**Ecole supérieure d'ingénierie en Sciences Appliquées**

**Rapport de Projet de Fin d’années pour l’obtention du**

**Diplôme de l’ESISA**

BAC+3 (Filière : Ingénierie Logicielle)

Sous le Thème

**Développement d’une application Mobile de Coupe d'Afrique : Fusion de Passion et de Découverte - Un Voyage Virtuel avec l'Application Tourisme Sportif**

**Réalisé par : Encadré par :**

BELHAJ Ikram Mr. IRAQI HOUSSAINI MEHDI

ABDELLAH Qatre En Nada Mr. MEKOUAR OTHMANE

DOUKAR Aya

**Jury :**

Mr. MEKOUAR Khalid

Mr. IRAQI HOUSSAINI MEHDI

Mr. MEKOUAR OTHMANE

Année universitaire : 2023 - 2024

**REMERCIEMENT**

Nous tenons à exprimer nos sincères remerciements à toutes les personnes qui ont contribué au succès de ce projet et qui nous ont apporté leur aide tout au long de la rédaction de ce rapport.

Tout d’abord, nous souhaitons adresser nos plus vifs remerciements a notre Professeur OTHMANE MEKOUAR, pour son soutien inestimable réalisation de ce projet. Ses conseils avisés et son encouragement constant ont été des éléments essentiels de notre parcours.

Nous tenons également a exprimer notre profonde gratitude envers notre Professeur et Encadrant, M. IRAQI HOUSSAINI, pour son encadrement de qualité, sa disponibilité et ses critiques constructives qui ont grandement enrichi notre travail. Nous lui sommes reconnaissants pour le temps précieux qu'il a consacré à guider notre réflexion et à nous aider à atteindre nos objectifs.

Enfin, nous tenons à remercier tous nos professeurs et le Directeur de notre établissement, M. MEKOUAR Khalid, ainsi que toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet. Leur soutien indéfectible et leur expertise ont été des éléments clés de notre réussite.

Nous sommes reconnaissants envers chacune de ces personnes pour leur contribution précieuse à notre développement professionnel et académique. Leur soutien continu a été une source d'inspiration tout au long de ce projet. Merci infiniment pour votre engagement et votre accompagnement.

Chapitre 1:

Présentation Générale

1. **Introduction**

Dans ce chapitre, nous présentons notre projet "Coupe d'Afrique Fusion de Passion et de Découverte ". Nous expliquons notre objectif d'offrir une expérience inoubliable aux fans de football visitant le Maroc pour la Coupe d'Afrique. Nous abordons la problématique du projet, décrivons les fonctionnalités principales de l'application, énumérons les livrables attendus et présentons le plan de développement en plusieurs phases. Ce chapitre pose les bases de notre projet en résumant ses objectifs, ses défis et sa planification prévue.

1. **Contexte du travail**

La Coupe d’Afrique de Football est bien plus qu’un simple événement sportif. C'est un rassemblement mondial où la passion pour le football se mêle à la découverte de nouvelles cultures et destinations. Dans ce contexte, notre projet prend vie : le développement d'une application mobile dédiée à la Coupe d'Afrique, fusionnant la passion du sport avec l'exploration touristique.

1. **Fonctionnalités principales**

Dans le cadre de la Coupe d’Afrique de Football 2025 au Maroc, notre application mobile cherche à offrir une expérience enrichissante qui combine la passion pour le sport et la découverte culturelle :

* Calendriers des évènements
* Recommandations de Restaurants et de Cafés
* Informations sur les options de transport
* Informations sur la culture locale
* Intégration de commentaires et d’avis
* Informations Pratique

1. **Livrables attendus**

L’achèvement réussi de ce projet est attendu avec la livraison des éléments suivants :

* Application mobile fonctionnelle : Une application conviviale sur iOS et Android pour gérer les réservations, consulter les prix des billets, obtenir des informations sur les villes hôtes et stades, et explorer les attractions locales.
* Calendrier des événements : Un calendrier complet des événements sportifs et culturels de la Coupe d'Afrique 2025 au Maroc.
* Système de commentaires et d’avis
* Documentation détaillée : Une documentation complète décrivant l'architecture, les fonctionnalités, les technologies utilisées et les instructions d'utilisation de l'application.
* Tests et validation : Des tests approfondis pour garantir la stabilité, la sécurité et le bon fonctionnement de l'application, ainsi que des validations utilisateur pour assurer son adéquation aux besoins.

1. **Planification du projet**

**Phase 1 : Planification et Conception (Semaines 1-2)**

* Définition des spécifications détaillées pour chaque fonctionnalité.
* Création de maquettes et de wireframes.
* Identification des ressources nécessaires.
* Élaboration d'un calendrier détaillé.

**Phase 2 : Développement (Semaines 3-6)**

* Développement des modules principaux : calendrier des événements, informations détaillées sur les événements, billetterie et réservations, recommandations de restaurants et de cafés, transport et itinéraires, informations sur la culture locale, intégration de commentaires et d'avis.

**Phase 3 : Tests et Finalisation** **(Semaines 7-8)**

* Tests approfondis sur chaque fonctionnalité.
* Test de l'interface utilisateur.
* Correction des bugs identifiés.
* Finalisation de l'application pour le déploiement.

**Phase 4 : Lancement** **(Semaine 9)**

* Déploiement sur les plateformes de téléchargement d'applications.
* Promotion du lancement.
* Suivi des retours des utilisateurs.

**Phase 5 : Maintenance et Amélioration Continue** **(Semaine 10 et au-delà)**

* Surveillance de l'utilisation et des retours des utilisateurs.
* Correction des bugs et des problèmes.
* Planification des mises à jour futures.

1. **Conclusion du chapitre**

En résumé, ce chapitre a jeté les bases de notre projet. Nous avons clairement expliqué notre objectif : créer une application pour la Coupe d'Afrique de Football 2025 au Maroc, offrant aux fans une expérience combinant sport et découverte. Nous avons présenté de façon concise les principales caractéristiques et objectives du projet, montrant notre engagement envers une application mobile utile et intéressante. Notre planification détaillée nous aidera à gérer efficacement notre temps et nos ressources. Ce chapitre est le point de départ de notre réussite.

Chapitre 2:

Contexte et Problématique

1. Cadre de projet
2. **Objectifs du projet (Description)**

L’objectif principal de notre projet est de développer une application mobile innovante dédiée à la Coupe d'Afrique de Football 2025 au Maroc. Cette application permettra de de réaliser les opérations suivantes :

* Faciliter la gestion des réservations d'hôtels et des billets
* Fournir des informations complètes sur les villes hôtes et les stades
* Promouvoir les attractions locales et la culture
* Offrir des calendriers des événements et des fonctionnalités pratiques
* Favoriser l'interaction et la rétroaction Assurer une expérience utilisateur optimale

1. **Cahier de charge**

Afin de bien gérer notre application et d’aboutir à des résultats qui répondent aux besoins, un cahier des charges a été établi dès le début de notre projet. Cela nous a aidés à nous orienter vers les bons axes pendant notre période d'apprentissage de cette nouvelle technologie.

1. **Problématique**

La problématique centrale de notre projet se formule comme suit :

Comment concevoir et développer une application mobile innovante qui permettra aux supporters de football voyageant au Maroc pour la Coupe d'Afrique de gérer facilement leurs réservations d'hôtels, de consulter les prix des billets, d'obtenir des informations sur les villes hôtes et les stades, et d'explorer les attractions locales, afin de vivre une expérience fluide et inoubliable pendant le tournoi ?

1. **Solution**

Pour répondre à la problématique, nous avons pensé à une application mobile super pratique. Avec cette application, les utilisateurs pourront acheter leurs billets, découvrir les endroits sympas à visiter, le tout avec des conseils personnalisés. Notre but ? Rendre leur séjour le plus agréable possible en regroupant toutes les infos et services dont ils ont besoin au même endroit.

1. Planification du projet
2. **Introduction**

Afin d’accomplir notre projet, l’adoption d’une démarche pour gérer le cycle de vie s’impose.

Dans cette section sera présentée la répartition des étapes de la réalisation de l’application.

1. **Auto-formation**

Pour arriver à achever cette mission il me fallait une auto-formation qui a pour rôle de prendre connaissance sur une nouvelle technologie comme React et Node JS l’utilisation de MongoDB.

1. **Diagramme de GANTT**

Conduire un projet, c’est assurer le pilotage d’un processus de changement avec des ressources dédiées en optimisant les compétences, l’organisation, les systèmes et les outils de conduite. Une approche managériale réactive, souple et systématique pour mener à bien des changements importants, complexes, ciblés sur le but à atteindre. Il y a trois niveaux de gestion du projet : La gestion de la production, des ressources et du temps. Ainsi, la planification du projet est devenue un outil managérial incontournable, elle fait partie des phases cruciales d’avant projet. Le planning est alors un outil de communication, de coordination, d’aide à la décision et d’anticipation dans les projets. Pour bien exprimer les différentes phases de ce projet, nous avons élaboré un diagramme de GANTT (souvent en complément d'un réseau PERT). A cet effet, j’ai fixé un planning, suivant lequel je pouvais déterminer quelle tâche sera faite à quel moment, et ainsi éviter le travail en désordre.

La figure ci-dessous montre le diagramme de Gantt effectué :

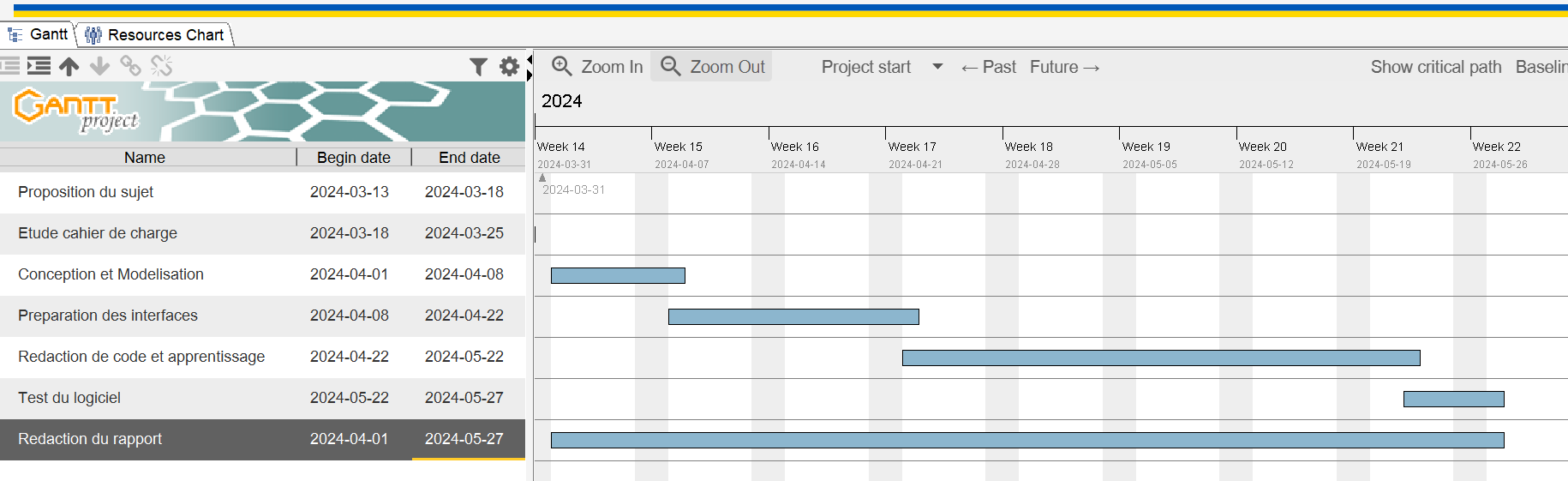


Figure 1 : Diagramme de GANTT

* Gantt Project

Un outil qui permet de gérer les projets sur le modèle des diagrammes de Gantt, ces graphiques peuvent d'ailleurs être imprimés, enregistrés aux formats PDF et HTML ou exportés dans Microsoft Project ou un tableur afin d'être distribués aux équipes concernées. L'application permet de décomposer vos projets en arborescence et d'assigner des ressources à chacune des tâches prévues au planning. Il est possible de créer des dépendances entre les activités. Cette fonctionnalité se révèle indispensable lorsque le travail accompli sur une tâche est nécessaire pour une autre partie du projet.

1. **Conclusion**

Cette partie décrit les besoins auxquels l’application devra répondre ainsi que les objectifs généraux à atteindre et les étapes de la réalisation. Le chapitre suivant sera consacré à la phase d’analyse et de conception.

Chapitre 3:

Analyse et Conception

1. Méthodologie Adoptée

Avant de programmer l'application mobile et de commencer à écrire le code, il est essentiel de passer par une phase de modélisation. Pour cela, nous avons utilisé le langage UML.

1. **Modèle en cascade**

Étant donné que les besoins auxquels doit répondre notre application mobile pour la Coupe d'Afrique sont stables et clairement identifiés dès le départ, nous avons choisi le modèle en cascade comme cycle de vie de développement. Ce modèle est jugé le plus adapté pour la conception de notre projet. Le principe du modèle en cascade est simple : on ne passe à la phase suivante que lorsque la précédente est validée.

Cette méthode présente de nombreux avantages, notamment celui de sécuriser la planification du projet puisque chaque étape est verrouillée avant de passer à la suivante. Voici les différentes phases du modèle en cascade appliquées à notre projet :

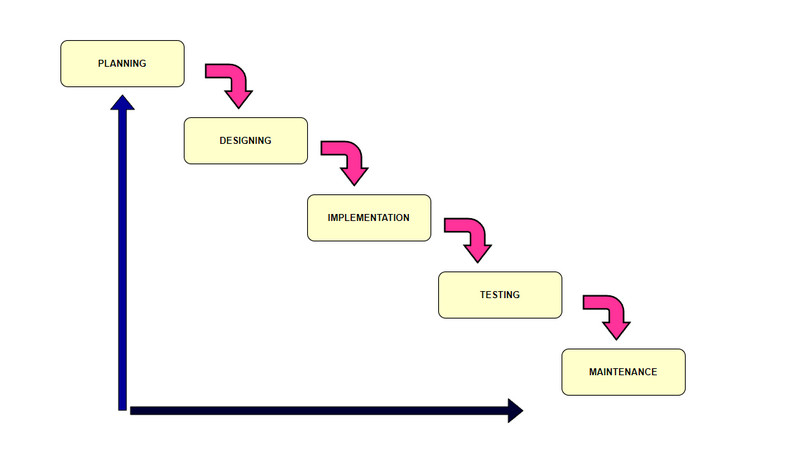


Figure 2 : Modèle en Cascade

1. **MVC**

Afin de clarifier l’architecture de note application et de simplifier la tâche du développement, nous avons choisi le patron MVC (Modèle-Vue-Contrôleur) qui est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Ce dernier regroupe les fonctions nécessaires en trois catégories :

* **Un Modèle** (modèle de données) :

Le modèle représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité. La base de données sera l'un de ses composants. Le modèle comporte des méthodes standards pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données. Les résultats renvoyés par le modèle ne s'occupent pas de la présentation. Le modèle ne contient aucun lien direct vers le contrôleur ou la vue.

* **Une Vue** (présentation, interface utilisateur) :

C’est avec quoi l'utilisateur interagit et se nomme précisément la vue. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toute action de l'utilisateur (hover, clic de souris, sélection d'un bouton radio, le fait de cocher une case, entrée de texte, de mouvements, de voix, etc.).

* **Un Contrôleur** (logique de contrôle, synchronisation) :

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de la vue et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification au modèle afin que les données affichées se mettent à jour. Le contrôleur n'effectue aucun traitement et ne modifie aucune donnée. Il analyse la requête du client et se contente d'appeler le modèle adéquat et de renvoyer la vue correspondant à la demande.

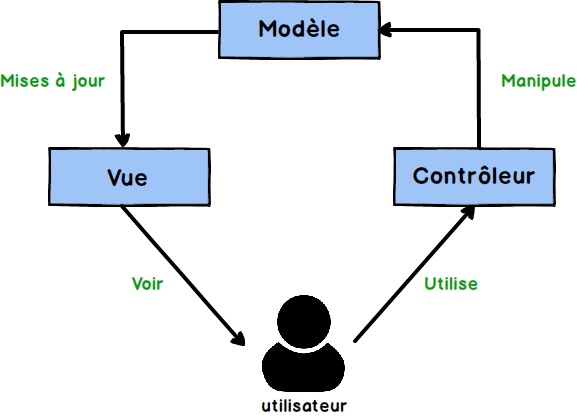
****

Figure 3 : Modèle mvc

1. **Langage UML (Outil de conception)**

Le langage de modélisation unifié (Unified Modeling Language) est un langage d'analyse, de conception et de modélisation orienté objet C’est une boite à outils qui permet d’améliorer progressivement une méthode de travail, tout en préservant un mode de fonctionnement. Le support logiciel choisi pour la conception et la modélisation UML est Enterprise architecte, jugé le support le plus complet avec tous les éléments, les relations et les diagrammes UML.

1. Modélisation du contexte

Dans cette partie, nous allons présenter l'ensemble des cas d'utilisation de notre application mobile pour la Coupe d'Afrique 2025. Pour atteindre notre objectif, il s'agit de répondre aux questions suivantes :

* Quels sont les utilisateurs du système ?
* Qu'attendent-ils de ce système ?

Pour cela, nous allons suivre la procédure suivante :

* Présentation des acteurs du système
* Lister les cas d'utilisation
* Donner une description textuelle pour chaque cas d'utilisation
* Construire les diagrammes de séquences

1. L’acteur du système

Dans système il y a deux acteurs qui sont :



Figure 4 : Modèle en Cascade

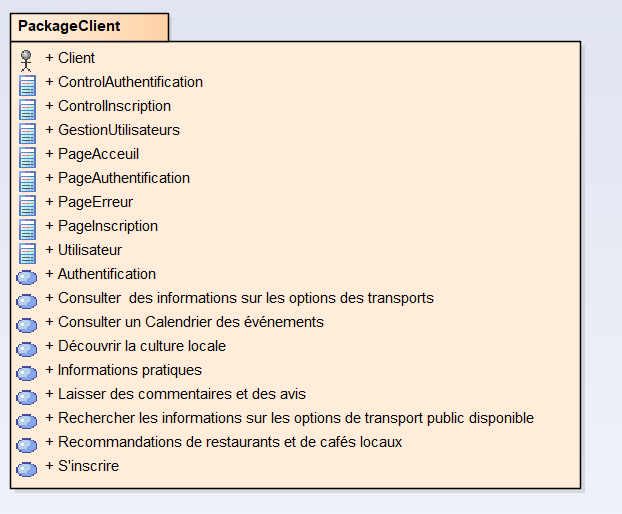
**Client**: Utilisateur principal de l'application, qui utilise les fonctionnalités pour acheter des billets et obtenir des informations locales.

1. Le diagramme de packages :

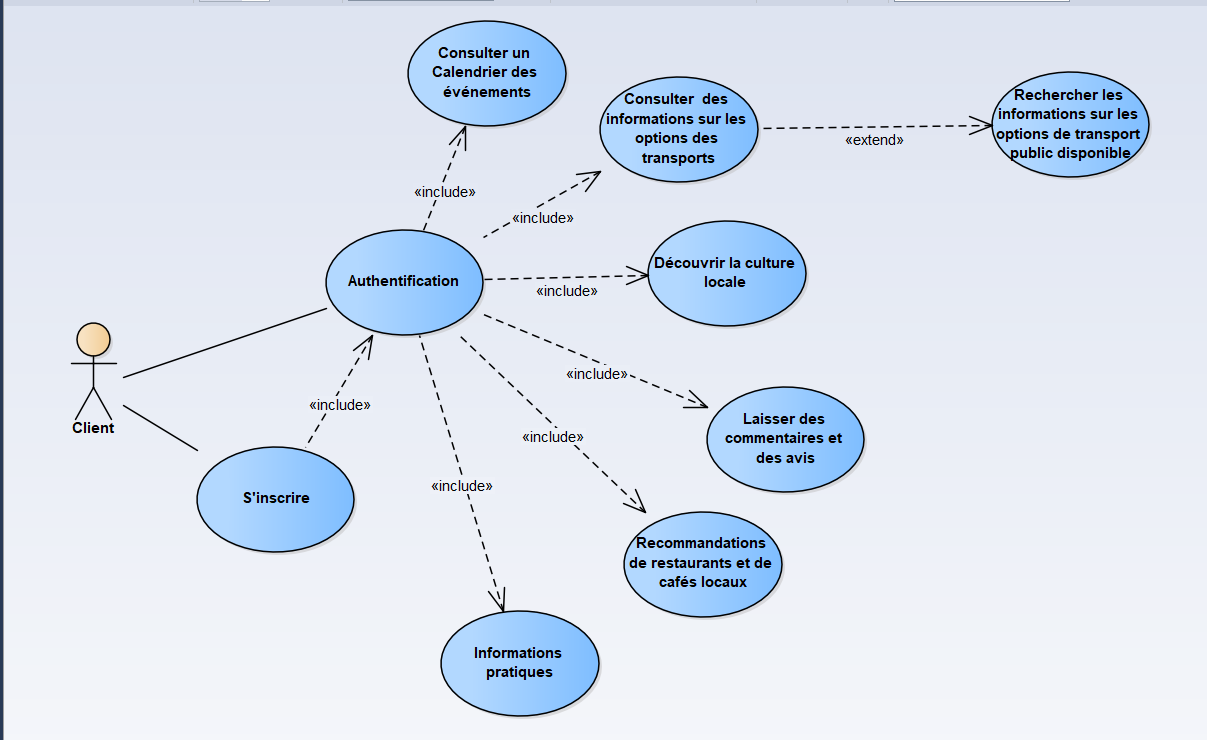
Les diagrammes de package (ou diagramme de paquetages) sont des diagrammes structurels utilisés pour représenter l'organisation et la disposition de divers éléments modélisés sous forme de paquetages.

Dans ce cas on a un seul packages :

* Client



1. Diagramme de cas d’utilisation :

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. 

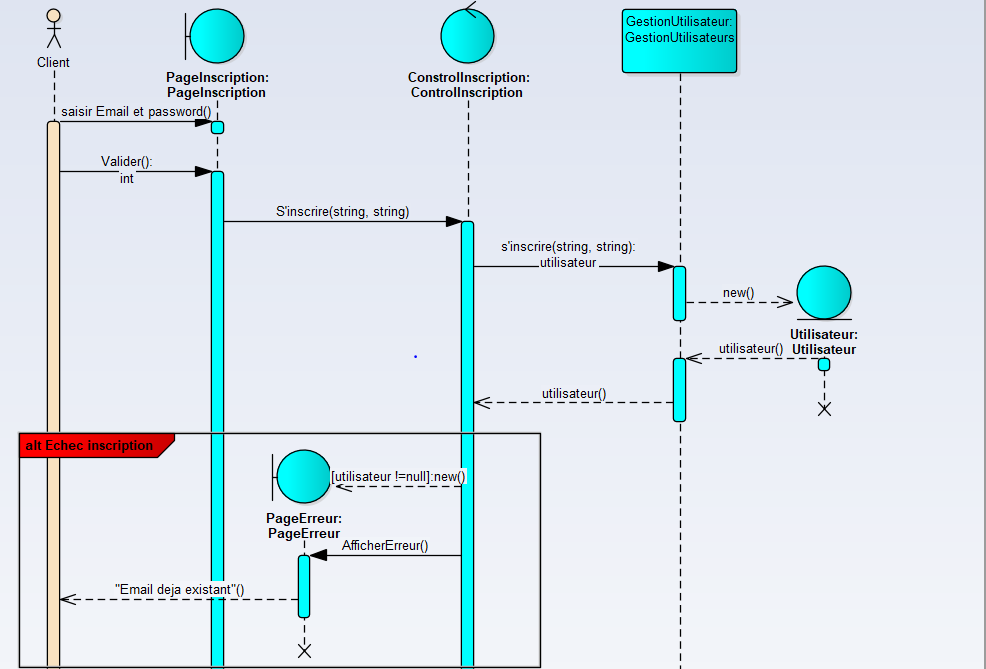
1. Diagramme de séquence :

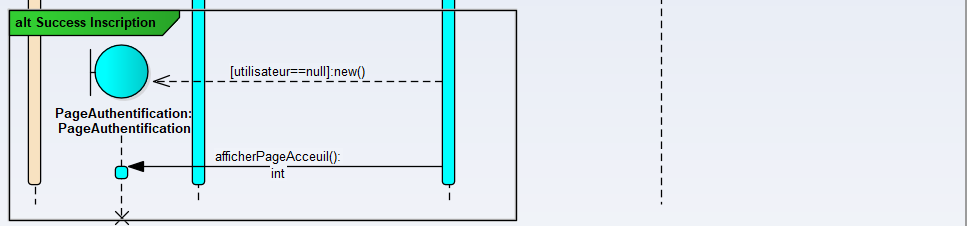
Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unifie Modeling Language.

On présentera quelques diagramme de séquence de la solution.

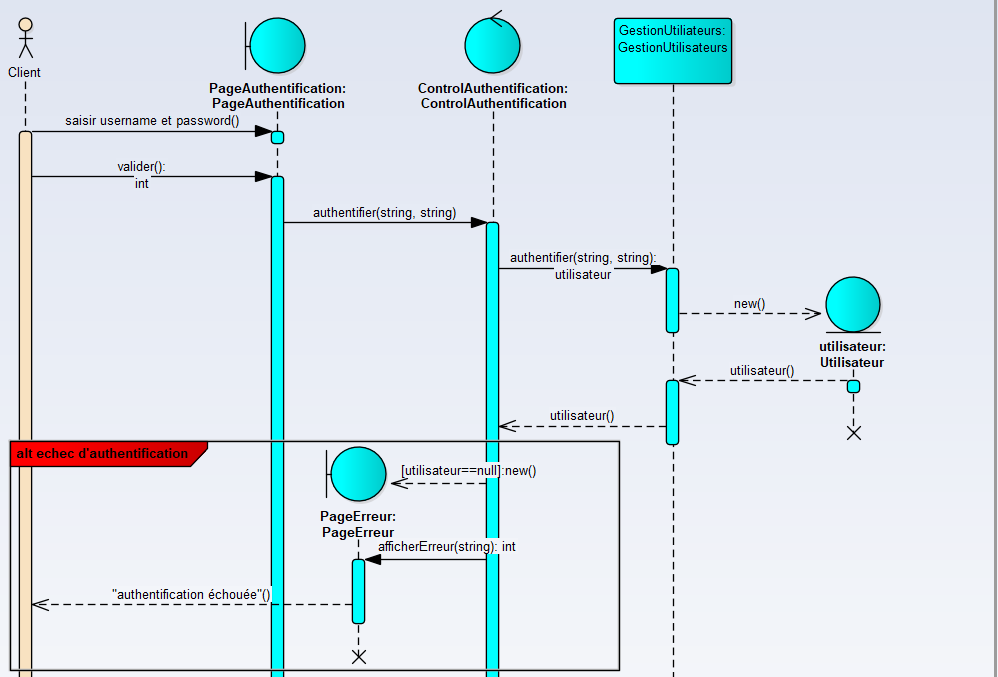
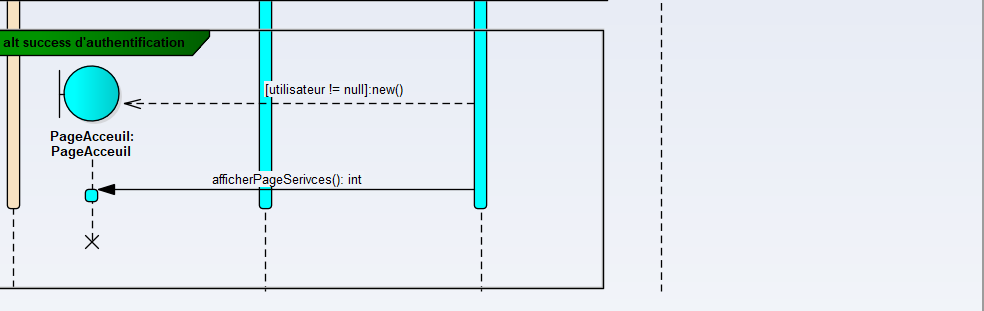
1. Diagramme de séquence <<inscription>>

* Le client saisie ses informations.
* Un message d’erreur s’affiche si les informations sont incorrectes.
* Le compte est créé avec succès si les informations sont bien saisies.





b) Diagramme de séquence <<authentification>>

* Le client saisie ses informations.
* Un message d’erreur s’affiche si les informations sont incorrectes.
* Si les informations sont bien saisies, le client accède à l'application avec toutes ses fonctionnalités.

1. Conclusion

Ce chapitre clarifie plus la méthodologie adoptée puis la conception qui consiste à déterminer les différentes fonctionnalités de mon application ainsi que rôles de chaque utilisateur.

Le chapitre suivant consacre à l’architecture et les technologies utilisées pour la réalisation de notre application.

Chapitre 4:

Technologies utilisées

**Introduction**

Après avoir achevé l’étape de conception de l’application, nous allons entamer dans ce chapitre la présentation de l’environnement de développement, en présentant les outils de développement et les différentes technologies utilisées.

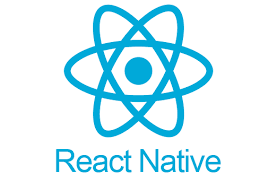
1. Environnement de travail
2. Les Frameworks et les technologies

 **Express.js**

**Express.js** est un framework web minimaliste et flexible pour **Node.js**, qui facilite la création d'applications web et d'API. Il fournit une série d'outils et de fonctionnalités pour développer des serveurs web en JavaScript de manière simple et efficace.

**Node.js :**

**Node.js** est une plateforme puissante et efficace pour développer des applications serveur en JavaScript, offrant des performances élevées, une grande scalabilité, et une riche bibliothèque de modules via NPM. Ses caractéristiques en font un choix populaire pour construire des applications modernes, rapides et évolutives.

**React Native**

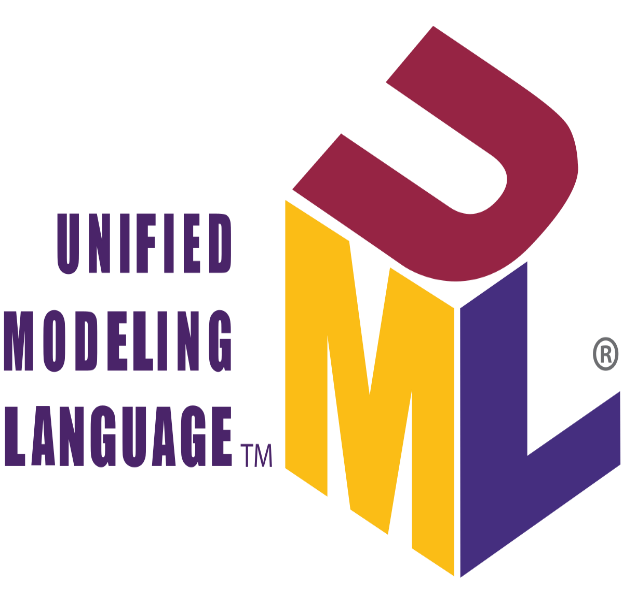
**React Native** est un puissant framework pour le développement d'applications mobiles en utilisant JavaScript et React. Il permet de créer des applications natives performantes pour iOS et Android avec un code partagé, offrant ainsi une solution efficace et rapide pour le développement multiplateforme.

 **MongoDB**

**MongoDB** est une base de données NoSQL flexible, performante et scalable, adaptée pour gérer des données variées et volumineuses. Sa capacité à s'adapter à des besoins changeants et à offrir une haute disponibilité en fait un choix populaire pour les applications modernes.

 **JavaScript**

**JavaScript** est un langage de programmation essentiel pour le développement web moderne, permettant de créer des interfaces utilisateur interactives et des applications serveur performantes. Sa flexibilité, sa facilité d'apprentissage, et son large écosystème en font un choix incontournable pour les développeurs.

**UML**

**UML** ou Unified Modeling Language, est un langage de modélisation visuelle utilisé dans le domaine du génie logiciel pour concevoir, représenter et spécifier des systèmes logiciels. Il offre une série de diagrammes standardisés qui permettent aux développeurs, aux architectes logiciels et aux autres parties prenantes de communiquer efficacement sur la structure, le comportement, les interactions et les exigences d'un système.

1. Les outils

** GitHub**

**GitHub** est bien plus qu'une simple plateforme de gestion de versions. C'est un écosystème complet qui révolutionne la manière dont les développeurs collaborent, partagent et construisent des logiciels. En tant que plateforme de développement de logiciels la plus utilisée au monde, GitHub offre une multitude de fonctionnalités puissantes.

 **Postman**

**Postman** est un outil populaire utilisé par les développeurs pour tester, déboguer et documenter les API. Il offre une interface utilisateur conviviale qui permet aux développeurs de créer, de partager et d'exécuter des requêtes HTTP, de visualiser les réponses et les données renvoyées par les serveurs, et de collaborer avec d'autres membres de l'équipe. Postman permet de gérer facilement les différentes méthodes HTTP telles que GET, POST, PUT et DELETE, ainsi que les en-têtes et les paramètres de requête.

ExpoGo

**ExpoGo** est une plateforme de développement d'applications mobiles qui simplifie le processus de création et de déploiement d'applications pour iOS et Android. Conçu pour les développeurs utilisant React Native, ExpoGo offre un ensemble d'outils et de services pour accélérer le développement d'applications mobiles. Avec ExpoGo, les développeurs peuvent créer des applications sans avoir besoin de configurer et de gérer les environnements de développement natifs pour iOS et Android.

**Entreprise architect**

**Enterprise Architect** est un logiciel de modélisation et de conception UML (Unified Modeling Language) largement utilisé par les architectes logiciels et les professionnels du développement pour concevoir et documenter des systèmes logiciels et des processus d'affaires. Il offre une gamme complète d'outils et de fonctionnalités pour créer des modèles visuels, des diagrammes et des documents qui capturent tous les aspects d'un système, depuis sa conception initiale jusqu'à sa mise en œuvre et sa maintenance.

**Visual studio code**

**Visual Studio Code**, souvent abrégé en VS Code, est un éditeur de code source léger, performant et extensible développé par Microsoft. Il est devenu l'un des éditeurs de code les plus populaires parmi les développeurs pour sa simplicité, sa polyvalence et sa richesse en fonctionnalités.

**Conclusion**

Ce chapitre illustre les outils de travail utilisés et les différentes normes de développement réalisées d'où les normes techniques, de nommage et d’organisation nécessaire au développement de l’application.

Le dernier chapitre consiste la phase d’implémentation qui contient les interfaces de l’application et sa fonctionnalité.