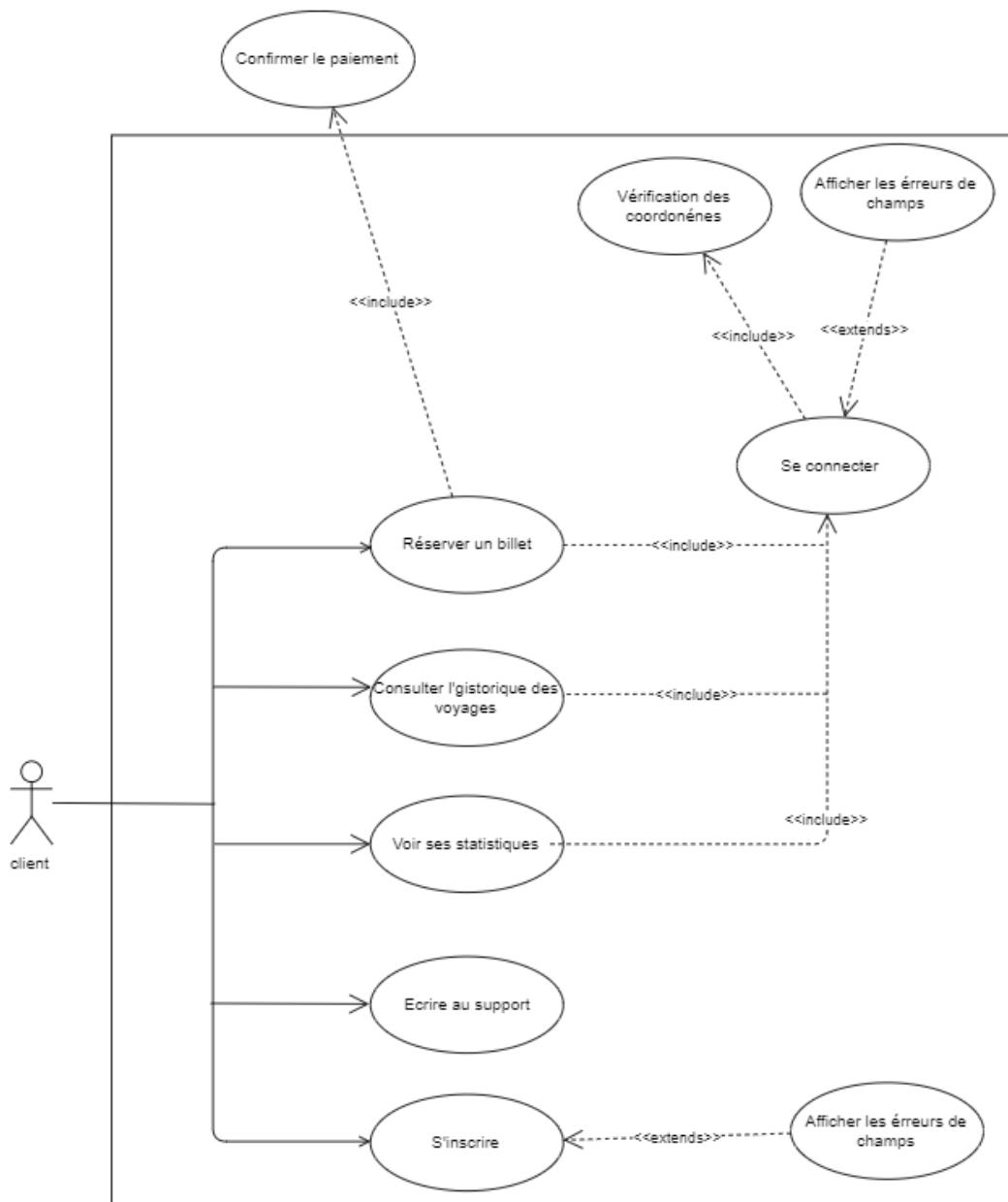


Figure 01 : cas d'utilisation du client

II.1.2) Diagramme de cas d'utilisation de l'administrateur

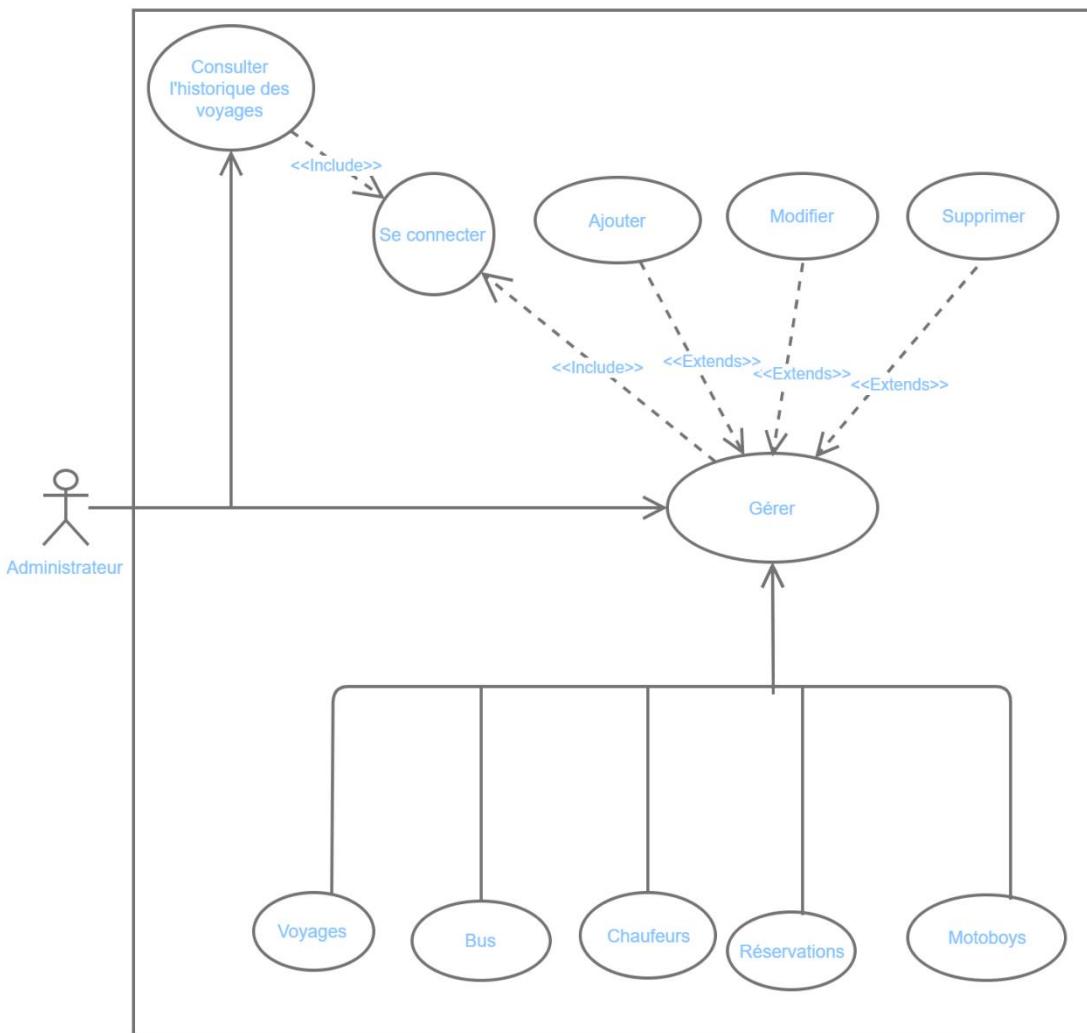


Figure 02 : cas d'utilisation de l'administrateur

III.1.3) Diagramme de cas d'utilisation du super-administrateur

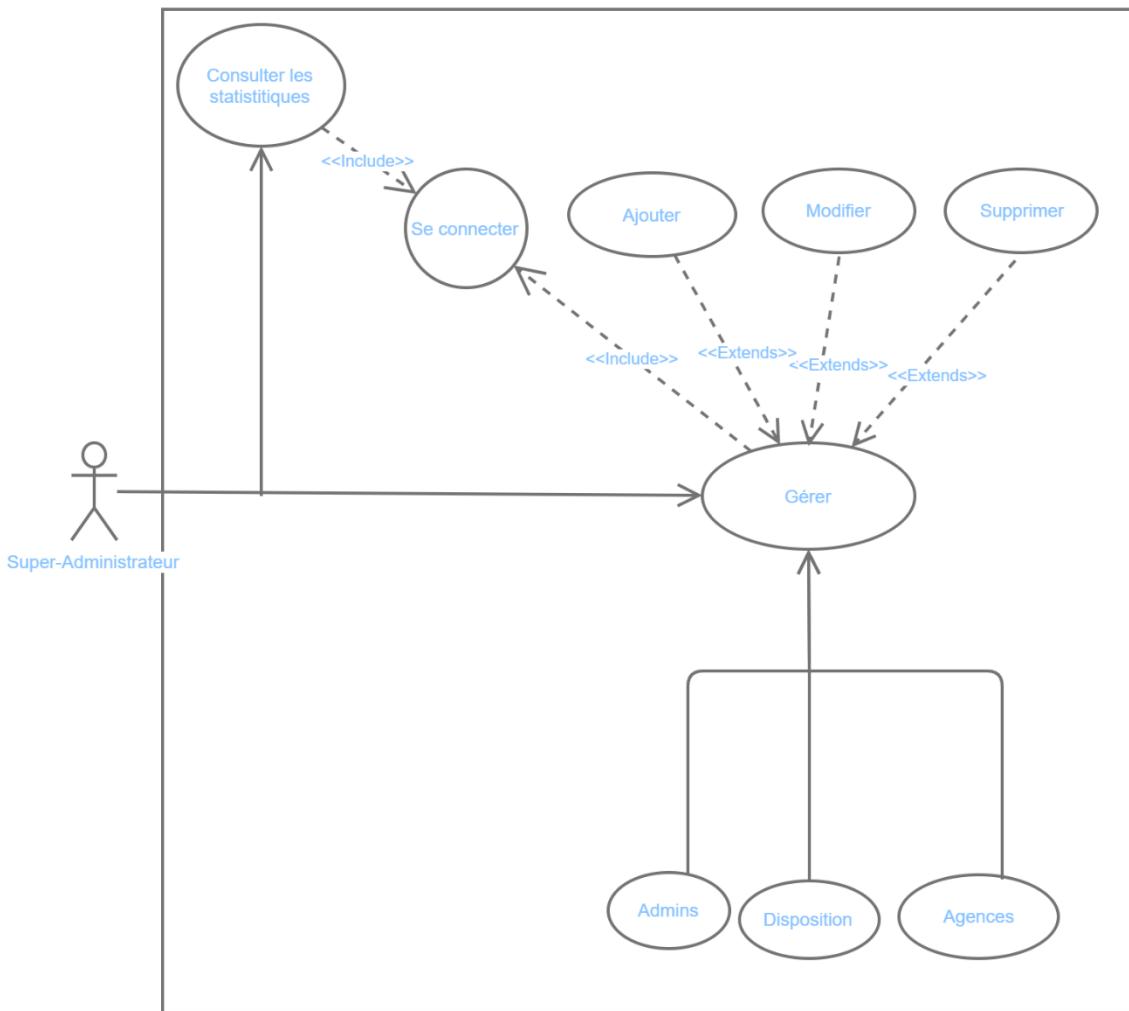


Figure 03 : cas d'utilisation du super-administrateur

II.1.3) Description des cas d'utilisations

Dans le but de mieux comprendre notre système et les interactions avec les utilisateurs, nous allons détailler dans cette partie les scénarios des principaux cas d'utilisation.

Tableau 09 : cas d'utilisation d'authentification

CU1 : Authentification
Résumé : permet aux différents acteurs d'accéder à leur espace de travail
Acteurs : administrateur, client, super-administrateur
Scénario nominal
DESCRIPTION DU SCENARIO NOMINAL

« Début »

01 : saisir les informations d'authentification

02 : cliquer sur login

« Fin » « Fin »

DESCRIPTION DU SCENARIO ALTERNATIF

Les informations entrées ne sont pas valides : Coordonnées invalides.

Tableau 10 : cas d'utilisation de création d'un administrateur

CU2 : Créer un administrateur
Résumé : L'on crée un administrateur pour une agence précédemment enregistré, afin qu'il puisse Gérer les ressources de son agence.
Acteur : Super-administrateur
Scénario nominal
DESCRIPTION DU SCENARIO NOMINAL
« Début »
01 : authentification
02 : saisir les informations de l'administrateur
03 : confirmer la création
« Fin »
Scénario alternatif
DESCRIPTION DU SCENARIO ALTERNATIF
Les informations entrées ne sont pas valides : Une erreur est renvoyée au super-administrateur en fonction du champ mal renseigné.

Tableau 11 : cas d'utilisation de réservation d'un ticket

CU3 : Réserver un ticket
Résumé : Permettre à l'utilisateur de faire un choix sur le voyage qu'il souhaite effectuer et de réserver son ticket.
Acteur : client
Scénario nominal

DESCRIPTION DU SCENARIO NOMINAL

« Début »

01 : se connecter

02 : Cliquer sur Faire une réservation

03 : Entrer les informations sur le voyage que l'on recherche (date, heure, ville de départ, ville d'arrivée).

04 : Cliquer sur Réserver, sur la carte du voyage qui nous intéresse.

05 : Entrer les informations du voyageur et de paiement, puis sur Réserver.

06 : Valider la requête de retrait reçu sur notre téléphone.

« Fin »

Scénario alternatif

DESCRIPTION DU SCENARIO ALTERNATIF

-Les informations entrées ne sont pas valides : Renvoi d'un message d'erreur correspondant au champ mal renseigné.

-L'utilisateur ne valide pas la demande de retrait : Message d'erreur « la requête a été annulée, veuillez réessayer s'il vous plaît ! »

Tableau 12 : cas d'utilisation ajouter un voyage

CU4 : Ajouter un voyage
Résumé : permettre de créer un nouveau voyage
Acteur : administrateur
Scénario nominal
<h4>DESCRIPTION DU SCENARIO NOMINAL</h4> <p>« Début »</p> <p>01 : se connecter</p> <p>02 : cliquer sur voyage</p> <p>03 : Entrer les informations du voyage telles que le code, le bus, le chauffeur, le motoboy, l'heure de départ, les villes de départ et d'arrivée.</p> <p>04 : Cliquer sur Ajouter.</p> <p>« Fin »</p>
Scénario alternative

DESCRIPTION DU SCENARIO ALTERNATIF

Les informations entrées ne correspondent pas au format demandé: Renvoi d'un message d'erreur correspondant au champ mal renseigné.

Tableau 13 : cas d'utilisation de reprogrammation d'un voyage

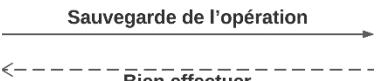
CU5 : Reprogrammer un voyage
Résumé : Permettre à un client d'annuler un voyage et de le reprogrammer plus tard
Acteur : Client
Scénario nominal
DESCRIPTION DU SCENARIO NOMINAL
« Début »
01 : authentification
02 : Se rendre dans la liste des tickets achetés et cliquer sur le bouton « annuler » du ticket
03 : Se rendre dans la section « reprogrammer un ticket » et cliquer sur le bouton reprogrammer du ticket concerné. Choisir le voyage à reprogrammer et valiser
« Fin »
Scénario alternatif
DESCRIPTION DU SCENARIO ALTERNATIF
SCA1 : Le voyage choisis n'a pas le même tarif que le ticket annulé : l'utilisateur reçoit un message d'erreur.
SCA2 : L'utilisateur essaie d'annuler moins d'1h avant le paiement : le client reçoit une erreur.

II.2) DIAGRAMME DE SEQUENCES

Le diagramme des séquences permet de présenter une documentation des interactions à mettre en œuvre entre les classes pour réaliser un résultat. UML étant conçus pour la programmation orientée objet, les communications entre les classes sont connues comme des messages. Ce diagramme énumère les objets horizontalement et le temps verticalement. Chaque message est modélisé en fonction du temps.

Tableau 15 : formalisme du diagramme de séquences

Composant	Description	Représentation graphique
-----------	-------------	--------------------------

Objet	Les objets sont les instances de classe et sont rangés horizontalement. Acteur Personne qui interagit ou communique avec le système	
Ligne de vie	La ligne de vie identifie l'existence de l'objet par rapport au temps	
Les messages	Qui vont de l'acteur vers l'objet sont dits « synchrones » et ceux qui vont de l'objet vers l'acteur sont dits « asynchrones »	
Activation	Représente le temps nécessaire pour qu'un objet accomplisse une tâche. Plus la tâche nécessite de temps, plus la boîte d'activation est longue.	

III.2.1) Diagramme de séquence : Authentification

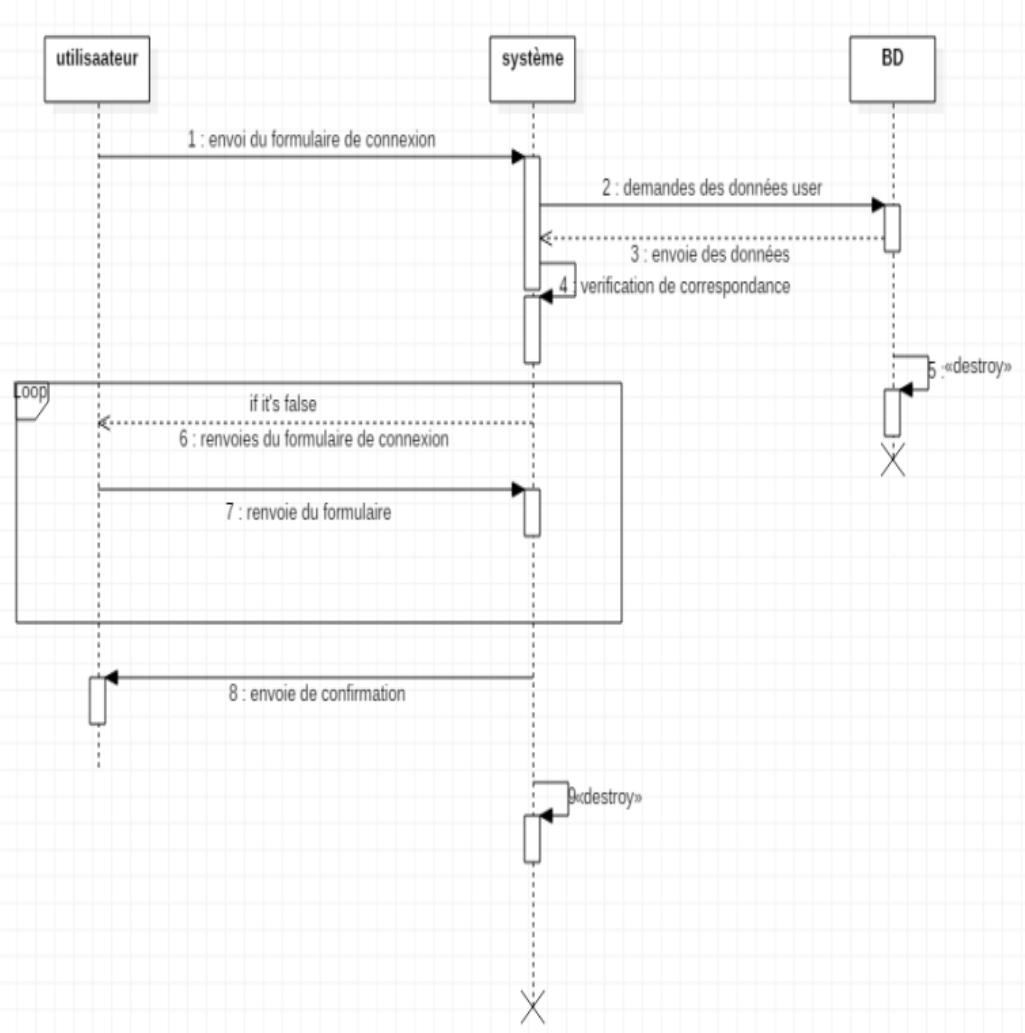


Figure 05 : Diagramme de séquence authentification

III.2.2) Diagramme de séquence : Réservation

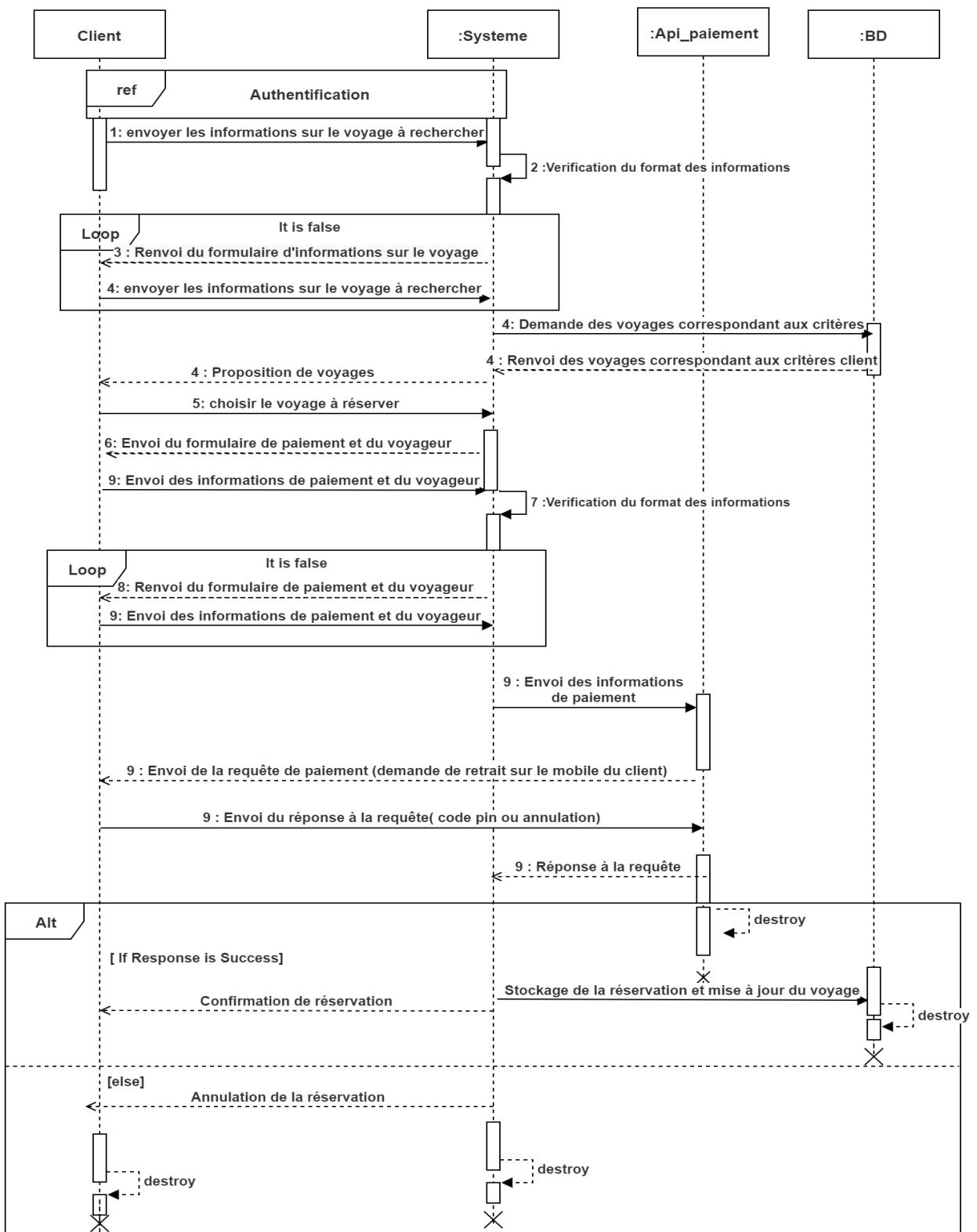
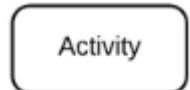
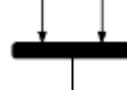
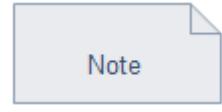


Figure 06 : Diagramme de séquence « réservation »

II.3) DIAGRAMME D'ACTIVITÉS

Dans le langage UML, le diagramme d'activité fournit une vue du comportement d'un système en décrivant la séquence d'actions d'un processus. Les diagrammes d'activité montrent les flux d'actions entre activités, mais, ils peuvent également montrer des flux parallèles, simultanés et les flux de remplacement.

Tableau 16 : formalisme du diagramme d'activités

Composant	Description	Symbole
Symbole de début	Représente le début d'un processus ou d'un flux de travail dans un diagramme d'activités. Il peut être utilisé seul ou avec un symbole de note qui explique le point de départ.	
Symbole d'activité	Indique les activités qui composent un processus modélisé. Ces symboles, qui comprennent de brèves descriptions dans la forme, sont les principales composantes d'un diagramme d'activités.	
Symbole de raccord	Indique le flux directionnel, ou flux de contrôle, de l'activité. Une flèche entrante marque le début d'une étape d'une activité ; une fois l'étape terminée, le flux se poursuit avec la flèche sortante.	
Symbole de raccord/barre de synchronisation	Associe deux activités simultanées et les réintroduit dans un flux où n'a lieu qu'une seule activité à la fois. Représenté par une ligne verticale ou horizontale épaisse.	
Symbole d'embranchement	Divise un flux d'activités en deux activités simultanées. Symbolisé par plusieurs lignes fléchées qui partent d'un raccord.	
Remarque	Permet aux créateurs d'un diagramme ou à leurs collaborateurs de communiquer des messages supplémentaires qui n'entrent pas dans le diagramme à proprement parler.	

	Permet de laisser des notes pour plus de clarté et de précision.	
L'activité de décision	L' activité de décision est introduite dans UML pour supporter les conditionnels dans les activités. Une activité de décision est modélisée comme un diamant sur un diagramme d'activité UML. L'activité de décision doit refléter l'activité précédente.	
Symbole de fin	Marque l'état final d'une activité et représente l'achèvement de tous les flux d'un procédé.	

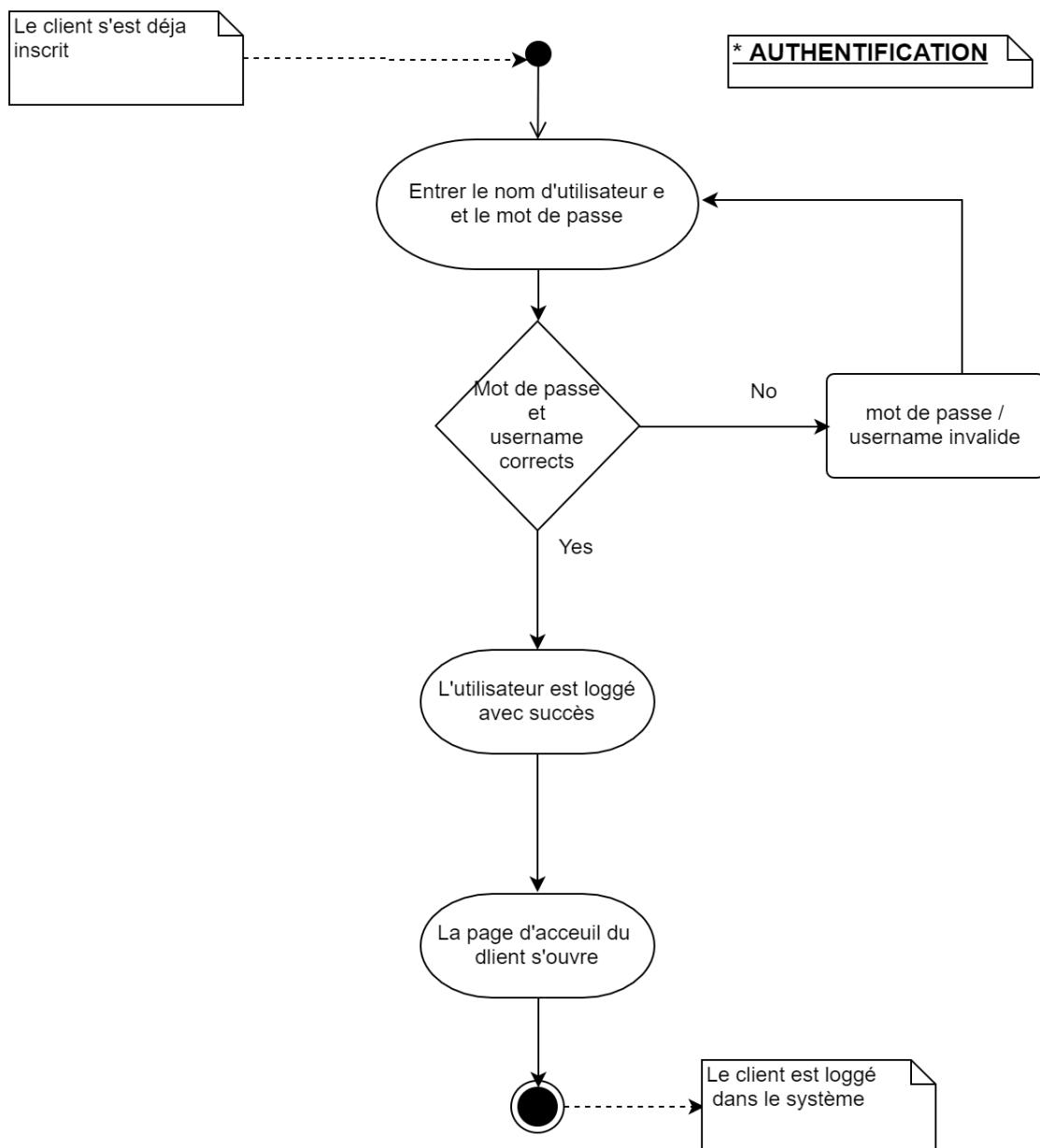


Figure 07 : Diagramme d'activité « authentification »

Conclusion

L'analyse faite ci-dessus nous a permis de faire un ensemble de choix quant aux méthodes à utiliser dans la mise en place de notre application notamment UML et 2TUP. Elle nous a également permis d'avoir une vision plus claire des différentes exigences fonctionnelles de notre système, mais aussi techniques. Il nous faut donc passer à la prochaine étape, la conception.

DOSSIER IV : DOSSIER DE CONCEPTION

Introduction

Au même titre que la démarche d'analyse, la phase de conception constitue une phase non négligeable dans la mise sur pieds d'un système informatique. Se focalisant principalement, sur les diagrammes tels que le diagramme de classe et d'états transition, elle fait correspondre la vision applicative (analyse) à la vision technique (environnement de développement et d'exécution) en donnant. Cela implique de traduire les spécifications fonctionnelles en une architecture logicielle, en définissant les modules, les relations entre les différents composants, les flux de données, les interfaces utilisateur, etc.

I- Dictionnaire de données

Le dictionnaire de donnée est un tableau qui sert de référence à l'ensemble de l'équipe du projet. Ce dernier décrit les types et les objets du système. Il s'agit donc d'une représentation des données du champ de l'étude ainsi que le lien sémantique entre ces données, Il doit être facilement compréhensible et accepté par tous.

Le tableau ci-dessous représente la liste des attributs composants toutes les classes formants notre système ainsi que leur description, leur taille et leur type.

Tableau 14 : liste des attributs

Numéro	Attribut	Libelle	Type	Taille
01	Id_user	Identifiant utilisateur	Integer	//
02	Id_agency	Identifiant agence	Integer	//
03	Id_ticket	Identifiant d'un ticket	Integer	//
04	Id_driver	Identifiant d'un chauffeur	Integer	
05	fullname	Nom complet de l'utilisateur	Chaine	30
06	Role	Role d'un utilisateur	Chaine	10
07	Agency_name	Nom de l'agence	Chaine	15
08	Price	Prix d'un ticket	Réel	10

09	travelType	Type du voyage	Chaine	30
10	travelState	Etat d'un voyage	Booleen	//
11	Logo	Lien vers le logo de l'agence	Chaine	70
12	Nb_place	Nombre de place d'un bus	Réel	10
13	Registration	Numéro d'immatriculation d'un bus	Chaine	30
14	Secteur	Secteur de départ d'un voyage	Chaine	30
15	Departure	Ville de départ d'un voyage	Chaine	30
16	Arrival	Ville d'arrivée d'un voyage	Chane	30
16	numeroCNI	Numéro de la carte d'identité nationale	Chaine	30
17	Message_content	Contenu d'un message	Chaine	255
18	Paiement_number	Numéro mobile money pour les paiements	Entier	13
19	Created_at	Date de création d'un element	Date	
20	Read	Statut d'un message	Chaine	20

II- Diagramme des classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation

Une classe est composée :

- ✓ Attributs : représentant des données dont les valeurs représentent l'état de l'objet.
- ✓ Méthode : il s'agit des opérations applicables aux objets.

Les associations sont des relations entre classes. Elles représentent une collaboration. Elles sont représentées par une ligne entre les classes.

Association simple : les associations simples sont des liaisons logiques entre entités.

Les cardinalités : précisent combien d'objets d'une classe considérée peuvent être liés à un objet de l'autre classe.

Tableau 17 : les cardinalités

Cardinalités	Désignation
1 / 1..1	Un et un seul
0..1	Zéro ou un
N	Entier naturel
m..n	De m à n (deux entiers naturels)
0..*	De 0 à plusieurs
1..*	De 1 à plusieurs

La figure ci-dessous est celle d'un diagramme de classes qui contient toutes les informations telles que les classes, les méthodes, les associations et les propriétés.

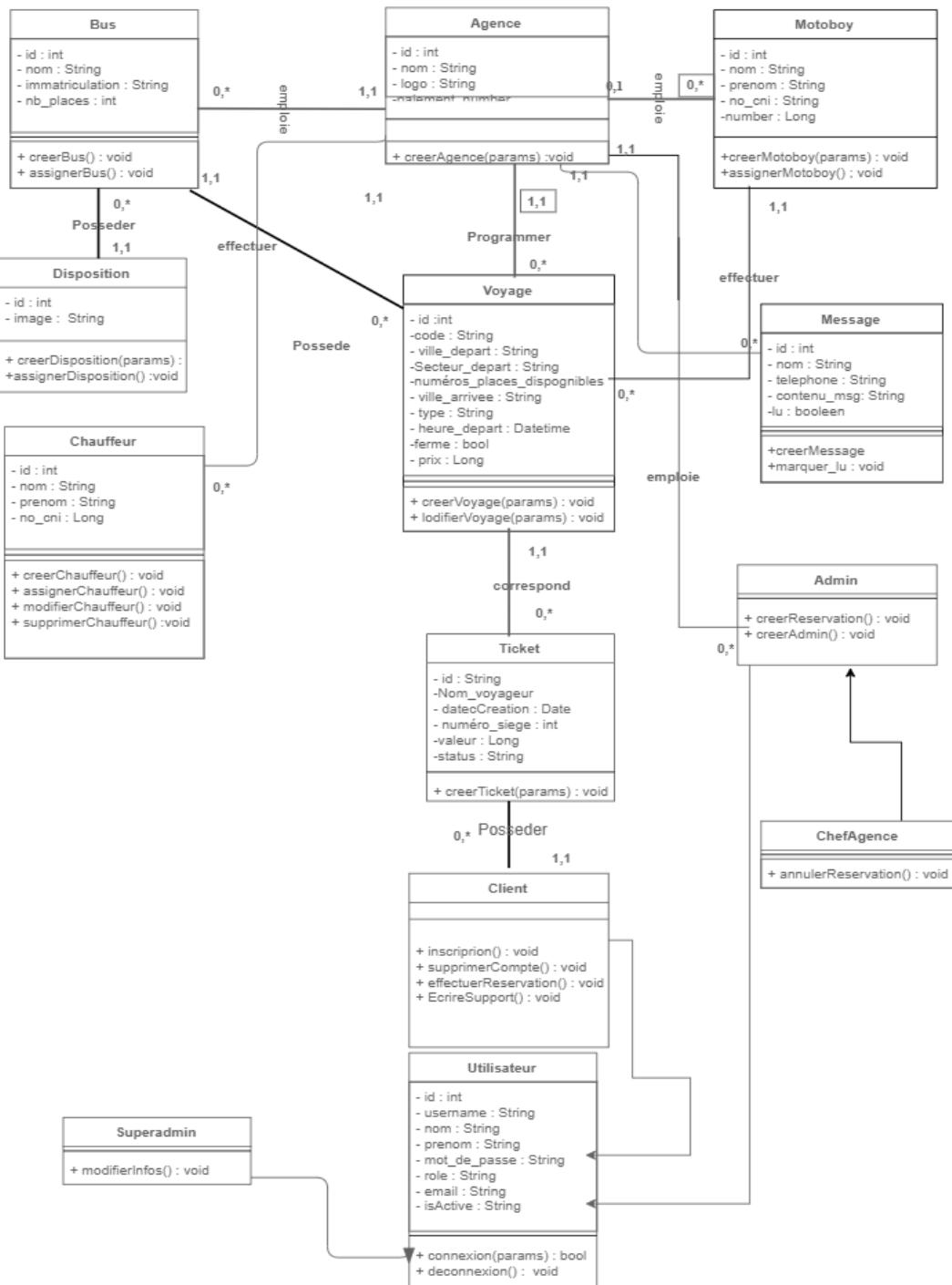


Figure 08 : diagramme de Classe

III- Diagramme d'état transition

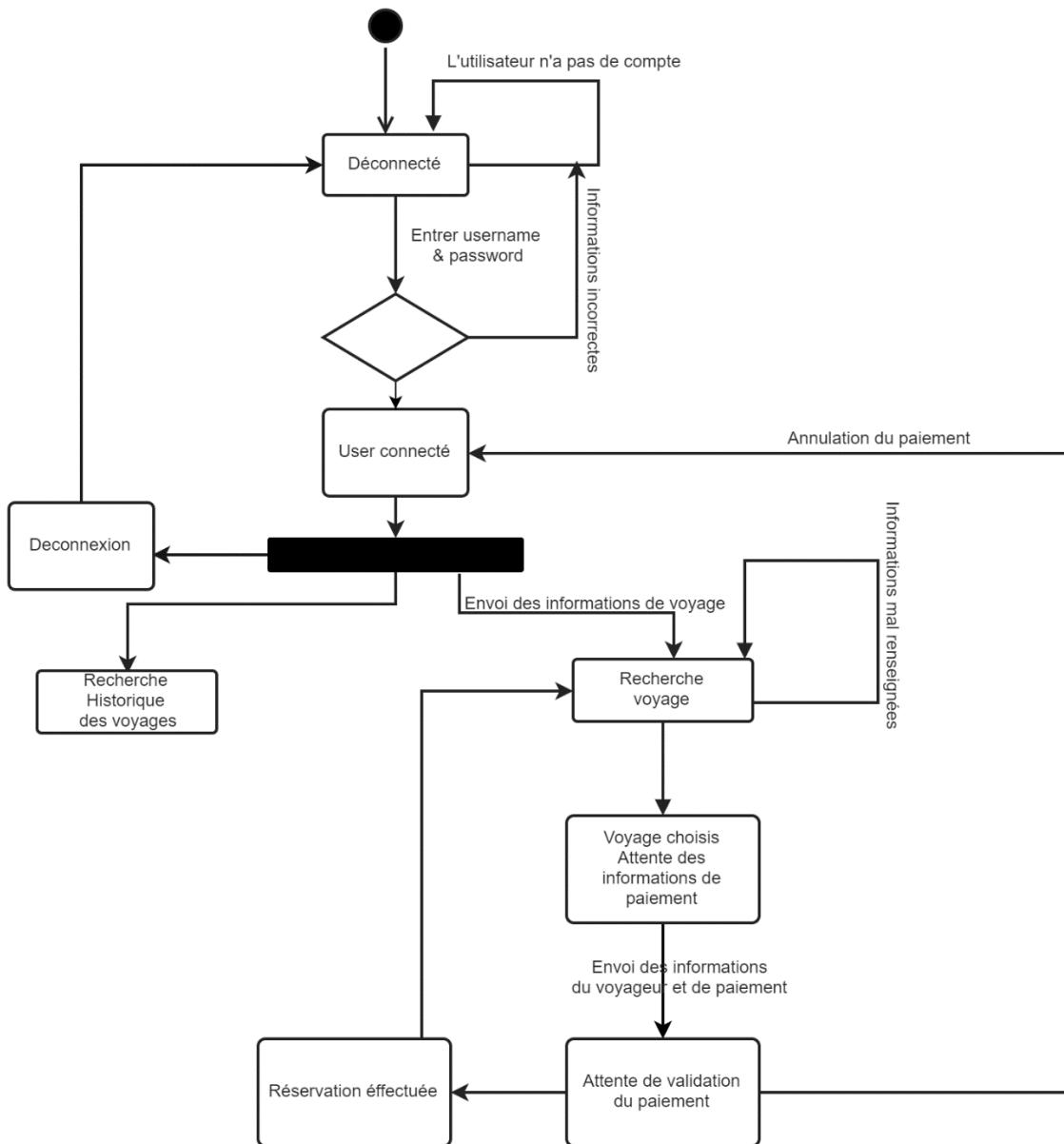
Le diagramme d'état-transition d'UML est un type de diagramme qui permet de modéliser le système sous formes d'états et de transitions entre ces états. Il est utilisé pour représenter le cycle d'un objet ou d'un système, en mettant l'accent sur les différentes situations dans lesquelles il peut se retrouver et les évènements qui provoquent les transitions entre ces situations.

Tableau 18 : formalisme du diagramme d'état-transition

Composant	Description	Symbole
Premier état	Marqueur du premier état du processus, représenté par un cercle noir avec une flèche de transition.	
Évènement	Instance qui déclenche une transition. Son nom figure au-dessus de la flèche de transition applicable	-
Etat	Rectangle aux coins arrondis qui indique la nature actuelle d'un objet.	
Terminator	Cercle avec un point à l'intérieur, qui signifie qu'un processus est terminé.	
Transition	Flèche allant d'un état à un autre et indiquant un changement d'état	
Comportement de transition	Comportement résultant de la transition d'un état, inscrit au-dessus de la flèche de transition.	-

Pseudo-état choix	Losange qui indique un état dynamique avec des résultats potentiels variables.	
Déclencheur	Type de message qui déplace activement un objet d'un état à un autre, inscrit au-dessus de la flèche de transition.	

Figure 09 : diagramme d'état-transition utilisateur et réservation



DOSSIER V : DOSSIER DE REALISATION ET DE DEPLOIEMENT

Introduction

La réalisation de notre projet a nécessité un certain nombre d'outils et de technologies que nous exposeront dans le présent dossier. Lors de cette réalisation, nous nous sommes assurés d'utiliser des technologies et outils modernes, afin de garantir la mise en œuvre d'un produit robuste et fiable, tout en gardant à l'esprit de concevoir une plateforme conviviale et agréable à l'utilisation.

Une fois la modélisation et la conception du produit terminées, l'étape suivante dans la mise sur pieds d'une solution est la réalisation suivie par le déploiement, la concrétisation de toute la phase de conception.

I- L'ÉTUDE TECHNIQUE : LA REALISATION

L'étude technique est une phase d'adaptation de conception à l'architecture technique. Elle a pour objectif de décrire au plan fonctionnel la solution à réaliser d'une manière détaillée ainsi que la description des traitements. Cette étude, qui suit l'étude détaillée, constitue le complément de spécification informatique nécessaire pour assurer la réalisation du futur système. Cette étude permet également de déterminer :

- ❖ La structure informatique de la base de données
- ❖ L'architecture des programmes
- ❖ La structure de chaque programme et l'accès aux données.

I.1) Environnement de réalisation

Pour la réalisation de notre application, nous avons eu recours à plusieurs moyens matériels et logiciels :

I.1.1) Matériels de base :

Tableau 17 : Matériels de base

Matériel	Caractéristique	Utilité
----------	-----------------	---------

Ordinateur portable	I7 2e Gen 6Go Ram	Machine de développement et de tests de la version web
2 téléphone mobile Android	- 6.6 " pouces - 4.7 " pouces	Les téléphone de taille d'écran différente sont là pour valider la compatibilité des vue su système sur diffèrent écran.

I.1.2) Choix des langages de développement et de SGBD :

1. Application mobile

- ❖ Frontend
 - XML :



Figure 09 : logo XML

L'Extensible Markup Language, généralement appelé XML, « langage de balisage extensible » en français, est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup Language (SGML). Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec pour chacun son vocabulaire et sa grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte enrichi, etc.) entre systèmes d'informations hétérogènes (interopérabilité).

https://fr.wikipedia.org/wiki/Extensible_Markup_Language

- ❖ Backend
 - JAVA

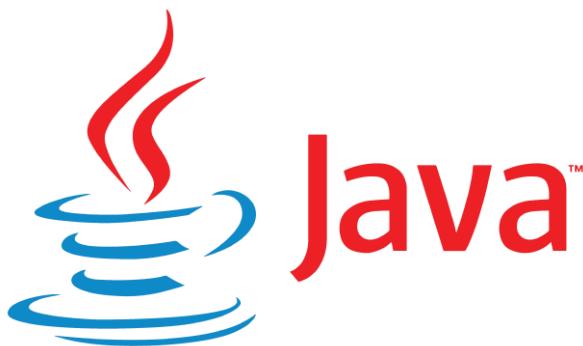


Figure 10 : logo Java

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

[https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage))

2. Application web

- ❖ Frontend
 - Tailwindcss

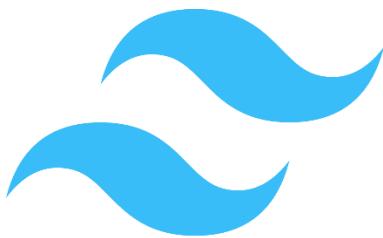


Figure 12 : logo Tailwindcss

Tailwindcss ou simplement tailwind est un framework de conception moderne et populaire qui simplifie le développement en fournissant une approche utilitaire(utility-first) pour la création d'interfaces utilisateur. Avec tailwindcss, l'on peut construire applications rapidement en utilisant des classes de bas niveaux prédéfinies. La principale force de tailwind réside dans son système de configuration qui permet au développeur de personnaliser chaque aspect du framework en utilisant un fichier de configuration dédié. Le concept utility-first fait référence à la fonctionnalité différentielle principale de Tailwind. Au lieu de créer des classes autour des composantes (bouton, panneau, menu, champ de texte...), les classes sont conçues autour d'un élément de style spécifique (couleur jaune, police de caractère grasse, texte large,

élément du centre...). Toutes ces classes sont nommées classes utilitaires (utility classes). Il y a beaucoup de classes utilitaires dans Tailwind CSS qui permettent de contrôler un grand nombre de propriétés CSS telles que les couleurs, la bordure, le type d'affichage (display), la taille du texte et la police calligraphique, la mise en page, l'ombrage...

- **JavaScript**

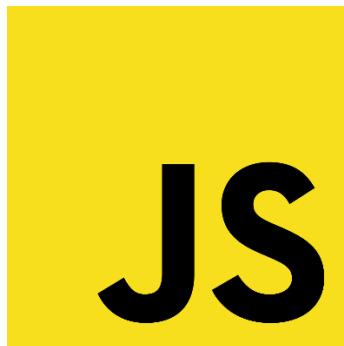


Figure 13 : logo JavaScript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web3. Une grande majorité des sites web l'utilisent4, et la majorité des navigateurs web disposent d'un moteur JavaScript5 pour l'interpréter. JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web3. Une grande majorité des sites web l'utilisent4, et la majorité des navigateurs web disposent d'un moteur JavaScript5 pour l'interpréter.

- **HTML**



Figure 14 : logo HTML

HTML est le langage de balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Il peut être assisté par des technologies telles que la feuille de style en cascade (CSS) et des langages de script tels que JavaScript.

- CSS



Figure 15 : logo css

CSS (Cascading Style Sheets) est un langage utilisé pour décrire la présentation et le style des documents HTML. Il permet de contrôler l'apparence des éléments d'une page web, tels que la couleur, la police, la disposition, etc. Introduit au milieu des années 1990, CSS est un élément essentiel du développement web moderne. Si le HTML permet de définir le squelette d'une page web, le CSS lui, se charge de paramétriser son apparence ; grâce à des feuilles de styles en cascade, des sélecteurs de diverses natures, des propriétés suivies de leurs valeurs. Il offre ainsi une multitude de possibilités pour personnaliser l'apparence des sites et créer des designs attrayants. Son utilisation dans le développement web est essentielle pour fournir une expérience utilisateur esthétique et cohérente.

❖ Backend

-Laravel

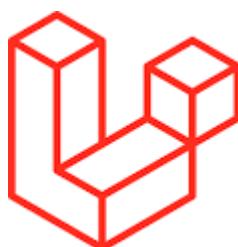


Figure 15 : logo de laravel

Laravel est un framework PHP qui propose des outils pour construire une application web. Il regroupe les meilleures librairies pour chaque fonctionnalité nécessaire à la création d'un site web. Il a été, en ce sens, construit en se basant sur Symfony. Laravel a été développé

et créé par Taylor Otwell comme une tentative de donner un excellent substitut à l'ancien Framework PHP nommé **CodeIgniter**, qui était le Framework le plus utilisé en php à cet époque. Créé en **2011**, Laravel automatise un large éventail de processus de développement logiciel et facilite les tâches de développement Web courantes telles que l'authentification, le routage, les sessions et la mise en cache. Laravel est devenu l'outil privilégié des développeurs. Grâce à sa grande facilité d'utilisation et à l'élégance de sa syntaxe, mais également à l'écosystème très complet qui l'environne (Homestead, Laracasts, Forge). Laravel est devenu depuis 2016 le Framework le plus utilisé dans le monde.

3. Outils de réalisation

- GitHub



Figure 16 : logo GitHub

GitHub (/gɪθəb/, entreprise GitHub, Inc.) est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang par Chris Wanstrath, PJ Hyett et Tom Preston-Werner. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres.

Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet. Le site est devenu le plus important dépôt de code au monde, utilisé comme dépôt public de projets libres ou dépôt privé d'entreprises.

-Visual Studio Code

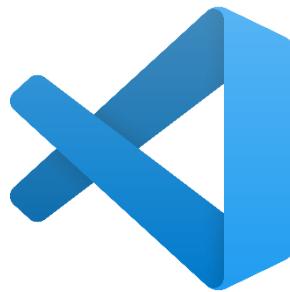


Figure 17: logo de visual studio code

Visual studio code, communément appelé **VS Code** est un éditeur de texte publié et maintenu par **Microsoft** depuis 2015. Il embarque un large éventail de fonctionnalités parmi lesquelles on retrouve la coloration syntaxique, la complétion intelligente de code, le refactoring, la gestion de code avec Git, un terminal riche etc. Ces fonctionnalités couplées à un système d'extensions très puissant, nous ont permis d'éditer notre code source, de le tester de manière intelligente et surtout rapide.

https://en.wikipedia.org/wiki/Visual_Studio_Code

- **Android studio**



Figure 18: logo de Android Studio

Android studio est un **IDE** pour le développement d'applications mobiles Android; il est basé sur IntelliJ IDEA et utilise le modèle de production GRADLE. Il est annoncé en 2013 par Google et après sa sortie en décembre 2013, il est peu à peu démocratisé par la communauté de développeurs mobiles. Android Studio permet principalement d'écrire du code source Java/Kotlin et les fichiers de configuration XML d'une application Android. Il possède également un système d'extensions très poussé permettant d'avoir accès à toujours plus d'outils de développement ainsi qu'un système de virtualisation permettant de prévisualiser l'application en cours de développement.

https://fr.wikipedia.org/wiki/Android_Studio

- **Canvas**



Figure 18 : logo canvas

Canva est une plateforme en ligne très populaire qui a vu le jour en 2009 sous le nom de Fusion book. Elle a subis un certain nombre de modifications en 15 ans et est aujourd’hui très populaire. La plateforme permet de créer facilement des designs professionnels. Que ce soit pour des présentations, des posters, des cartes de visites, Canva propose de nombreux outils et modèles pour réaliser des créations graphiques attrayantes. Avec son interface conviviale, canva rends la création de designs accessible à tous, même aux personnes sans expérience en design. Sélectionner un modèle prédéfinis et y renseigner ses informations semblent suffisant pour faire ses premiers pas sous canva.

- https://fr.wikipedia.org/wiki/Canva_Pty_Ltd

- **Microsoft office Word 2013**



Figure 19 : logo Microsoft Word

Microsoft Word, qui est un logiciel de traitement de texte couvre deux notions, assez différentes en pratique : Un éditeur de textes interactif et un compilateur pour un langage de mise en forme de textes.

- **Microsoft office Power point 2013**



Figure 20 : logo Microsoft PowerPoint

Microsoft Office PowerPoint est un créateur de présentations (succession de diapositives). Il est utilisé pour créer des présentations avec du texte, avec des images, sons, vidéos et autres objets

- **Gitmind**



Figure 2 : logo GitMind

GitMind est un créateur de cartes conceptuelles en ligne gratuit. C'est un outil parfait pour vous permettre de visualiser des concepts complexes, de générer de nouvelles idées, de dresser des listes et des plans et de préparer des présentations. Comparé à d'autres essais gratuits de logiciels de cartographie mentale, GitMind fournit un service totalement gratuit permettant de créer un nombre illimité de cartes mentales, de les enregistrer simultanément dans le cloud et de collaborer en ligne. Nous l'avons utilisé afin de modéliser nos différents diagrammes autant pour l'analyse, la conception que le déploiement.

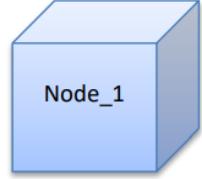
II- MISE EN PLACE : LE DÉPLOIEMENT

II.1) Diagramme de déploiement

II.1.1) Formalisme

Le diagramme de déploiement peut être mis en œuvre dès la phase de conception pour documenter l'architecture physique du système ou plus tard en fonction du bien vouloir du développeur.

Tableau 19 : formalisme du diagramme de composants

Élément	Description	Représentation
Les composants	Les objets sont les instances de classe et sont rangés horizontalement.	
Les nœuds	Un nœud représente l'ensemble d'équipements matériels du système. Cette entité est représentée par un cube tridimensionnel.	
Les dépendances	Une dépendance est utilisée pour mobiliser la relation entre deux composants. La notion utilisée pour cette relation de dépendance est une flèche de pointilles.	
Association	Relier les nœuds	

II.1.2) Diagramme de déploiement

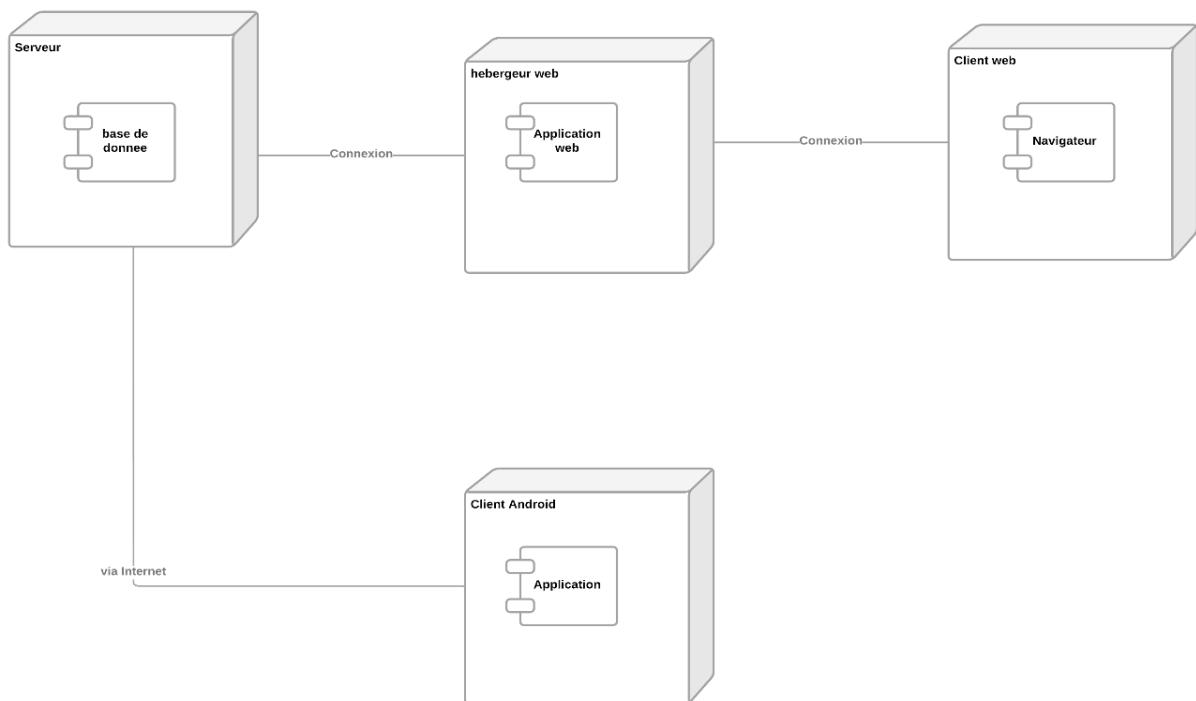


Figure 25 : Diagramme de déploiement

II.2) Diagramme de composants

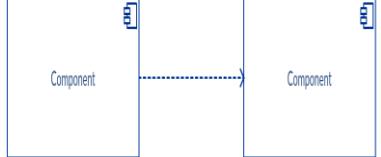
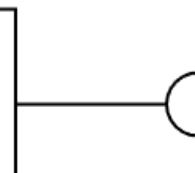
II.2.1) Formalisme

Les diagrammes de composants UML représentent les **relations entre les différents composants d'un système dans une vue d'ensemble statique**. Ils peuvent inclure des aspects de modélisation à la fois logiques et physiques. Dans le contexte de l'UML, les composants sont des éléments modulaires d'un système. Les diagrammes de composants UML doivent communiquer :

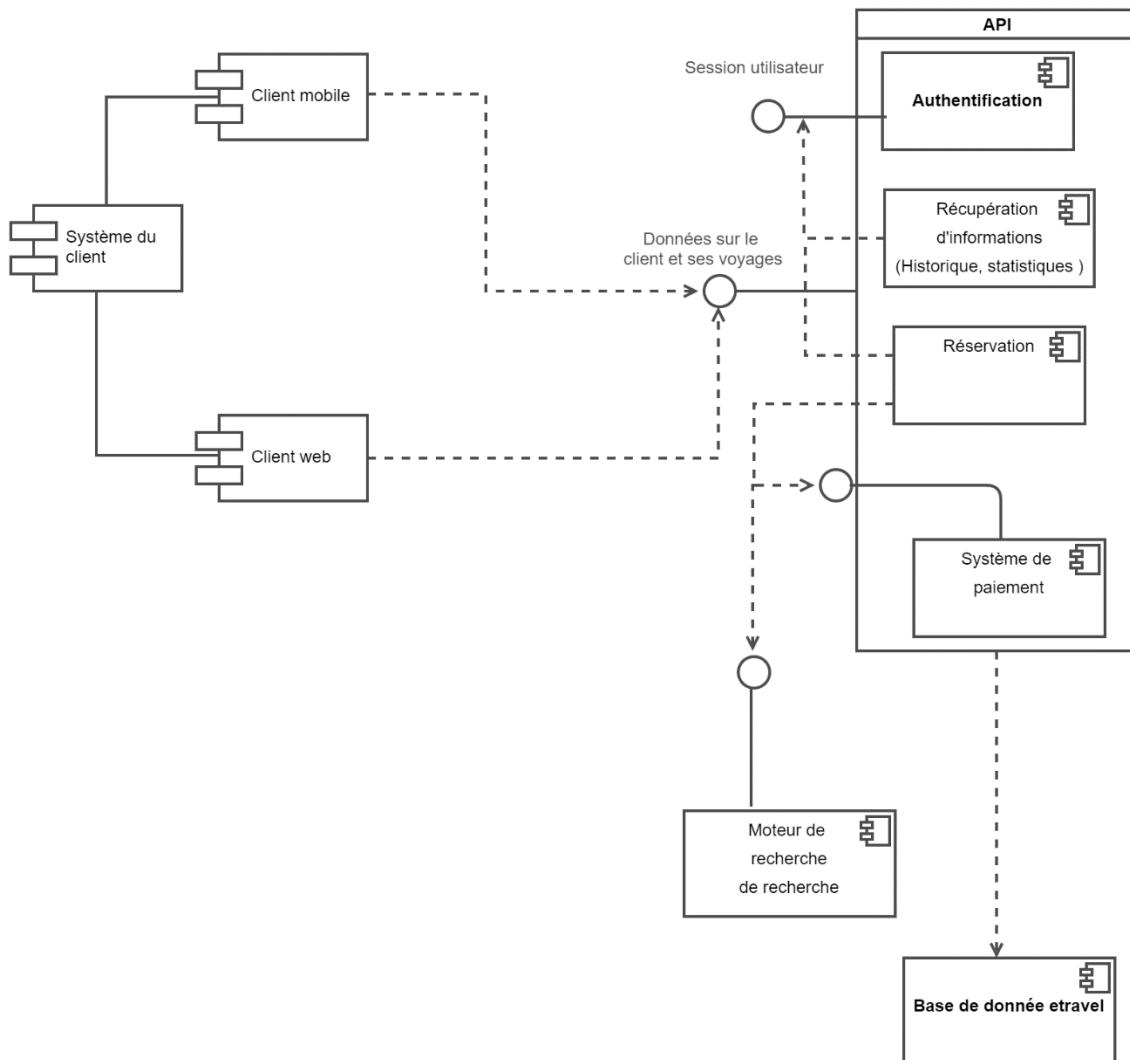
- la portée de votre système ;
- la structure globale de votre système logiciel ;
- les objectifs que le système permet aux entités humaines ou non humaines (appelées acteurs) d'atteindre.

Tableau 19 : formalisme du diagramme de composants

Composant	Description	Symbole
-----------	-------------	---------

Symbole de composant	<p>Entité tenue d'exécuter une fonction. Un composant fournit et consomme un comportement par le biais d'interfaces, ainsi que par le biais d'autres composants. En UML 1.0, un composant est modélisé sous forme de bloc rectangulaire avec deux rectangles plus petits qui dépassent sur le côté. En UML 2.0, un composant est modélisé sous forme de bloc rectangulaire avec une petite image de l'ancienne forme UML 1.0.</p>	<p>cône du composant et le stéréotype du composant.</p> 
Dépendances	<p>Indique les relations de dépendance entre les différentes parties de votre système. Les dépendances sont représentées par des lignes pointillées reliant un composant (ou élément) à un autre.</p>	
Interfaces fournies :	<p>une ligne droite s'étend à partir de la boîte du composant avec un cercle à l'extrémité. Ces symboles représentent les interfaces où un composant produit des informations utilisées par l'interface requise d'un autre composant.</p>	
Interfaces requises	<p>Une ligne droite s'étend à partir de la boîte du composant avec un demi-cercle à son extrémité (parfois également représenté par une ligne en pointillé avec une flèche ouverte). Ces symboles représentent les interfaces où un composant a besoin d'informations pour remplir sa fonction.</p>	

II.2.2) Diagramme de composants utilisateur et réservation



III- PRODUCTION DES PROGRAMMES

III.1) Descriptif du produit

Notre projet est constitué de 3 couches pour la version mobile et d'une couche web.

III.1.1) Version web

- Couche des clients
- Couche des administrateurs
- Couche des super-administrateurs

III.1.2) Version mobile

- Couche clients

III.2) Architecture

L'architecture de notre application est de type client-serveur, où un ordinateur/mobile interagit avec un serveur distant. La réalisation de notre projet, nous avons opté pour une architecture MVC afin de garantir une assurance de la maintenabilité, la modularité de l'application et la rapidité de développement.

MVC littéralement Modèle Vue Contrôleur est une architecture qui organise l'interface Homme-Machine d'une manière à ce que le développement puisse se faire en couches indépendantes. Il impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle de données, le contrôleur et la vue.

III.3) Charte graphique

III.3.1) Arrière-plan et texte

Nous avons opté pour un fond blanc. En effet, les sites conçus avec un fond blanc connaissent plus de succès que les autres. La plupart des moteurs de recherche ou des boutiques virtuelles s'affichent sur fond blanc. Le fond blanc renvoie au papier. Il conforte le lecteur et communique une impression de sérieux, de sobriété, de détachement et d'objectivité. Les images, elles permettent à l'utilisateur d'avoir plus d'illustration et le mettre d'avantage dans le contexte. Nous avons écrit le texte en noir ou en couleurs sombres, pour plus de confort pour le lecteur.

III.3.2) Usage des couleurs

Les couleurs étant un élément important pour définir l'esprit de la plateforme d'une part et conforter l'utilisateur d'autre part, nous avons choisis :

- **Bleu** : Souvent signe de confiance, la fiabilité et la tranquillité. Ses différentes teintes nous aiderons à créer une sensation de sécurité et de professionnalisme dans notre application de réservation de billets de bus.
- **Gris** : Le gris est une couleur neutre et polyvalente qui peut être utilisée pour créer un aspect moderne et élégant. Nous l'utiliserons comme couleur d'accent et parfois comme fond pour mettre en valeur d'autres éléments de conception.
- **Blanc** : Le blanc est une couleur propre et minimaliste qui peut être utilisée pour créer une sensation de simplicité et de clarté. Il sera souvent utilisé comme couleur de fond

pour mettre en valeur d'autres éléments de conception et couleur d'accent dans le cas du mode « Dark. ».

II.3.3) Charte graphique stable et robuste

Notre charte graphique est basée sur la technique des feuilles de style CSS ce qui la rend facile à manipuler :

- Il vous suffira de changer les définitions de style figurant dans les feuilles de style CSS pour que la présentation de notre site soit mise à jour.
- Le nombre de page web en ligne n'est pas statique (on peut changer le contenu textuel et visuel du site et notre charte graphique reste stable et robuste).
- Le mode nuit pour réduire la lumière forte et protéger les yeux des utilisateurs.

Conclusion

En définitive, la réalisation et le déploiement d'une plateforme nécessite une attention particulière au niveau des outils à utiliser; pour produire de la qualité, Il est essentiel, de faire de bons choix.

DOSSIER 6 : TESTS DE FONCTIONNALITES

Introduction

Lors du développement d'une plateforme, il est essentiel de garantir un bon fonctionnement et une expérience utilisateur optimale. Les tests de fonctionnalités jouent un rôle crucial dans cet objectif en permettant de vérifier la conformité aux exigences du cahier des charges, d'évaluer l'ergonomie et l'expérience utilisateur, et de détecter d'éventuels bugs ou problèmes.

Ce chapitre est consacré à la présentation des tests de fonctionnalités que nous avons réalisés dans le cadre de la mise en place de cette plateforme. Nous avons suivi une méthodologie rigoureuse et mis en œuvre différentes techniques de test pour garantir la qualité et la fiabilité de notre application ; techniques que nous détaillerons dans les sections suivantes.

I. Pourquoi les tests de fonctionnalités?

Comme évoqué précédemment, les tests de fonctionnalités constituent un élément essentiel de la mise en place d'un système. Ceux-ci permettent de garantir que notre solution comble les exigences préalablement établis (dans le cahier de charge); et couvre les possibles erreurs de l'utilisateur (il ne faut jamais lui faire confiance) tout en procurant au destinataire final une interface respectant les principes de l'UI et de l'UX tels que l'intuitivité, la facilité, l'harmonie de couleurs etc.; afin de garantir une expérience optimale sur le plan fonctionnel et visuel. Tout ceci en simulant un environnement assez similaire à l'environnement de production pour repérer les potentielles erreurs et les corriger.

II. Analyse des exigences

Il a été défini dès le cahier de charge, un ensemble de fonctionnalités requises pour notre plateforme de réservation en même temps que des exigences non fonctionnelles à l'instar de l'ergonomie, d'une interface facile d'utilisation, etc. Les plus importantes à tester sont l'inscription et la connexion, la recherche et la réservation des billets, l'ajout d'un voyage.

III. Conception des cas de test

Les fonctionnalités à tester dans une application peuvent si elles sont mal intégrées, produire un mauvais résultat (sans donner d'indication à l'utilisateur); et ce, de plusieurs manières. Il est donc important de mettre sur pieds des scenarios de tests liés à chacune des manières.

A- Authentification

1. Cas de test : Authentification réussie

- **Description :** Vérifier que l'utilisateur peut s'authentifier avec des identifiants valides.
- **Prérequis :** Avoir un compte utilisateur enregistré dans le système.
- **Étapes :**
 1. Ouvrir l'application et accéder à l'écran d'authentification.
 2. Saisir l'identifiant et le mot de passe corrects.
 3. Cliquer sur le bouton de connexion.
- **Résultat attendu :** L'utilisateur est redirigé vers la page d'accueil ou une autre page appropriée après une authentification réussie.

2. Cas de test : Authentification échouée en raison d'identifiants incorrects

- **Description :** Vérifier que l'utilisateur reçoit un message d'erreur approprié en cas d'identifiants incorrects.
- **Prérequis :** Avoir un compte utilisateur enregistré dans le système.
- **Étapes :**
 1. Ouvrir l'application et accéder à l'écran d'authentification.
 2. Saisir un identifiant ou un mot de passe incorrect.
 3. Cliquer sur le bouton de connexion.
- Résultat attendu : L'utilisateur reçoit un message d'erreur indiquant que les identifiants sont incorrects et reste sur la page d'authentification.

3. Cas de test : Authentification échouée en raison d'un compte inexistant

- **Description :** Vérifier que l'utilisateur reçoit un message d'erreur approprié en cas de compte inexistant.
- **Prérequis :** Aucun compte utilisateur enregistré dans le système.
- **Étapes :**
 1. Ouvrir l'application et accéder à l'écran d'authentification.
 2. Saisir un identifiant et un mot de passe valides.
 3. Cliquer sur le bouton de connexion.
- Résultat attendu : L'utilisateur reçoit un message d'erreur indiquant que le compte n'existe pas et reste sur la page d'authentification.

4. Cas de test : Déconnexion réussie

- **Description :** Vérifier que l'utilisateur peut se déconnecter avec succès de son compte.
- **Prérequis :** Être connecté à un compte utilisateur.
- **Étapes :**
 1. Ouvrir l'application et accéder à la page de profil ou de paramètres.
 2. Cliquer sur le bouton de déconnexion.
- Résultat attendu : L'utilisateur est déconnecté et redirigé vers la page d'authentification ou une autre page appropriée.

B- Reservation

1. Cas de test : Réservation réussie

- **Description :** Vérifier que l'utilisateur peut effectuer avec succès une réservation

- **Prérequis :** Être connecté à un compte utilisateur(client).
- **Étapes :**
 1. Ouvrir l'application et accéder à la page de réservation
 2. Rechercher un voyage en entrant des données valides
 3. Choisir le voyage à réserver.
 4. Entrer les informations du voyageur et de paiement
 5. Valider la réservation
 6. Valider le paiement
- Résultat attendu : La réservation est faite et l'utilisateur reçoit un message de confirmation.
 1. **Cas de test : Réservation échouée en cas de non-paiement**
- **Description :** Vérifier reçoive un message d'erreur en cas de non-paiement
- **Prérequis :** Être connecté à un compte utilisateur (client).
- **Étapes :**
 1. Ouvrir l'application et accéder à la page de réservation
 2. Rechercher un voyage en entrant des données valides
 3. Choisir le voyage à réserver.
 4. Entrer les informations du voyageur et de paiement
 5. Valider la réservation
 6. Ne pas valider le paiement
- Résultat attendu : L'utilisateur reçoit un message d'erreur sur l'annulation de la requête et est renvoyé vers la page précédente.

DOSSIER 7 : GUIDES D'INSTALLATION ET D'UTILISATION

Introduction

Dans ce chapitre dédié aux guides d'installation et d'utilisation d'Etravel, nous allons fournir toutes les informations nécessaires pour installer facilement et l'utiliser de manière efficace. L'installation d'une application peut parfois être un processus complexe, c'est pourquoi nous avons préparé des instructions détaillées pour vous guider tout au long du processus. De plus, nous avons également préparé un guide d'utilisation complet qui vous aidera à tirer le meilleur parti de notre plateforme.

L'objectif de ce chapitre est de vous permettre d'installer rapidement l'application sur votre appareil, de vous familiariser avec ses différentes fonctionnalités et de vous donner des conseils pour une utilisation **optimale**. Que vous soyez un utilisateur novice ou expérimenté, ces guides vous aideront à explorer toutes les possibilités offertes par notre plateforme de réservation de billets de transport interurbain. Nous vous encourageons à suivre attentivement les instructions fournies et à consulter régulièrement ce chapitre en cas de besoin. Nous espérons que ces guides d'installation et d'utilisation vous permettront de profiter pleinement d'Etravel. N'hésitez pas à plonger dans les sections suivantes pour commencer votre expérience avec notre application !

Si vous avez des questions supplémentaires, n'hésitez pas à nous contacter ou à consulter la FAQ. Nous sommes là pour vous aider.

I- Guide d'installation

Notre plateforme étant web et mobile, les nécessités pour l'utiliser sont relativement mineures. Nous présenterons les exigences à respecter par votre appareil avant de pouvoir profiter de la plateforme E-travel.

- ✓ Pour la plateforme Android, il suffit d'avoir un téléphone mobile avec le système Android installé version 7.1 Minimum (**Nougat**).

Pour la plateforme web, il est nécessaire de

- ✓ disposer d'un ordinateur (pc ou mac), d'une tablette ou d'un téléphone mobile supportant les dernières versions de navigateurs modernes à l'instar de **Mozilla Firefox, Google chrome, Safari etc.** (**un seul navigateur suffit**). La compatibilité avec un système d'exploitation particulier n'est pas nécessaire, étant donné que notre plateforme est hébergée sur un serveur sur le web.
- ✓ Avoir une connexion à internet.
- ✗ Internet explorer est proscrit, n'embarquant pas les spécificités nécessaires pour une expérience utilisateur optimale.

II- Guide d'utilisation

II.1) Utilisation générale

II.1.1) Vérifier et modifier mes informations

Si vous souhaitez consulter ses informations enregistrées (nom d'utilisateur ou adresse email), Modifier votre mot de passe pour une raison ou une autre, vous pouvez suivre les étapes suivantes.

1. Accéder à votre espace de travail, puis à l'onglet « informations personnelles » Vous pouvez le faire soit avec le bouton « Mes informations » sur le menu :



II.2) Pour les Clients

Les principales fonctionnalités auxquelles vous pouvez accéder en tant que client sont :

- Inscription
- Connexion à votre compte
- Consultation de la liste de vos tickets
- Faire une réservation

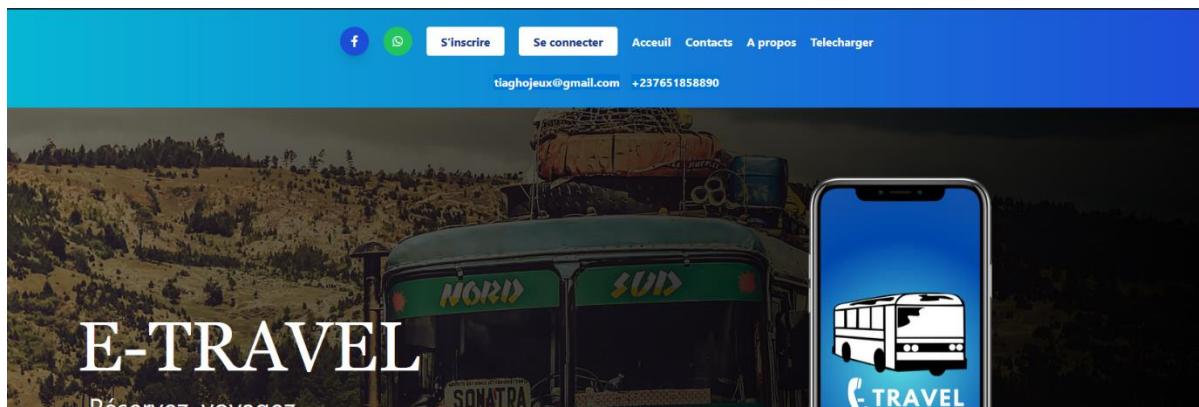
II.2.1) Inscription et connexion

2. Rendez-vous sur le site <https://etravel.fun> :

Si vous souhaitez vous inscrire, passez à l'étape 1.a ; si vous souhaitez vous connecter, passez directement à l'étape 3.

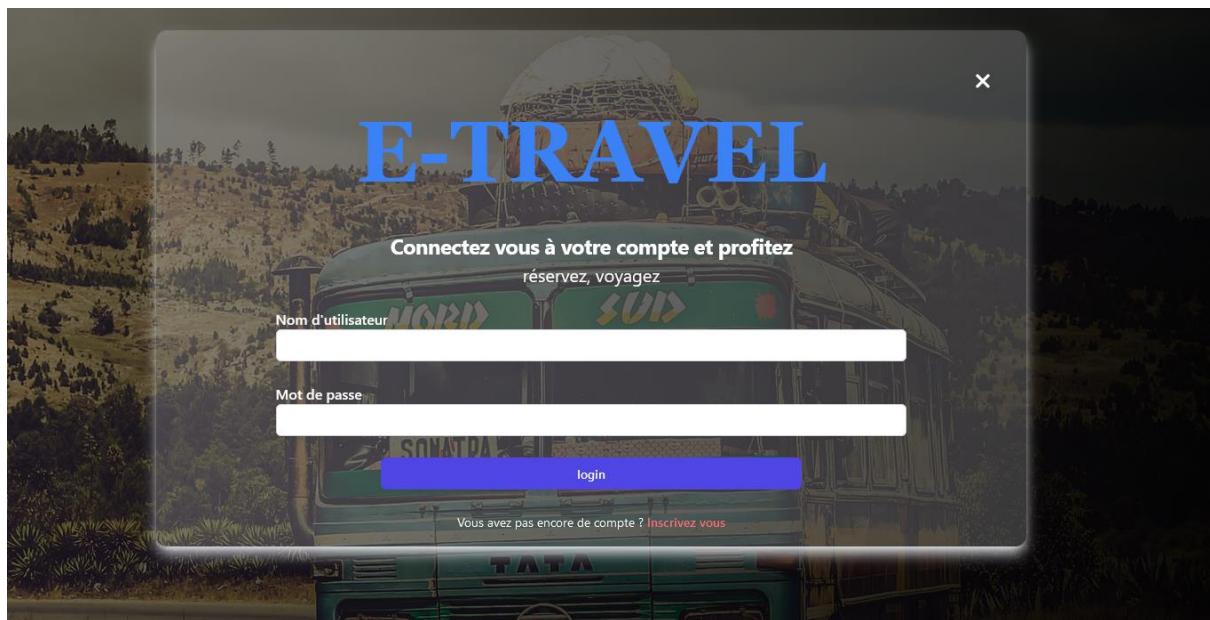
1. a- Si vous êtes sur ordinateur, tapez sur le bouton « s'inscrire » ou sur « je m'inscris ».

1. b- Si vous êtes sur téléphone, tapez sur le bouton de barre de menu situé dans le coin supérieur gauche et cliquez sur le bouton « s'inscrire ».



3. Dans le formulaire qui d'affiche à vous, veuillez remplissez les informations pour vous inscrire : Nom d'utilisateur (sans espaces de préférences) ; Votre nom complet, votre adresse email (facultative), Votre mot de passe (que vous allez confirmer dans le champ suivant). Vous serez ensuite redirigés vers la page de connexion, passez à l'étape 3.

4. Si vous n'êtes pas à la page de connexion, retournez à l'accueil du site et cliquez sur le bouton « se connecter» près du bouton « s'inscrire ». Dans le formulaire de connexion qui s'affiche, remplissez votre nom d'utilisateur et votre mot de passe entrés lors de votre inscription



Sur mobile les formulaires de connexion et d'inscription se présenteront différemment du web :

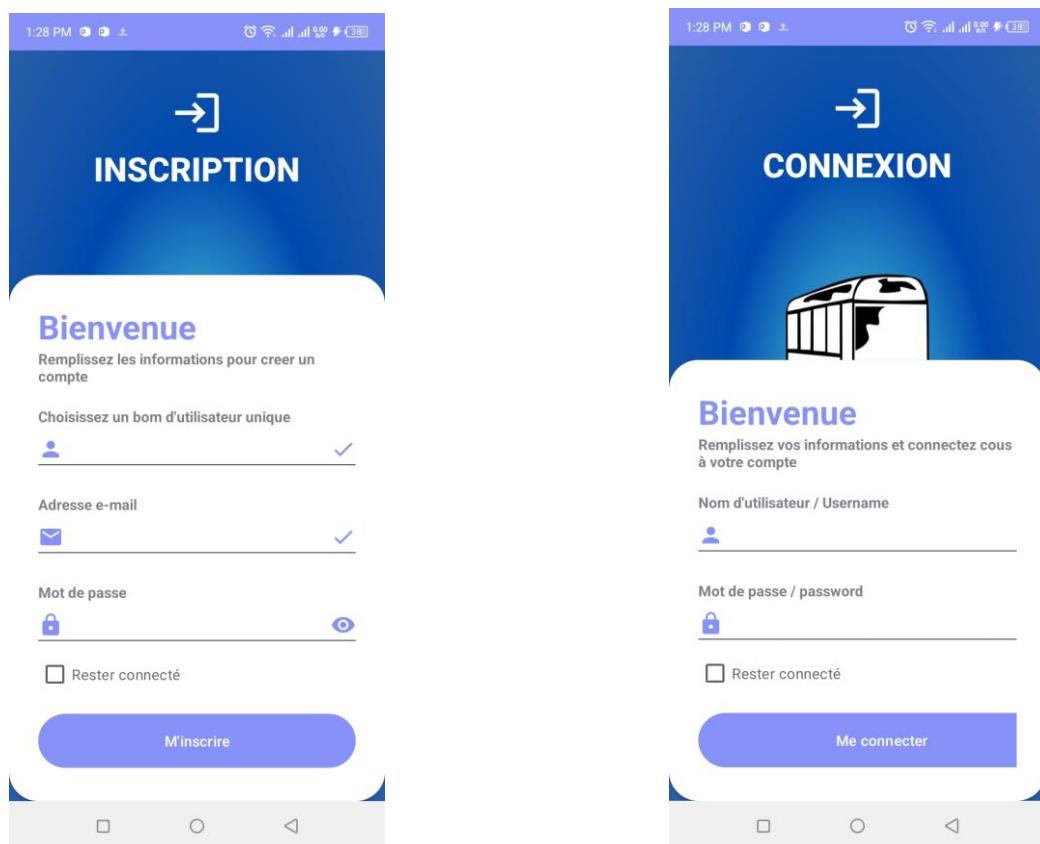


Figure 03 : Vues de l'inscription et connexion sur application mobile

Une fois votre connexion validée, vous serez redirigés vers votre espace de travail.

Si vous êtes sur mobile, le menu disparait place à une icône (trois barres verticales) sur laquelle vous pourrez cliquer pour faire apparaître ou disparaître le menu.

II.2.2) Réservation

1. Afin d'effectuer une réservation, sur l'onglet « Faire une réservation » du menu ; ensuite, remplissez le formulaire contenant les informations du voyage que vous souhaitez trouver (date de départ, heure de départ (une estimation),
2. Une fois le formulaire rempli et envoyé, vous recevrez la liste des voyages disponibles pour vos critères ! Cliquez sur le bouton « Réserver » du voyage qui vous intéresse.

Agence :	MT Travel	Agence :	Beute voyage	Agence :	trésor voyage
Depart :	yaoundé	Depart :	yaoundé	Depart :	yaoundé
arrivée :	douala	arrivée :	douala	arrivée :	douala
Secteur :	Biyemassi	Secteur :	Biyemassi	Secteur :	Biyemassi
Date :	2023-09-25	Date :	2023-09-25	Date :	2023-09-25
Heure :	08:30:00	Heure :	08:30:00	Heure :	08:30:00
Type :	classic	Type :	VIP	Type :	classic
Prix :	7500 Fcfa	Prix :	14700 Fcfa	Prix :	7500 Fcfa
Réserver		Réserver		Réserver	

3. Remplissez ensuite vos informations (Nom complet du voyageur, numéro du siège), numéro MTN Mobile money ou Orange money pour le paiement.

NB : Le numéro doit débuter par 237 ex : 237654896630.

II.2.3) Consultation des anciens tickets ou ticket actuels

II.3) Pour les administrateurs

II.1.1) Connexion

Les administrateurs des différentes agences ne peuvent pas d'eux même créer de compte, mais ils peuvent se faire en créer un à la demande de l'agence. Pour la connexion au compte, ils doivent se rendre à l'adresse web que leur donnera le super-administrateur, et entrer leurs informations de connexion.

II.3) FAQ

Ici vous trouverez des questions fréquemment posées sur notre système :

a. E-travel est-il payant

Les services de E-travel sont totalement gratuits, les seuls frais que vous payez sont ceux correspondant au tarif d'un ticket que vous souhaitez réserver.

b. Quels modes de paiement sont actuellement acceptés ?

Avec etravel, vous pouvez payer par MTN Mobile money ou Orange money, mais nous prévoyons d'ajouter d'autres services de paiement.

c. Que faire si je rencontre un problème avec une agence ou avec e-travel ?

Vous pouvez si vous rencontrez un problème, vous rendre sur l'accueil de la plateforme et cliquer sur le bouton « nous contacter », puis, remplissez les informations du formulaire, (votre

nom, adresse mail ou numéro de téléphone, l'agence concernée, le message à envoyer) ; enfin, cliquez sur le bouton « envoyer ».

d. Comment puis-je effectuer une réservation ?

Pour effectuer une réservation, cliquez sur le bouton « faire une réservation de votre menu » et suivez les étapes du « **II.1.2)** »

CONCLUSION GÉNÉRALE

BIBLIOGRAPHIE

Afin de mener à bien notre projet et la rédaction de ce rapport, nous avons fait usage des documents ci-après :

- Cahier de Développement web deuxième année IAI Bafoussam,
- Cahier de cours UML 2eme année,
- Cahier des charges GL2 et SR2,
- Cahier de rédaction scientifique 2eme année,
- Rapport de fin de stage sur le thème,

WEBOGRAPHIE

Les ressources web utilisées sont les suivantes :

- <https://www.lucidchart.com>** : Approfondissement des connaissances sur le formalisme UML et ses diagrammes.
- <https://www.wikipédia.com>** : Ressources multiples et variées destinées à une connaissance générale sur les langages, les concepts, les méthodes, etc.
- <https://www.tailwindcss.com>** : Documentation de la librairie, nous aidant à saisir les concepts fondamentaux de l'outil; mais aussi les concepts plus avancés.
- <https://www.laravel.com>** : Documentation du framework laravel, nous aidant à saisir les concepts fondamentaux du framework; mais aussi les concepts plus avancés.

TABLE DE MATIÈRES

DÉDICACE	2
REMERCIEMENTS.....	i
AVANT-PROPOS	Erreur ! Signet non défini.
SOMMAIRE.....	ii
RÉSUME	Erreur ! Signet non défini.
ABSTRACT.....	Erreur ! Signet non défini.
LISTE DES ABREVIATIONS.....	Erreur ! Signet non défini.
LISTE DES FIGURES	iii
LISTE DES TABLEAUX.....	Erreur ! Signet non défini.
INTRODUCTION GÉNÉRALE	1
PARTIE 1 : RAPPORTE D'INSERTION.....	2
CHAPITRE 1 : PRESENTATION DE L'ENTREPRISE	3
I- PRESENTATION GENERALE DE LA STRUCTURE	3
I.1) Historique	3
I.2) Situation géographique.....	3
II- MISSIONS ET ORGANES DE GESTION	4
II.1) Services	4
II.2) Missions	4
II.3) Organes de gestion	5
CHAPITRE 2 : DÉROULEMENT DU STAGE	6
I- ACCUEIL	6
II- TÂCHES EFFECTUÉES.....	6
III- ÉVALUATION.....	6
III.1) Difficultés	6
III.2) Bilan de l'expérience	7
PARTIE 2 : CAHIER DES CHARGES	8

CHAPITRE 1 : DOSSIER D'ÉTUDE	12
I- CONTEXTE ET JUSTIFICATION DU PROJET.....	Erreur ! Signet non défini.
II- ÉTUDE DE L'EXISTANT	9
II.1) Présentation.....	Erreur ! Signet non défini.
II.2) Point fort	10
II.3) Point faible	10
II.4) Proposition de solution.....	Erreur ! Signet non défini.
III- LES OBJECTIFS DU PROJET	Erreur ! Signet non défini.
III.1) Objectif pour l'administration	13
III.2) Objectif pour le client	13
III.3) Objectif pour les collecteurs	Erreur ! Signet non défini.
IV- EXPRESSIONS DES BESOINS DE L'UTILISATEUR	13
IV.1) Besoin fonctionnel du projet	13
IV.2) Au niveau du client.....	13
IV.3) Au niveau du collecteur.....	13
IV.4) Au niveau de l'administrateur	14
IV.5) Besoin non fonctionnel du projet	14
IV.6) Fiabilité.....	14
IV.7) Ergonomie et bonne interface.....	14
IV.8) Sécurité.....	14
IV.9) Aptitude à la maintenance et la réutilisation	14
IV.10) L'intégrité.....	14
IV.11) Disponibilité : 24h/24 et 7j/7 BESOINS	14
V- ESTIMATION DU COÛT DU PROJET ET EQUIPE DE TRAVAIL.....	15
V.1) Équipe de travail	15
V.2) Resource matérielles	16
V.3) Ressources humaines	16
V.4) Grand total	16
VI- PLANIFICATION DU PROJET	14

VII- LES CONTRAINTES DU PROJET	17
CHAPITRE 2 : DOSSIER CONCEPTION	19
I- PRÉSENTATION DE LA MÉTHODE 2TUP	Erreurs ! Signet non défini.
II- CONCEPTION	Erreurs ! Signet non défini.
III- PRÉSENTATION DE UML.....	Erreurs ! Signet non défini.
III.1) Diagramme fonctionnel	Erreurs ! Signet non défini.
III.1.1) Diagramme des cas d'utilisation.....	Erreurs ! Signet non défini.
III.1.2) Diagramme de cas d'utilisation du client	Erreurs ! Signet non défini.
III.1.3) Diagramme de cas d'utilisation du collecteur.....	27
III.1.4) Diagramme de cas d'utilisation de l'administrateur	28
III.1.5) Diagramme de cas d'utilisation global	Erreurs ! Signet non défini.
III.1.6) Description des cas d'utilisation	28
III.2) Modélisation conceptuelle des données.....	44
III.2.1) Dictionnaire des données.....	Erreurs ! Signet non défini.
III.2.2) Diagramme de séquence	Erreurs ! Signet non défini.
III.3) Quelques diagrammes de séquences du système.....	Erreurs ! Signet non défini.
III.3.1) Collecte d'un client.....	Erreurs ! Signet non défini.
III.3.2) Création d'un utilisateur	Erreurs ! Signet non défini.
III.3.3) Authentification	Erreurs ! Signet non défini.
III.4) Représentation des classes	Erreurs ! Signet non défini.
CHAPITRE III : DOSSIER DE RÉALISATION.....	Erreurs ! Signet non défini.
I- L'ÉTUDE TECHNIQUE	Erreurs ! Signet non défini.
I.1) Environnement de réalisation.....	Erreurs ! Signet non défini.
I.1.1) Matériels de base :	Erreurs ! Signet non défini.
I.1.2) Choix des langages de développement et de SGBD :	Erreurs ! Signet non défini.
I.1.3) Application web	Erreurs ! Signet non défini.
I.1.4) Outil de développement.....	Erreurs ! Signet non défini.
I.2) Modélisation physique des données	Erreurs ! Signet non défini.
II- PRODUCTION DES PROGRAMMES	56

II.1) Descriptif du produit	56
II.1.1) Version mobile.....	56
II.1.2) Version web	56
II.2) Architecture.....	57
II.3) Charte graphique	57
II.3.1) Arrière-plan et texte	57
II.3.2) Usage des couleurs.....	57
II.3.3) Charte graphique stable et robuste	58
III- PHASE DE DÉPLOIEMENT.....	Erreur ! Signet non défini.
III.1) Diagramme de déploiement.....	Erreur ! Signet non défini.
III.1.1) Formalisme	Erreur ! Signet non défini.
III.1.2) Diagramme de déploiement	Erreur ! Signet non défini.
PARTIE III – BILAN	Erreur ! Signet non défini.
CHAPITRE 1 : PRÉSENTATION DE L'APPLICATION	Erreur ! Signet non défini.
I- LES VUES	Erreur ! Signet non défini.
I.1) Version mobile	Erreur ! Signet non défini.
I.2) Version web.....	Erreur ! Signet non défini.
I.2.1) Les modèles	Erreur ! Signet non défini.
CHAPITRE II – APPORTS	Erreur ! Signet non défini.
I- APPORTS AU NIVEAU DES CONNAISSANCES TECHNIQUESErrer	Erreur ! Signet non défini.
II- APPORTS AU NIVEAU DE LA CONCEPTION ET DU DÉVELOPPEMENTErre	Erreur ! Signet non défini.
III- ÉVALUATION.....	Erreur ! Signet non défini.
III.1) Bilan quantitatif	Erreur ! Signet non défini.
CONCLUSION GÉNÉRALE.....	69
WEBOGRAPHIE	71
TABLE DE MATIÈRES	72
ANNEXES.....	Erreur ! Signet non défini.

