

Rapport PFA Génie Informatique

Abir Lakhal

Juin 2021

Dédicace

Je dédie ce travail à tous ceux qui me sont chers :

A ma mère qui m'a arrosée de tendresse et des espoirs, son amour a fait de moi ce que je suis aujourd'hui.

A la mémoire de mon père qui m'a toujours soutenue et motivée, il est toujours dans mon cœur, paix à son âme.

A toute ma famille et mes amis.

Remerciements

C'est avec un grand plaisir que je réserve ces lignes en signe de profonde reconnaissance et de gratitude à Madame Feyrouz Hamdaoui pour l'encadrement et les recommandations précieuses, sa disponibilité et ses encouragements tout au long de cette période, je la remercie vivement pour m'avoir fait confiance.

Je tiens à remercier toute l'équipe de ItGate pour leur collaboration, leurs explications ainsi que pour la documentation mise à ma disposition.

J'adresse également mes remerciements à tous les enseignants du Département Informatique pour leur contribution à ma formation d'ingénieur.

Table des matières

Dédicace	i
Remerciements	ii
Introduction générale	1
1 Etude préalable	2
1.1 Introduction	2
1.2 Contexte du stage	2
1.2.1 Cadre du projet	2
1.2.2 Présentation de l'établissement	3
1.2.3 Structure sociale de ItGate	3
1.2.4 Les services de ItGate	4
1.3 Présentation du sujet :	5
1.3.1 Problématique et motivations :	5
1.4 Expression des besoins	6
1.4.1 Etude des applications existantes	6
1.4.2 Solution proposée	7
1.5 Objectifs :	7
1.6 Spécification des besoins	7
1.6.1 Besoins fonctionnels :	7
1.6.2 Besoins non fonctionnels	8
1.7 Modèle de cycle de vie	9
1.8 Diagramme de Gantt	12

1.9	Conclusion	13
2	Etude conceptuelle	14
2.1	Introduction	14
2.2	Diagramme de cas d'utilisation	15
2.2.1	Description textuelle	16
2.3	Diagramme de classes	19
2.3.1	Dictionnaire des données	21
2.4	Diagramme de séquence	23
2.5	Architecture système	25
2.5.1	Module de développement	25
2.5.2	MVC-MVVM	26
2.5.3	Vue d'ensemble de l'application	26
2.6	Conclusion	28
3	Réalisation	29
3.1	Introduction	29
3.2	Environnement de développement	29
3.2.1	Environnement technique	29
3.2.2	Environnement logiciel	30
3.3	Interfaces de l'application	32
3.3.1	Interface Principale	33
3.3.2	Interface Inscription	33
3.3.3	Interface Authentification	34
3.3.4	Interface A propos	34
3.3.5	Interface Contact	35
3.3.6	Interface Liste des événements	35
3.3.7	Interface Details d'événement	36
3.3.8	Interface Création d'événement	36
3.3.9	Interface Paiement	37
3.3.10	Interface billet	37
3.3.11	Interface Profile	38

3.3.12 Interface dashboard administrateur	38
3.4 Conclusion	39
Conclusion et perspectives	40
Bibliographie	41

Table des figures

1.1	logo de ItGate group	3
1.2	Cycle de vie UP	11
1.3	Planning du projet	12
2.1	Diagramme de cas d'utilisation global	15
2.2	Diagramme de classes	20
2.3	Diagramme de séquence "Authentification"	23
2.4	Diagramme de séquence "Créer événement"	24
2.5	Diagramme de séquence " Payer e_ticket"	25
2.6	Visual Studio Code	26
2.7	Vue d'ensemble de l'application	27
3.1	Visual Studio Code	31
3.2	Postman	31
3.3	Postman	32
3.4	mongodb compass	32
3.5	Interface Principale	33
3.6	Interface Inscription	33
3.7	Interface Authentification	34
3.8	Interface A propos	34
3.9	Interface Contact	35
3.10	Liste des événements	35
3.11	Details d'événement	36
3.12	Interface Création d'événement	36

3.13 Interface Paiement	37
3.14 Interface billet	37
3.15 Interface Profile	38
3.16 dashboard administrateur	39

Liste des tableaux

1.1	Tableau de comparaison entre les applications existants	6
1.2	Etude comparative entre les méthodologies de gestion de projet	10
2.1	Description cas d'utilisation "Gérer compte utilisateur"	16
2.2	Description cas d'utilisation "Gérer compte utilisateur"	16
2.3	Description cas d'utilisation "Consulter messagerie"	16
2.4	Description cas d'utilisation "Gérer événement"	17
2.5	Description cas d'utilisation "Suivre achat"	17
2.6	Description cas d'utilisation "Gérer compte"	17
2.7	Description cas d'utilisation "Envoyer message anonyme"	18
2.8	Description cas d'utilisation "s'authentifier"	18
2.9	Description cas d'utilisation "Consulter événement"	18
2.10	Description cas d'utilisation "Payer e-ticket"	19
2.11	Description cas d'utilisation "Télécharger e-ticket"	19
2.12	Description cas d'utilisation "Consulter historique"	19
2.13	Table Client	21
2.14	Table Administrateur	21
2.15	Table Organisateur	21
2.16	Table Catégorie	22
2.17	Table message	22
2.18	Table Evénement	22
2.19	Table Billet	22
2.20	Table Paiement	23

Introduction générale

La réservation et l'achat de tickets d'entrée pour un événement (une conférence, un festival, un match ou de l'achat d'un bon de participation à une formation. . .), ne sont pas toujours avantageux en direct, ce qui nous motive à donner plus d'importance à la vente électronique.

Les billetteries numériques sont des outils rapides, faciles à utiliser et efficaces qui remplacent les billets au format papier. Ils sont utilisées par les compagnies aériennes ainsi que dans les transports en commun. Ils sont également employés pour l'achat ou la réservation d'une place de spectacle comme les tickets de cinéma. Bénéficiant de ses atouts, beaucoup d'organisateurs évènements s'orientent encore plus vers l'utilisation de la billetterie électronique notamment avec les conditions sanitaires nationales et internationales favorisant la transformation digitale de tous les services.

Notre travail s'inscrit dans le cadre du projet de fin de 4ème année en génie informatique (spécialité Génie Logiciel) ayant comme objectif "La conception et la réalisation d'une application web de billetterie électronique" pour le compte de ItGate, une société d'ingénierie informatique, spécialisée en développement des applications web, mobiles, la création des sites, la conception graphique et la gestion de communauté.

Ce rapport se compose de trois chapitres.

Nous débuterons le premier chapitre par la présentation générale de la société d'accueil, la problématique, la solution proposée ainsi que la démarche de travail. Par la suite, nous allons introduire la partie de l'étude conceptuelle de l'application. Enfin, nous décrirons l'architecture et la phase de la réalisation du projet.

Chapitre 1

Etude préalable

1.1 Introduction

Le billet électronique (aussi appelé e-ticket) est un billet dématérialisé qui remplace un support d'information matériel par des informations numériques.[1]

Dans ce chapitre, nous allons décrire et définir en premier lieu le cadre de ce projet et en deuxième lieu nous allons présenter l'organisme qui a accueilli ce stage de fin d'année. Dans une troisième étape, nous présenterons les motivations, la problématique, l'analyse de l'existant ainsi que la solution proposée qui a permis à ce stage d'aboutir à ses fins.

1.2 Contexte du stage

1.2.1 Cadre du projet

Le projet entre dans le cadre universitaire académique au sein de l'Ecole Pluridisciplinaire Internationale de Sousse (EPI) pour valider le projet de fin de 4 ème année Génie Informatique spécialité Génie logiciel. Ce projet a été effectué au sein de la société ItGate durant la période du 3 mois.

1.2.2 Présentation de l'établissement

ItGate Group est une société créée en 2015 et spécialisée en développement et services informatiques : l'expérience utilisateur, la conception, le développement et le Webdesign, Elle déclare avoir les valeurs comme : organisation, collaboration, confiance et responsabilité.



FIGURE 1.1 – logo de ItGate group

1.2.3 Structure sociale de ItGate

Le travail d'équipe et l'adaptabilité structurelle basé sur la logique adhocratique sont les facteurs de succès d'ItGate.

En fait, elle repose sur une structure polycellulaire qui cherche à réduire les niveaux hiérarchiques dans le but d'assurer une progression horizontale des carrières, de stimuler la créativité individuelle et permettre une flexibilité d'organisation.

ItGate dispose de trois unités, chacune étant responsable d'un département, comprenant des personnels compétents et autonomes qui interagissent de manière dynamique avec les autres unités.

Au cours de chaque projet, ItGate met en interaction des membres de différentes unités pour assurer la bonne gestion et génération du projet.

Les départements existants dans ItGate sont :

- Département de développement :

Ce département est constitué des personnes compétentes et qualifiées dans des technologies de développements variées délivrant des projets de qualité pour des clients nationaux et internationaux. Ce département se divise en sous départements web et mobile où les développeurs sont amenés à produire des applications Android et IOS.

- Département de designer graphique :

Constitue le noyau d'élaboration des maquettes et des chartes graphiques innovantes et attractives.

- Département de test :

Celui-ci s'intéresse au projet en cours et après la réalisation, afin de tester ses comportements.

1.2.4 Les services de ItGate

Dans le cadre de ses activités de son administration quotidienne, ItGate s'occupe notamment de :

- mettre en place des solutions informatiques fiable, innovante et moderne ;
- développer des applications web modernes ;
- développer des applications mobiles pour les plateformes Android et IOS ;
- développer des supports et des interfaces graphiques pour ses clients ;
- promouvoir et de mettre en œuvre les stratégies de référencement pour une meilleure visibilité sur Internet ;

ItGate est divisée en deux organismes : centre de formation professionnelle, spécialisé dans plusieurs secteurs (informatique, marketing, management, etc.), et boîte de développement Web, Mobile et Embarqué où j'ai effectué mon projet de fin de 4ème année.

1.3 Présentation du sujet :

Le sujet de notre projet de fin d'année consiste à développer une application web de billetterie électronique, qui facilite et accélère le marketing en ligne des événements.

D'une part, Les organisateurs présentent leurs évènements, les clients visualisent et achètent les billets de l'autre.

1.3.1 Problématique et motivations :

Vu les circonstances sanitaires à l'échelle national et international due au pandémie de Coronas virus et avec toutes les mesures prises notamment de distanciation sociale, notre sujet trouve tout son intérêt face à un risque élevé qui est lui d'être achevé du jour au lendemain.

Vu le grand nombre des évènements organisés chaque jour, la promotion, la visibilité et le choix deviennent de plus en plus complexes ; de plus, on se retrouve face à d'énormes difficultés pour le client aussi que pour l'organisateur. En effet :

- La nécessité d'une recherche quotidienne pour pouvoir visualiser et choisir l'évènement ;
- C'est indispensable de se déplacer, faire la queue et gaspiller du temps, tout en ayant le risque que les tickets soient épuisées ;
- Risque de perdre les billets ou les oublier le jour de l'évènement ;
- Les couts d'impression des tickets en papier qui augmentent de temps en temps ;
- La désignation d'agents de guichets de confiance et la recherche des points de ventes accessibles par un maximum de public cible ;

Face à ces problèmes, la réalisation d'une application de billetterie électronique est une solution adaptée à la gestion de tous les types d'évènements. On trouvera les outils nécessaires pour promouvoir les événements, mettre en place une billetterie en ligne, suivre les ventes en un coup d'œil.

1.4 Expression des besoins

L'expression des besoins est le processus permettant de définir les besoins et les attentes dans notre projet. Pour ce faire, il est indispensable d'adopter une démarche basée sur l'étude de l'existant et une interprétation comparative qui permet de positionner notre projet par rapport aux applications similaires.

1.4.1 Etude des applications existantes

Le tableau ci-dessous illustre le résultat de l'étude comparative que nous avons menés afin de détecter et d'évaluer les applications dites similaires à la notre.

Application web	Avantages	Inconvénients
Teskerti (Tunisienne) 	-L'application est facile à utiliser. -L'application est liée à Google Agenda. -Possibilité d'appliquer un code promo.	-Seul l'administrateur peut créer l'événement. -La carte géographique n'est pas bien précise.
Tikashop (Tunisienne) 	-Une alerte pour informer des prochains événements. -Choisir une date pour se présenter aux guichets de vente spécifiés.	-Problème de chargement de Google Maps.
Billetweb (Française) 	-L'organisateur peut créer et gérer son propre événement. -Récupération des billets perdus. -Existence d'un service technique (chat).	-Pas de possibilité de laisser un feedback pour les organisateurs.

TABLE 1.1 – Tableau de comparaison entre les applications existants

1.4.2 Solution proposée

Malgré qu'elles répondent à certaines attentes, l'étude du marché des applications existantes nous a permis de dégager plusieurs limites. Pour se positionnant par rapport à ces derniers, notre solution propose de développer une application web baptisée "RESERVI" qui permet non seulement de faciliter les processus de vente et d'achat des billets électroniques mais qui représente également un outil aux organisateurs pour promouvoir leurs évènements.

1.5 Objectifs :

Dans un objectif de digitaliser le processus de billetterie, notre application vise principalement à offrir à ses utilisateurs des services de vente/achat des e-tickets en ligne.

Ce projet est composé de deux parties :

- Partie administrateur :

Cette partie permettra à l'administrateur de suivre les données relatifs aux évènements et aux réservations à travers un Dashboard.

- Partie utilisateur :

Cette partie est dédiée à la fois aux organisateurs à gérer leurs évènements et aux clients pour visualiser et acheter les e-tickets.

1.6 Spécification des besoins

Les besoins d'utilisation de notre application sont regroupés en besoins fonctionnels et en besoins non fonctionnels.

1.6.1 Besoins fonctionnels :

Notre application doit permettre à l'utilisateur de :

- Aider à faire la promotion de l'évènement :

Notre application doit disposer d'une affiche virtuelle à travers laquelle le client peut consulter une grande variété des événements, il sera donc indispensable d'y présenter les prix et les descriptions de chaque événement pour faciliter la sélection et l'achat.

- S'inscrire :

Jusqu'à ce stade, l'utilisateur est toujours anonyme mais pour pouvoir bénéficier de toutes les fonctionnalités, il faut qu'il s'inscrive.

- Créer des évènements :

L'organisateur crée son évènement dans l'application en mettant toutes les informations qui y sont relatives. Une validation est indispensable pour que l'événement soit ajouté à la liste des évènements.

- Réserver et acheter en ligne :

C'est une phase très sensible, pour cela qu'elle exige à haut niveau de sécurité, pour terminer la procédure de paiement avec succès, le client doit remplir le formulaire et indiquer le numéro de sa carte et sa valeur de vérification dite CVV.

- Recevoir le ticket :

La page finale représente le billet électronique accompagné des détails à savoir : (nom de l'événement, la date, le temps...) ainsi que QR Code. Le client aura la permission de télécharger le billet.

1.6.2 Besoins non fonctionnels

Notre application est soumise à un ensemble de contraintes techniques et ergonomiques à respecter pour la réalisation et le bon fonctionnement.

Les principaux besoins non fonctionnels sont :

- Fiabilité : L'application doit toujours s'exécuter sans erreur et doit être satisfaisante ;

- Gestion des erreurs : L'ambiguïté doit être indiquée par des messages d'erreur bien claires pour guider correctement l'utilisateur et le familiariser avec notre site Web ;
- Ergonomie : En terme de navigation entre les différentes pages, les couleurs utilisées et les mises en page doivent être adaptée à l'expérience de l'utilisateur, sans effort (claire et simple à utiliser) ;
- Sécurité : Notre solution doit d'abord respecter la confidentialité des données personnelles de l'utilisateur, qui reste l'une des restrictions les plus importantes de l'application ;
- Aptitude à la maintenance et à la réutilisation : Le système doit être conforme aux normes et à une architecture bien structurée afin de pouvoir être entretenu et réutilisé ;

1.7 Modèle de cycle de vie

Un modèle de cycle de vie décrit la séquence des phases d'un projet.

Pour assurer un bon rendement de développement en termes de qualité et de productivité, on doit choisir selon notre projet la méthodologie à adopter.

Le tableau ci-dessous (figure 1.2) montre l'implémentations de plusieurs méthodologies de développement de logiciels qui peut être adopté au genre de notre travail à savoir le modelé en cascade, le modelé en V, le modelé UP, et le modelé 2TUP.

Méthodes	Les principes fondamentaux
En Cascade [2]	Un modèle de gestion linéaire qui divise les processus de développement en phases de projet successives. Chaque phase est effectuée une seule fois. Les sorties de chaque phase antérieure sont intégrées comme entrées de la phase suivante.
En V [3]	Un modèle utilisé dans différents processus de développement, la lettre V fait référence à la vision schématique de ce cycle qui prend la forme d'un V : une phase descendante suivie d'une phase ascendante. Le cycle en V associe à chaque phase de réalisation une phase de validation.
UP [4]	Une méthode de développement pour les logiciels orientés objets. UP est piloté par les cas d'utilisation et centré sur l'architecture logicielle. Cette méthode découpe le projet en mini-projets : des itérations qui donnent lieu à des incrémentums.
2UP [5]	Un processus unifié qui 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.

TABLE 1.2 – Etude comparative entre les méthodologies de gestion de projet

Notre travail est basé sur un processus de développement bien défini qui décrit chaque partie du projet de façon indépendante. C'est pour cela qu'on est besoin d'un cycle de développement itératif et incrémental qui découpe le projet en mini-projets tout en commençant par une étude préliminaire.

Notre choix s'est alors porté vers la méthode UP vu qu'elle est caractérisée par une approche adéquate à notre projet et qu'elle respecte notre plan de travail.

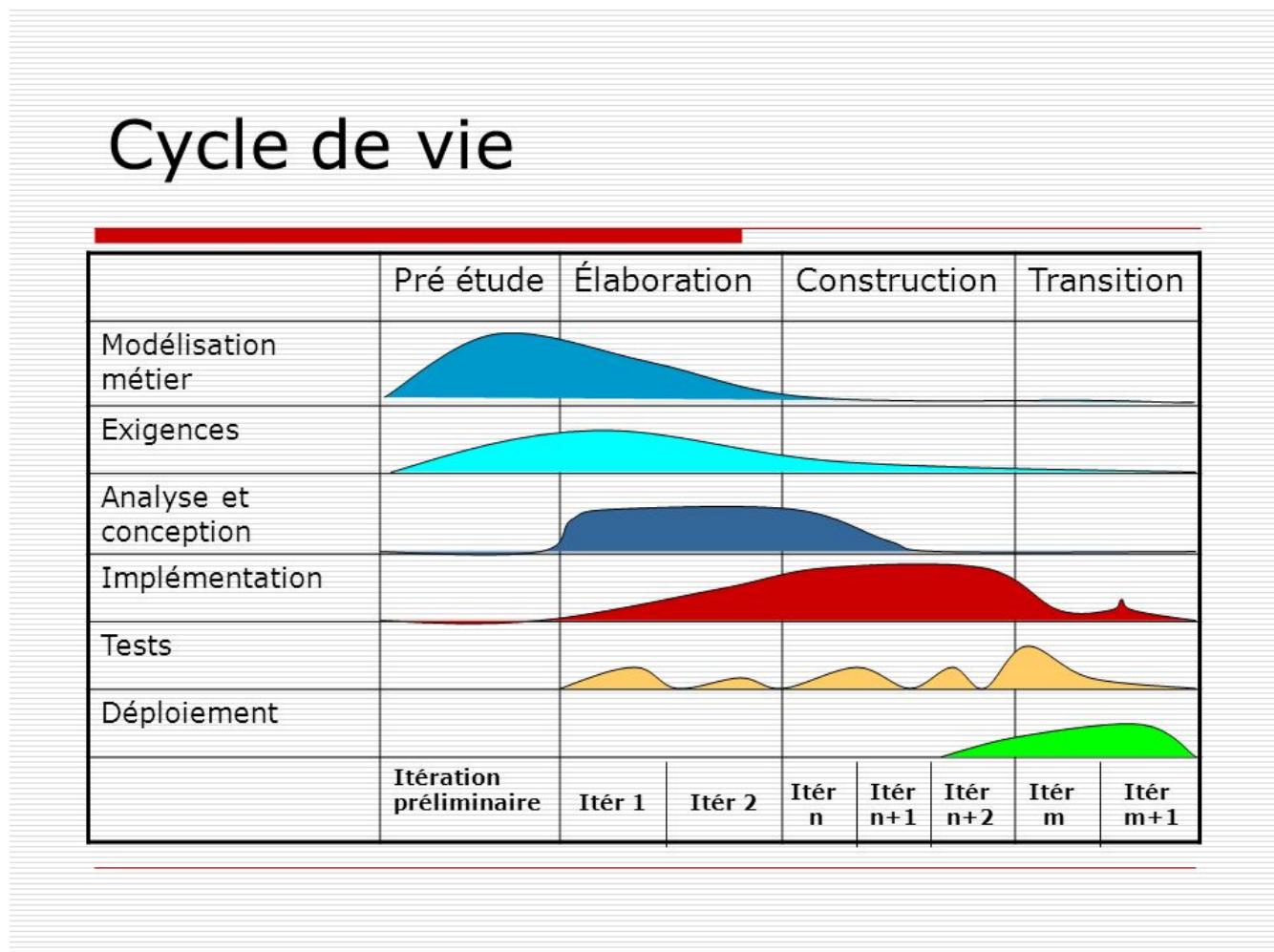


FIGURE 1.2 – Cycle de vie UP

1.8 Diagramme de Gantt

Ayant une réflexion sur la planification structurelle et opérationnelle de notre projet, nous avons pu faire une estimation de travail à effectuer.

Le diagramme de Gantt est une représentation visuelle de l'échéancier d'un projet. Les dates de l'échéancier sont représentées en haut du diagramme, et les différentes tâches et activités sont listées sur la gauche.[6]

Le diagramme suivant (figure 1.3) définit la période de chaque tâche principale pour réaliser notre projet.

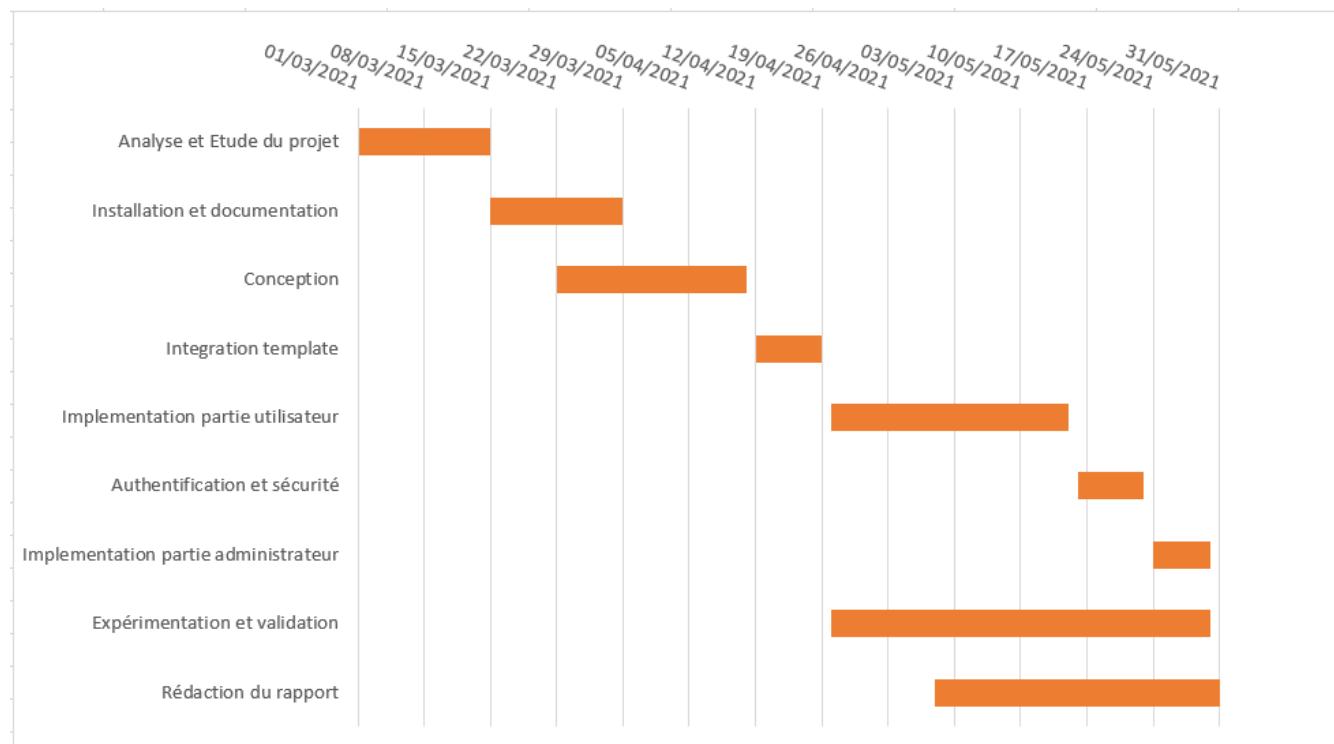


FIGURE 1.3 – Planning du projet

1.9 Conclusion

Dans ce premier chapitre nous avons mis le sujet dans son cadre général et nous avons précisé nos besoins après avoir effectué une étude analytique et comparative entre des exemples d'applications de billetterie en ligne en déduisant leurs avantages et leurs défaillances que nous allons mettre en considération dans notre projet. Dans le deuxième chapitre, nous allons aborder l'étude conceptuelle de notre application, mettant en œuvre les tâches qui nous ont été confié au sein de cette société durant la période de stage.

Chapitre 2

Etude conceptuelle

2.1 Introduction

L'étude conceptuelle est la tâche la plus importante du processus de développement d'un système d'informatique.

Dans ce chapitre, nous allons identifier les cas d'utilisation et tracer le diagramme de classes et les diagrammes de séquence. Par la suite, nous présenterons l'architecture de notre application.

Pour modéliser les fonctionnalités et vulgariser les aspects liés à la conception et à l'architecture de ce système, nous avons adopté une approche orientée objet basée sur un outil de modélisation UML (Unified Modeling Language).

UML est un langage de modélisation constitué de diagrammes intégrés utilisés par les développeurs pour la représentation visuelle des objets, des états et des processus dans un logiciel ou un système.[7]

UML n'a cessé d'évoluer, il est préférable puisqu'il permet de :

- * obtenir une modélisation de très haut niveau indépendante des langages et des environnements ;
- * faire des simulations avant de construire un système ;
- * exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, etc... ;

2.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est généralement la première étape UML pour la conception d'une application conforme aux attentes des utilisateurs.

Ses principaux rôles sont de recueillir, d'analyser et d'organiser les besoins, ainsi que de recenser les grandes fonctionnalités d'un système.

Le diagramme suivant représente le cas d'utilisation général de la billetterie électronique.

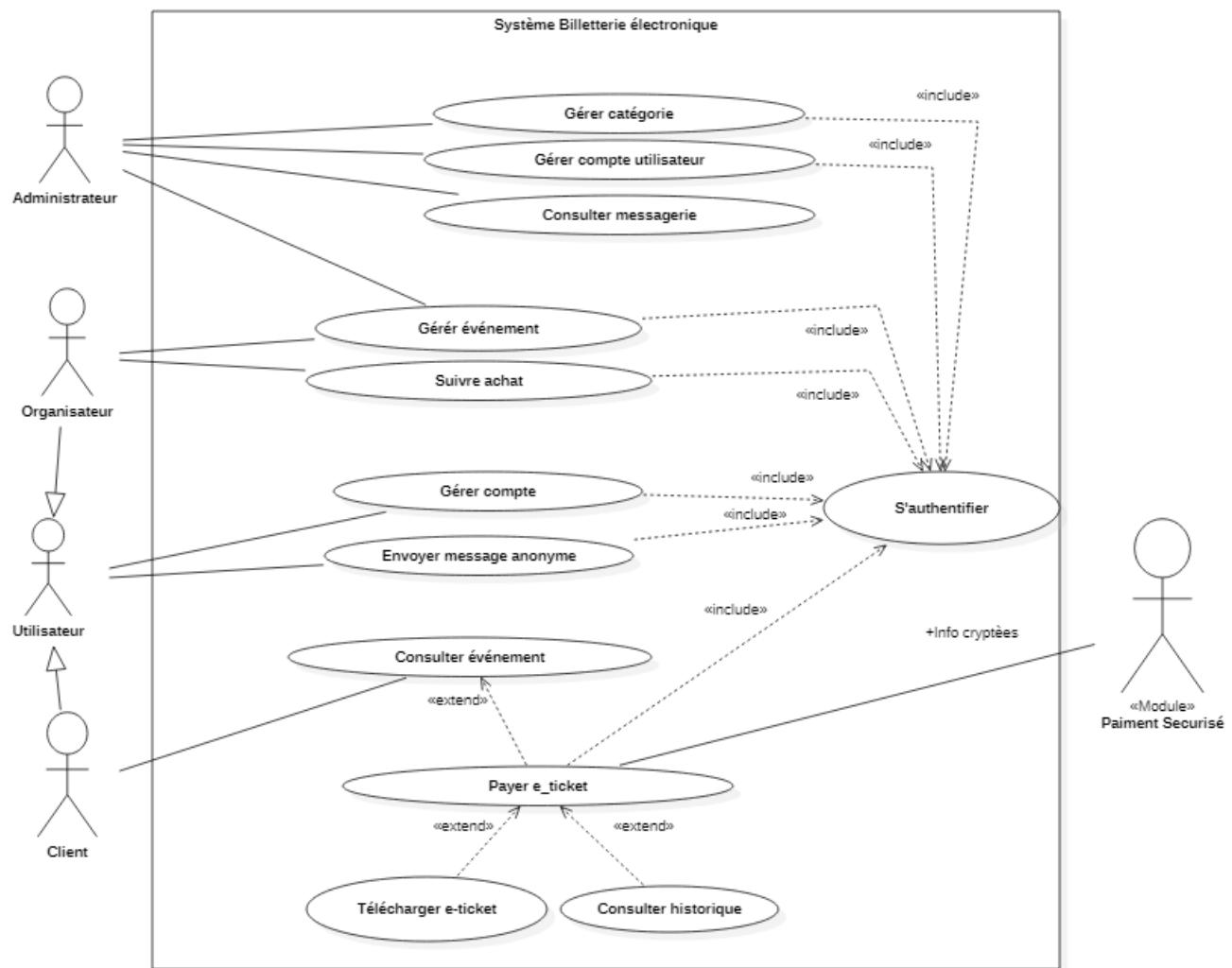


FIGURE 2.1 – Diagramme de cas d'utilisation global

2.2.1 Description textuelle

Sommarie Gérer catégorie	
Cas d'utilisation	Gérer catégorie
Acteur	Admin
Objectif	Modifier, ajouter ou supprimer catégorie.
Précondition	S'authentifier
Postcondition	
Scénario	Appuyer sur le volet catégorie.

TABLE 2.1 – Description cas d'utilisation "Gérer compte utilisateur"

Sommarie Gérer compte utilisateur	
Cas d'utilisation	Gérer compte utilisateur
Acteur	Admin
Objectif	Modifier le compte en cas d'un problème.
Précondition	S'authentifier
Postcondition	
Scénario	Consulter le compte et le modifier.

TABLE 2.2 – Description cas d'utilisation "Gérer compte utilisateur"

Sommarie Consulter messagerie	
Cas d'utilisation	Consulter messagerie
Acteur	Admin
Objectif	Consulter la messagerie pour savoir les avis des utilisateurs de façon anonyme.
Précondition	S'authentifier
Postcondition	
Scénario	Cliquer sur messagerie et consulter les messages anonymes.

TABLE 2.3 – Description cas d'utilisation "Consulter messagerie"

Sommaire Gérer événement	
Cas d'utilisation	Gérer événement
Acteur	Admin
Objectif	Modifier l'événement.
Précondition	S'authentifier
Postcondition	
Scénario	Consulter l'événement et le modifier.

TABLE 2.4 – Description cas d'utilisation "Gérer événement"

Sommaire Suivre achat	
Cas d'utilisation	Suivre achat
Acteur	Organisateur
Objectif	Suivre les ventes de l'événement.
Précondition	S'authentifier
Postcondition	
Scénario	Accéder aux détails de l'événement.

TABLE 2.5 – Description cas d'utilisation "Suivre achat"

Sommaire Gérer compte	
Cas d'utilisation	Gérer compte
Acteur	Client et Organisateur
Objectif	Modifier les informations de l'utilisateurs.
Précondition	S'authentifier
Postcondition	
Scénario	Cliquer sur "My profile" et modifier les informations.

TABLE 2.6 – Description cas d'utilisation "Gérer compte"

Sommarie Envoyer message anonyme	
Cas d'utilisation	Envoyer message anonyme
Acteur	Client et Organisateur
Objectif	Envoyer un message anonyme donner avis ou réclamer.
Précondition	
Postcondition	
Scénario	Cliquer sur "Contact Us".

TABLE 2.7 – Description cas d'utilisation "Envoyer message anonyme"

Sommarie s'authentifier	
Cas d'utilisation	s'authentifier
Acteur	Client, Organisateur et Admin
Objectif	S'authentifier pour accéder au profil.
Précondition	
Postcondition	
Scénario	Taper sur "Login" puis entrer l'adresse mail et le mot de passe.

TABLE 2.8 – Description cas d'utilisation "s'authentifier"

Sommarie Consulter événement	
Cas d'utilisation	Consulter événement
Acteur	Client
Objectif	Consulter événement pour voir les nouveautés et reserver
Précondition	
Postcondition	
Scénario	Consulter les nouveaux événements ou les choisir par catalogue.

TABLE 2.9 – Description cas d'utilisation "Consulter événement"

Sommarie Payer e-ticket	
Cas d'utilisation	Payer e-ticket
Acteur	Client
Objectif	Payer le ticket pour avoir le billet avec le QRCode
Précondition	S'authentifier
Postcondition	
Scénario	Appuyer sur "Buy" et remplir les informations puis en cliquant sur "Continue to check" le billet sera prêt à télécharger.

TABLE 2.10 – Description cas d'utilisation "Payer e-ticket"

Télécharger e-ticket	
Cas d'utilisation	Télécharger e-ticket
Acteur	Client
Objectif	Enregistrer le e-ticket.
Précondition	Payer e-ticket
Postcondition	
Scénario	Cliquer sur "Download" et le billet va être enregistré.

TABLE 2.11 – Description cas d'utilisation "Télécharger e-ticket"

Consulter historique	
Cas d'utilisation	Consulter historique
Acteur	Client
Objectif	Consulter et voir ses achats.
Précondition	S'authentifier
Postcondition	
Scénario	Cliquer sur "My history" et la liste des achats s'affiche.

TABLE 2.12 – Description cas d'utilisation "Consulter historique"

2.3 Diagramme de classes

Le diagramme de classes offre une représentation graphique de la structure statique d'un système décrivant les structures des classes de notre application ainsi que leurs liens.

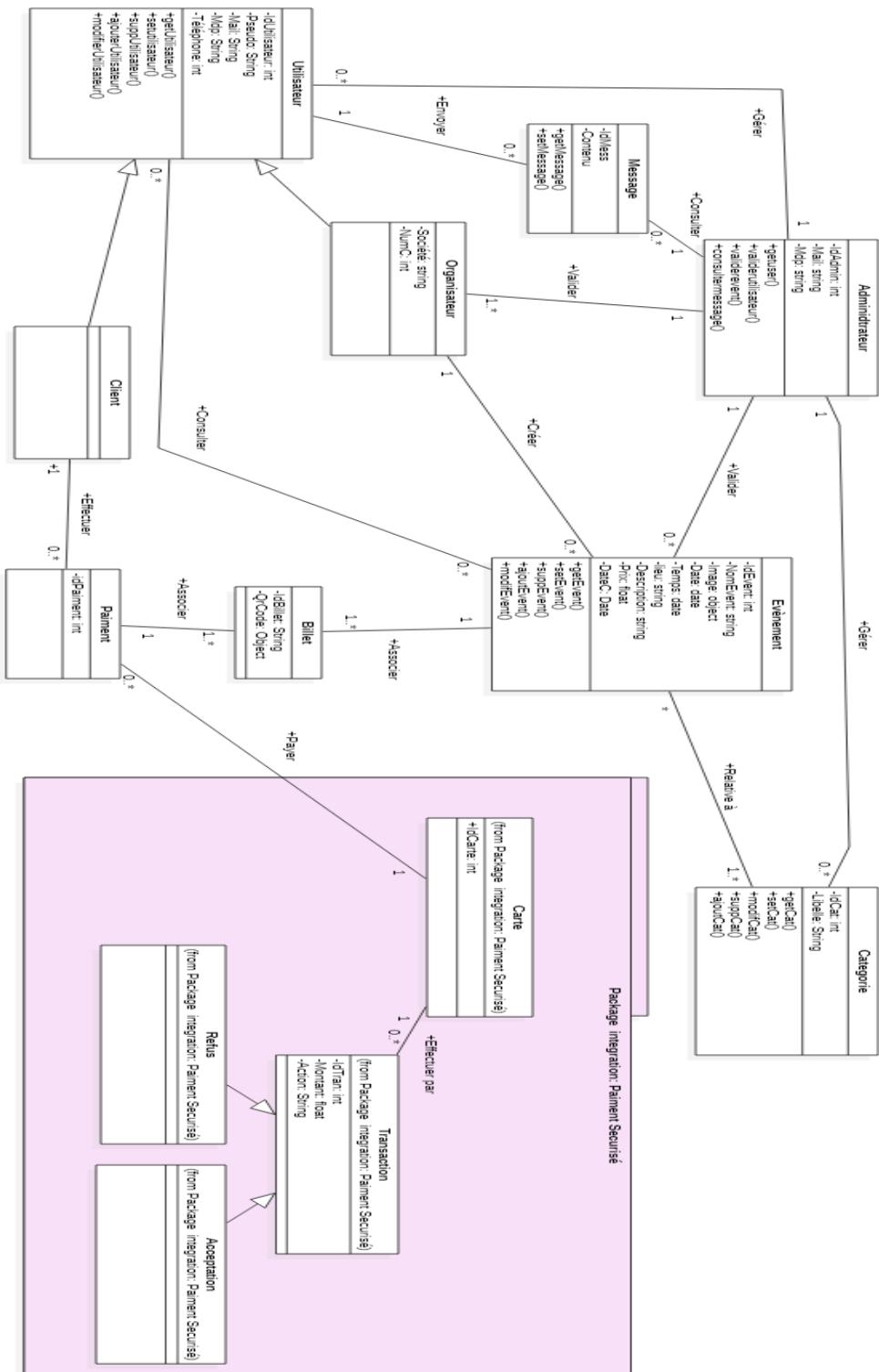


FIGURE 2.2 – Diagramme de classes

2.3.1 Dictionnaire des données

Attributs	Désignation	Type
IdClient	Identifiant de client	String
Pseudo	Pseudo de client	String
Mail	L'adresse mail de client	String
Mdp	Mot de passe de client	String
Téléphone	Numéro de téléphone de client	String

TABLE 2.13 – Table Client

Attributs	Désignation	Type
IdAdmin	Identifiant de l'admin	String
Mail	Mail de l'admin	String
Mdp	Mot de passe de l'admin	String

TABLE 2.14 – Table Administrateur

Attributs	Désignation	Type
IdOrganisateur	Identifiant de l'organisateur	String
Pseudo	Pseudo de l'organisateur	String
Mail	Mail de l'organisateur	Date
Mdp	Mot de passe de l'organisateur	String
téléphone	Numéro de téléphone de l'organisateur	String
Institue	Institue de l'organisateur	String
NumC	Numéro de compte de l'organisateur	String
Statue	Compte validé ou non	Booléen

TABLE 2.15 – Table Organisateur

Attributs	Désignation	Type
IdCat	Identifiant de catégorie	String
Libellé	Libellé de categorie	String

TABLE 2.16 – Table Catégorie

Attributs	Désignation	Type
IdMessage	Identifiant du message	String
Contenu	Contenu du message	String

TABLE 2.17 – Table message

Attributs	Désignation	Type
IdEvent	Identifiant de l'événement	String
Titre	Titre de l'événement	String
Image	Image de l'événement	Objet
Date	Date de l'événement	Date
Temps	Temps de l'événement	String
Lieu	Lieu de l'événement	String
Description	Description de l'événement	String
Prix	Prix de l'événement	Float
Capacité	nombre de place	Entier

TABLE 2.18 – Table Evénement

Attributs	Désignation	Type
IdBillet	Identifiant de billet	String
QrCode	Le QrCode de billet	Object

TABLE 2.19 – Table Billet

Attributs	Désignation	Type
IdPaiement	Identifiant de paiement	String

TABLE 2.20 – Table Paiement

2.4 Diagramme de séquence

Les diagrammes de séquences permettent de décrire les interactions entre les acteurs et le système selon un ordre chronologique.[8] Ils apportent une vue dynamique au système à et à toutes les interactions entre les différents composants.

Les diagrammes de séquence suivants représentent quelques cas d'utilisation de notre application.

2.4.1 Diagramme de séquence "Authentification"

Le diagramme de séquence présenté dans la figure 2.3 montre l'authentification de l'utilisateur par étapes.

L'utilisateur (client, organisateur) et l'administrateur saisies l'email et le mot de passe. Le contrôleur d'authentification va donc vérifier les données saisis : si les données sont conformes avec ceux qui sont stockées dans la base de données alors le contrôleur va demander de créer un jeton de service JWT puis l'interface principale s'affiche, sinon un message d'erreur va être affiché.

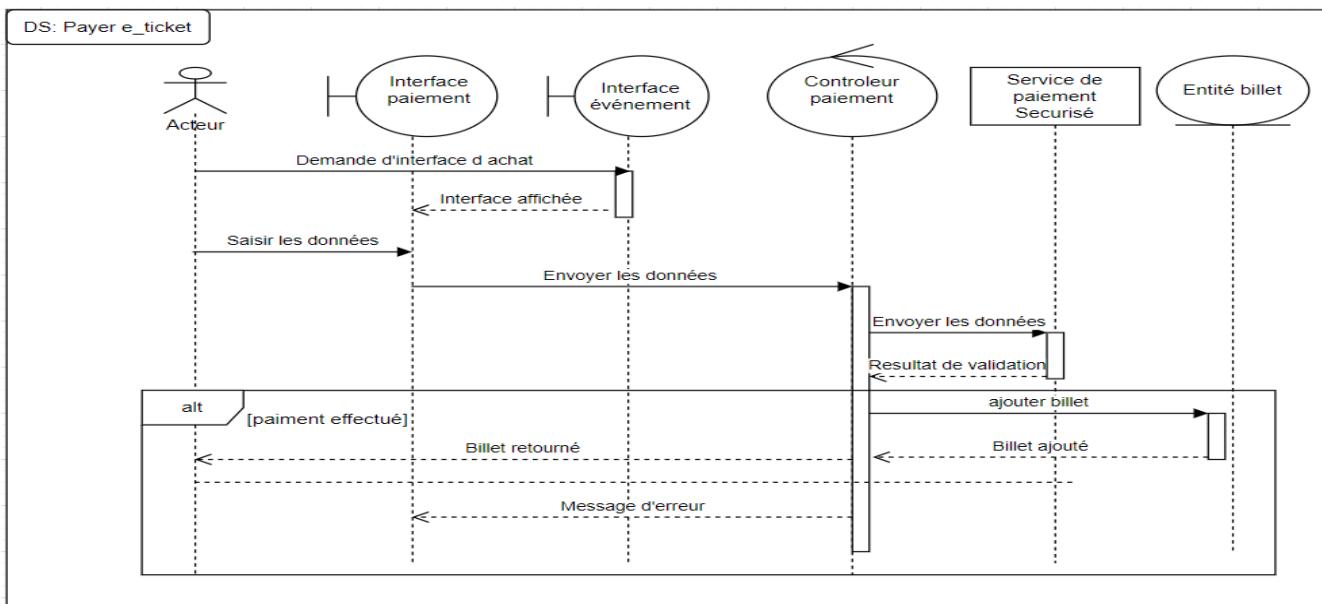


FIGURE 2.3 – Diagramme de séquence "Authentification"

2.4.2 Diagramme de séquence "Créer événement"

Ce diagramme de séquence (figure 2.4) montre les processus de création d'un événement. L'organisateur doit tout d'abord s'authentifier, après il demande la page de la création d'événement. Un formulaire s'affiche, et les données saisies par l'organisateur vont être ajoutées à la base de données.

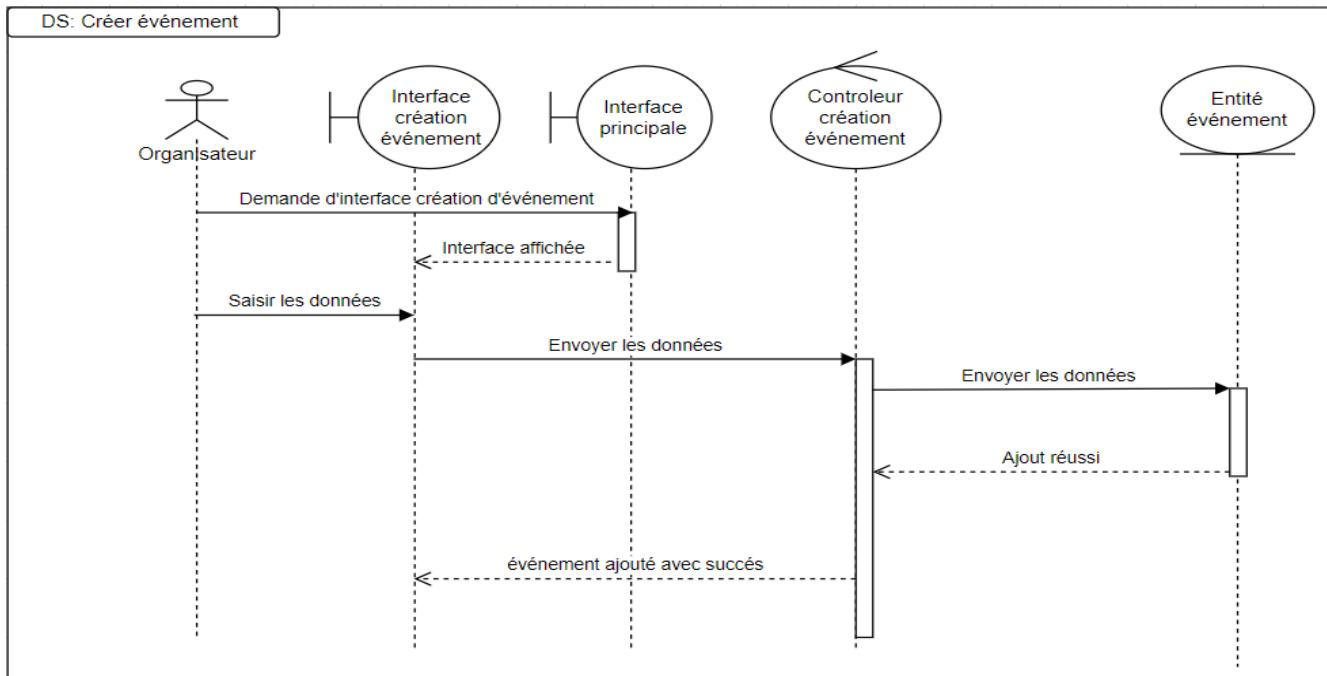


FIGURE 2.4 – Diagramme de séquence "Créer événement"

2.4.3 Diagramme de séquence "Payer e_ticket"

Le diagramme de séquence ci-dessous simule l'action de paiement de billet, après le choix de l'événement, le client clique sur "buy" et il va être dirigé vers la page de paiement, il saisit les informations qui seront validées par le service de paiement. Si les données sont validées alors le contrôleur affiche le billet.

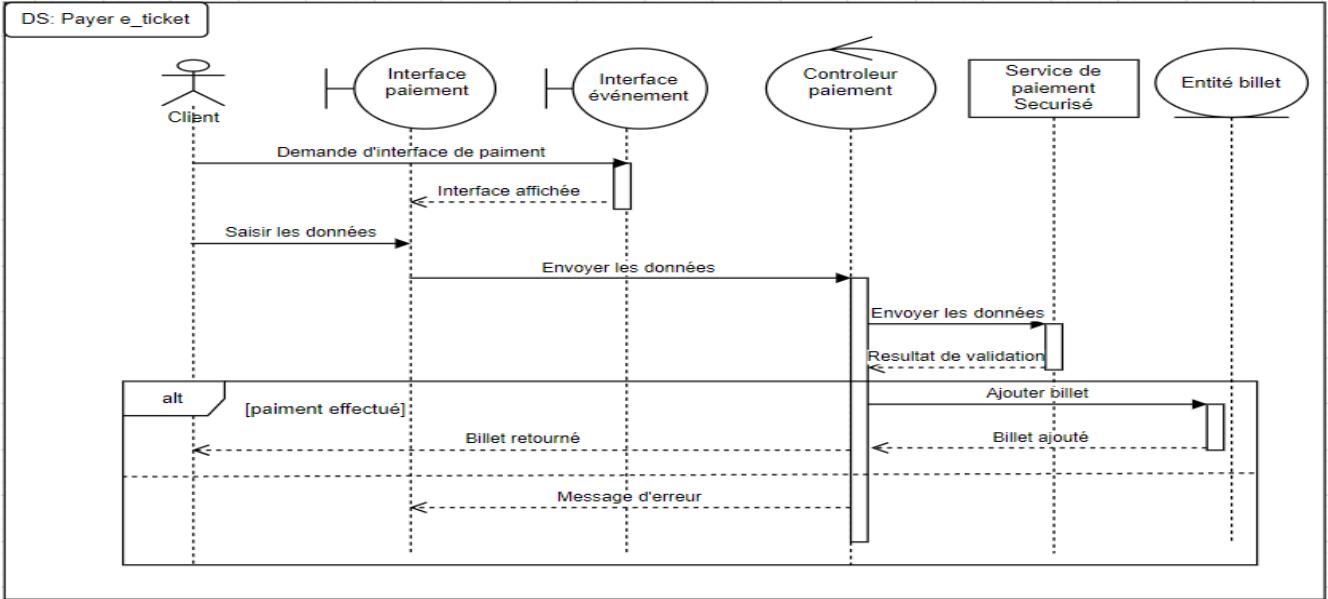


FIGURE 2.5 – Diagramme de séquence " Payer e_ticket"

2.5 Architecture système

Dans notre travail, toutes les données sont centralisées sur un seul serveur qui joue le rôle de l'administrateur or les utilisateurs utilisant simplement un client léger. Pour cela, il est bien clair que notre projet est basé sur l'architecture client-serveur.

Un client : est d'abord actif, il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur.

Un serveur : est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse. [9]

2.5.1 Module de développement

Mean Js : est un acronyme annonçant une pile de différentes technologies, utilisé pour créer des applications Web.

Chaque technologie a sa fonction propre, qui correspond à une couche de l'architecture MEAN. On reconnaîtra ainsi Mongodbs – ExpressJs – AngularJs – NodeJs. [10]



FIGURE 2.6 – Visual Studio Code

Dans la technologie MEAN, nous n'utilisons qu'une seule langage à la fois côté client et côté serveur. NodeJS utilise JavaScript sur le back-end. AngularJS est un framework qui apporte également la mise en œuvre de JavaScript au niveau du front-end. [11]

2.5.2 MVC-MVVM

MVC et MVVM sont des modèles de design pattern de structure.

MVC signifie Model-View-Controller et MVVM signifie Model-View-View-Model.

- Model : code qui s'occupe de la façon dont les données sont stockées ;
- View : code qui s'occupe de la façon dont les données sont affichées ;
- Controller : code qui s'occupe de la façon dont les données sont créées / mises à jour / supprimées ;
- ViewModel : code qui s'occupe à la fois de la façon dont les données sont stockées et de la façon dont elles sont affichées ;

Ces modèles aident à développer des applications qui sont librement couplées, faciles à tester et à entretenir. [11]

Notre travail regroupe ces deux modèles, en effet : on a adopté MVC au niveau du backend (Node.js) et le MVVM dans le frontend (Angular.js).

2.5.3 Vue d'ensemble de l'application

Les composants de notre système se réagissent entre eux pour répondre à nos besoins.

La figure suivante (figure 2.7) donne une vue d'ensemble de notre application et illustre les différents échanges entre les modules.

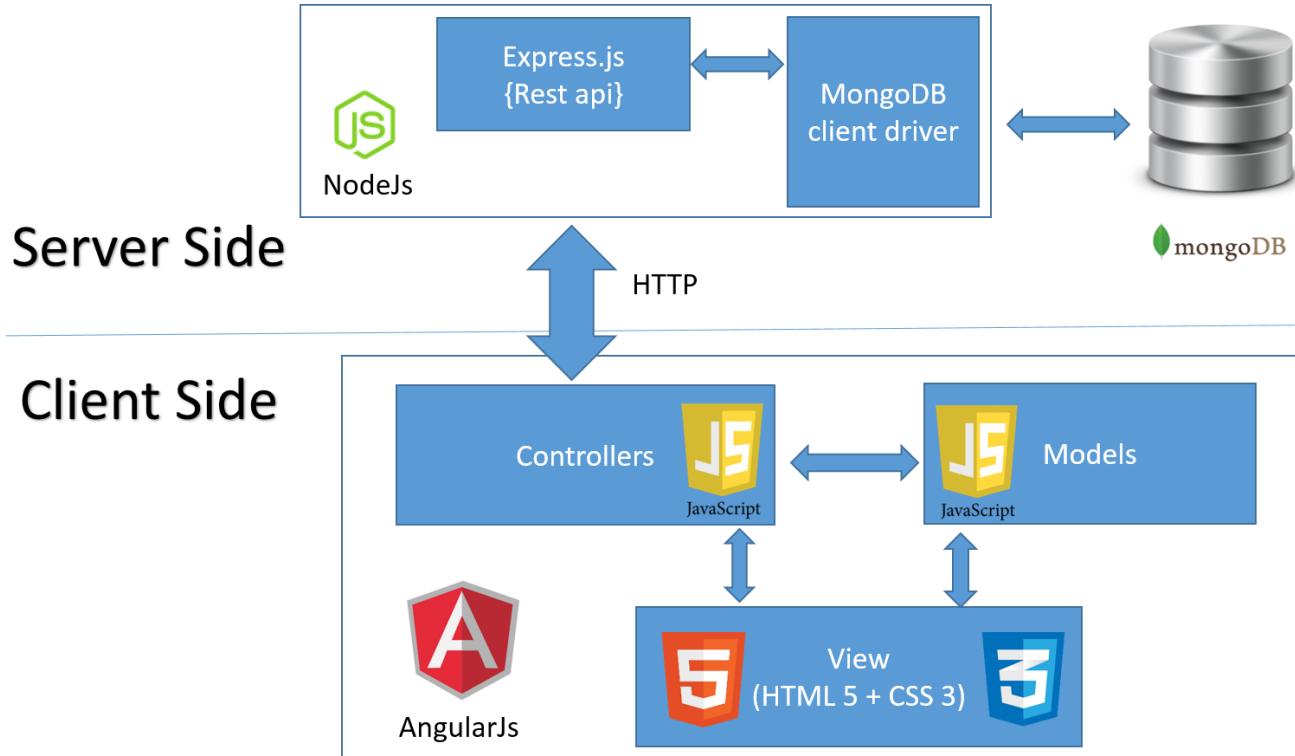


FIGURE 2.7 – Vue d'ensemble de l'application

Comme le montre l'image ci-dessus, nous avons AngularJS comme langage côté client qui traite la demande d'un client.

- Chaque fois qu'un utilisateur fait une demande (authentification, création d'événement, achat d'un billet,...), elle est d'abord traitée par AngularJS.
- Ensuite, la requête entre dans la deuxième étape, où nous avons Node.js comme langage côté serveur et ExpressJS comme framework web backend.
- Node.js gère les demandes client/serveur et ExpressJS fait une demande à la base de données.
- Dans la dernière étape, MongoDB (base de données) récupère les données et envoie la réponse à ExpressJS.
- ExpressJS renvoie la réponse à Nodejs et à son tour à AngularJS, puis affiche la réponse à l'utilisateur.[12]

2.6 Conclusion

Lors de ce chapitre, nous avons présenté l'étude conceptuelle que nous avons effectuée à travers la description des diagrammes de cas d'utilisation, de classes et de séquence.

Dans le chapitre suivant nous allons montrer les détails de la réalisation de notre application.

Chapitre 3

Réalisation

3.1 Introduction

Basé sur l'étude conceptuelle présentée dans le chapitre précédent, nous décrivons la phase de réalisation au niveau de cette dernière partie. Nous allons présenter au premier lieu, l'environnement du travail utilisé pour le développement de notre application. Par la suite, nous allons montrer les principales interfaces et leurs principes de fonctionnements.

3.2 Environnement de développement

Cette partie représente l'environnement de développement que nous avons choisi pour mettre en place notre application.

3.2.1 Environnement technique

MongoDB

MongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Le serveur et les outils sont distribués sous licence SSPL, les pilotes sous licence Apache et la documentation sous licence Creative Commons4. Il fait partie de la mouvance NoSQL. [13]

Express

ExpressJs est un framework Node.js qui permet la mise en place rapide d'un serveur de requêtes http. Il récupère ainsi les requêtes Ajax du client et les ajoute à la pile de requêtes qui doivent être traitées par Node.js. [14].

AngularJS

AngularJS est un framework Javascript totalement placé du côté client. AngularJS vous permet de mieux organiser votre code Javascript, en vue de créer des sites web dynamiques (bien qu'à la base, AngularJS n'avait pas été pensé pour du contenu dynamique). Basé du côté client, il vous permet de créer de l'HTML interactif. A part la librairie d'AngularJS. [14]

NodeJS

Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur, il utilise la machine virtuelle V8, la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications CommonJS.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js[15].

JWT : Json Web Token

JSON Web Token (JWT) est un standard ouvert. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique[16].

3.2.2 Environnement logiciel

Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence

de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré[17].



FIGURE 3.1 – Visual Studio Code

Postman

Postman est un logiciel qui permet de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en Collections. Cette classification permet notamment de regrouper des requêtes de façon « fonctionnelle » (par exemple enchaînement de création d'un événement).

A screenshot of the Postman application interface. At the top, there is a header bar with a 'POST' method button, a URL input field containing 'localhost:5000/api/event/create', and a large blue 'Send' button. Below the header, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body' (which is underlined in red), 'Pre-request Script', 'Tests', and 'Settings'. Under the 'Body' tab, there are five options: 'none', 'form-data' (which is selected and highlighted in orange), 'x-www-form-urlencoded', 'raw', 'binary', and 'GraphQL'. Below these options, there are three data entries in a table:

Key	Value	Description
description	enjoy salsa	
cat	607f5a1fac70ee23c4bcadc5	
picture	Select Files	

FIGURE 3.2 – Postman

Postman assure également la gestion des environnements, qui permet de contextualiser des variables et d'exécuter des requêtes ou des séries de requêtes dans différents configurations[18].



FIGURE 3.3 – Postman

mongodb compass

MongoDB Compass est un client graphique pour MongoDB. Compass se connecte au cluster MongoDB et présente deux onglets - Databases et Performance[19].



FIGURE 3.4 – mongodb compass

3.3 Interfaces de l'application

Après avoir explicité l'environnement de développement, nous exposons quelques interfaces réalisées de notre projet. Ces derniers sont les résultats de la consommations des apis dans la partie frontend.

3.3.1 Interface Principale

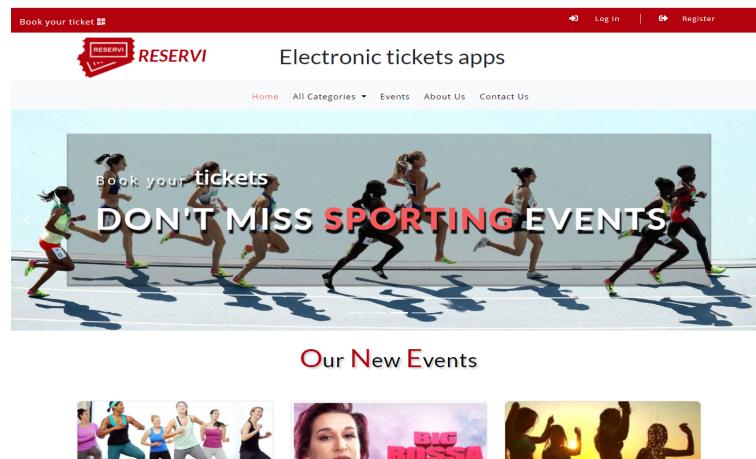


FIGURE 3.5 – Interface Principale

Cette interface représente l'interface principale l'utilisateur peut visualiser l'application sans authentification.

3.3.2 Interface Inscription

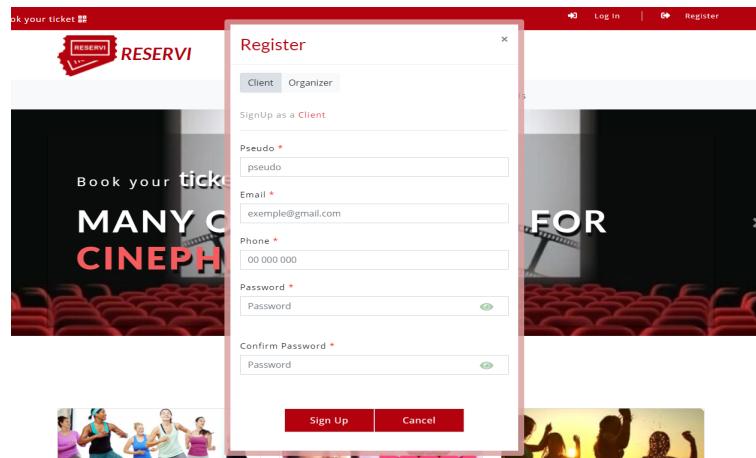


FIGURE 3.6 – Interface Inscription

Si l'utilisateur n'a pas de compte, il peut s'inscrire via cette interface.

3.3.3 Interface Authentification

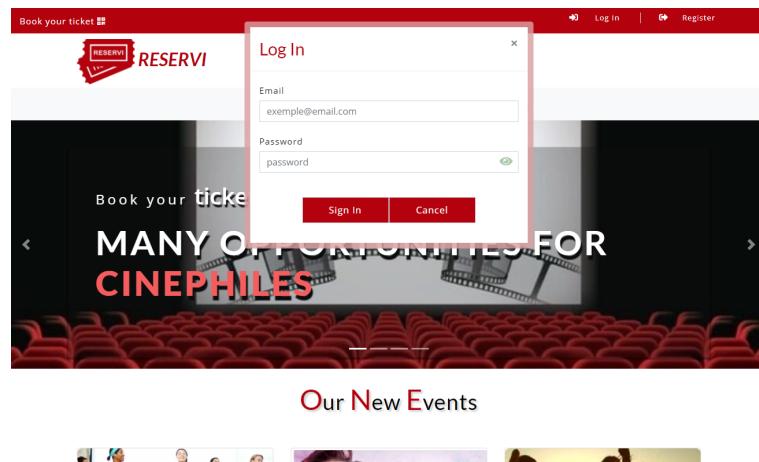


FIGURE 3.7 – Interface Authentification

Pour que l'utilisateur peut bénéficier des fonctionnalités de l'application, il doit s'authentifier.

3.3.4 Interface A propos

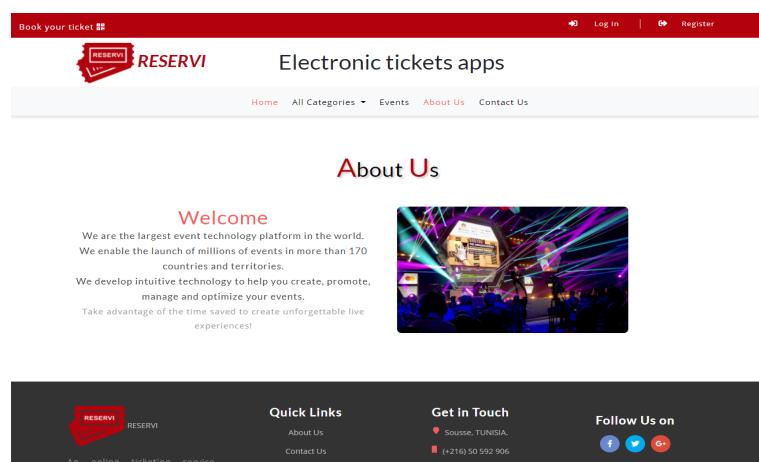


FIGURE 3.8 – Interface A propos

Cette interface contient une brève description de l'application.

3.3.5 Interface Contact

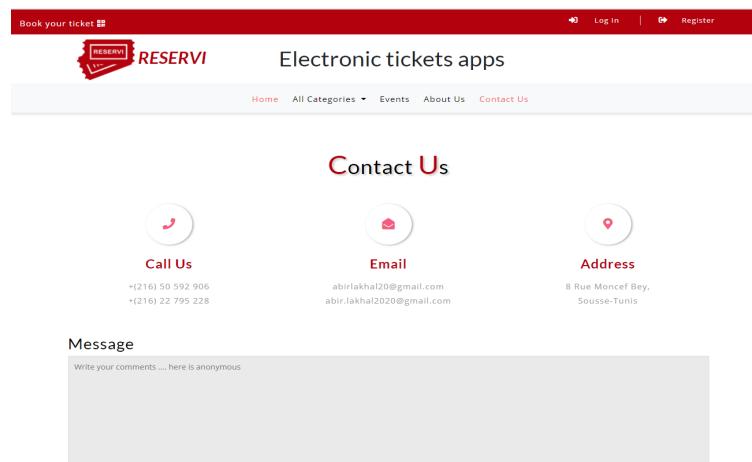


FIGURE 3.9 – Interface Contact

Dans cette interface il y a tous les contacts de l'administrateur, l'utilisateur peut aussi envoyer un message anonyme.

3.3.6 Interface Liste des événements

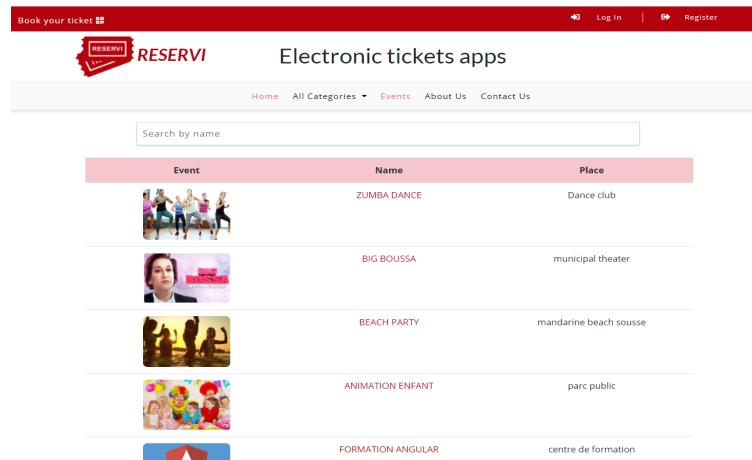


FIGURE 3.10 – Liste des événements

En navigant vers cette page, on trouve la liste de tous les événements. On peut faire la recherche d'un événement particulier à l'aide de la barre de recherche en tapant le nom.

3.3.7 Interface Details d'événement

The screenshot shows the RESERVIA website interface. At the top, there's a navigation bar with links for 'Home', 'All Categories', 'Events', 'About Us', and 'Contact Us'. On the right side of the header, there are 'Log In' and 'Register' buttons. Below the header, there's a banner for an event titled 'BIG BOSSA' featuring a woman's face. A text overlay below the banner reads: 'the big boussa is an ordinary woman, a civil servant like so many others'. To the right of the banner, event details are listed: 'Big Boussa' and '10 D', 'DATE: 30/05/2021', 'TIME: 11h', and 'PLACE: municipal theater'. Below these details are three buttons: '< < Go back', 'Add To favorites', and 'buy'. At the bottom of the page, there's a footer section with 'Quick Links' (About Us, Contact Us, Events), 'Get in Touch' (location: Sousse, Tunisia, phone: +(216) 50 592 906, email: abir.lakhal2020@gmail.com, abir.lakhal2020@gmail.com), and 'Follow Us on' social media links for Facebook, Twitter, and Google+.

FIGURE 3.11 – Details d'événement

Chaque événement à une interface qui affiche ses détails.

3.3.8 Interface Crédation d'événement

The screenshot shows the 'Create Event' form on the RESERVIA website. The form has fields for 'Event Title', 'Date' (format jj/mm/aaaa), 'Time' (dropdown menu), 'Event Place', 'Event Price', 'Event Description', 'Event Category', and a file upload field for 'Picture' with the placeholder 'Choisir un fichier' and 'Aucun fichier choisi'. At the bottom of the form are two buttons: 'Create' and 'Cancel'.

FIGURE 3.12 – Interface Crédation d'événement

En remplissant le formulaire, l'organisateur peut ajouter un événement.

3.3.9 Interface Paiement

The screenshot shows the payment page of the RESERVIT website. At the top, there's a red header bar with the text "Book your ticket" and "RAJA". Below it is a logo for "RESERVIT" with a red ticket icon. The main title "Electronic tickets apps" is centered above a form. The form fields include "Name on Card" (Abir Lakkal), "Credit card number" (1111-2222-3333-4444), "Exp Month" (September), "Exp Year" (2022), and "CVV" (352). At the bottom of the form are two buttons: "Continue to checkout" and "back >". Below the form is a dark footer bar with links for "Quick Links", "Get in Touch" (Sousse, TUNISIA), and "Follow Us on" (Facebook, Twitter, Google+).

FIGURE 3.13 – Interface Paiement

Cette interface est consacrée pour le paiement en ligne, le client remplit ce formulaire pour réservé et acheter son billet.

3.3.10 Interface billet

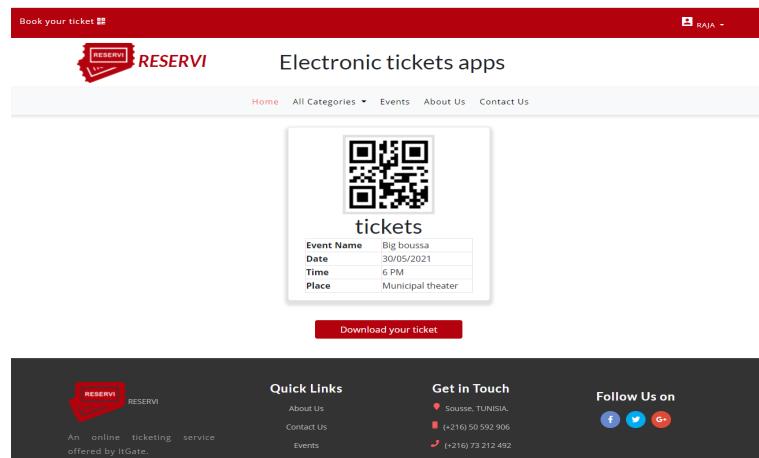


FIGURE 3.14 – Interface billet

Après avoir effectuer le paiement, le client reçoit son billet avec le QRCode et les détails de l'évènement.

3.3.11 Interface Profile

The screenshot shows the 'Clients' section of the RESERVIT website. At the top, there's a red header bar with the text 'Book your ticket' and 'RESERVIT'. Below it is a navigation bar with links for 'Home', 'All Categories', 'Events', 'About Us', and 'Contact Us'. On the right side of the header is a user icon labeled 'ABIR'. The main content area has a title 'Electronic tickets apps' and a sub-section 'Clients'. It displays a table with the following data:

Pseudo	abir
Email	abir@gmail.com
Phone	50592906
Institute	school
Account	123ase45

At the bottom of this section are two buttons: '< < Go back' and 'Edit'.

At the very bottom of the page is a dark footer bar with the RESERVIT logo, the text 'An online ticketing service', 'Quick Links' (with links to 'About Us' and 'Contact Us'), 'Get in Touch' (with address 'Sousse, TUNISIA.' and phone '+216 50 592 906'), and 'Follow Us on' with social media icons for Facebook, Twitter, and Google+.

FIGURE 3.15 – Interface Profile

Cette interface représente le profil du client, En cliquant sur "Edit" l'utilisateur peut modifier ses données.

3.3.12 Interface dashboard administrateur

Pseudo	Show	Update	delete
INES			
WASSIM			
HAJER			
MAHMOUD			
RAJA			
ZEINEB			
SARRA			

FIGURE 3.16 – dashboard administrateur

Cette partie est consacrée pour l’administrateur, via cette interface il peut visualiser et gérer les utilisateurs (clients et organisateurs), les événements et les messages.

3.4 Conclusion

Tout au long de ce dernier chapitre, nous avons décrit l’environnement de développement. Ensuite nous avons présenté quelques interfaces importantes et ses fonctionnalités de notre application.

Conclusion et perspectives

Ce rapport présente le travail que nous avons réalisés au sein de la société de développement informatique ItGate dans le cadre de notre projet de fin d'année.

L'étude conceptuelle approfondie guidé par les besoins détectés nous a permis de fonder une vue d'ensemble de projet où les composants MEAN .JS (MongoDB, Express, Angular, Node .JS) interagissent et faisant appel à certains modèles comme MVC, MVVM. Nous avons mis en œuvre cette étude dans les différents interfaces de notre application qui ont répondus aux besoins spécifiés.

Tout au long de la période de stage nous avons pu les compétences sollicités dans le cadre professionnel à savoir : gestion du temps et de stress, development best practice, l'esprit d'analyse et de critique, C'était également une opportunité de mettre en œuvre les notions techniques et académiques acquises tout au long de notre formation.

Ce stage nous a donc permis de confronter les problématiques que pose le passage de la théorie à la pratique.

Dans ce terme, nous avons fixé comme perspectives l'ajout d'une partie mobile qui permet aux organisateurs de contrôler l'accès des clients le jour de l'évènement l'accès des clients.

Bibliographie

- [1] <https://fr.wikipedia.org/wiki/billet> (dernière visite le 20/05/2020).
- [2] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/modele-en-cascade/> (dernière visite le 08/06/2020).
- [3] <https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm> (dernière visite le 08/06/2020).
- [4] <https://www.emse.fr/picard/cours/2a/gp/gp-introduction.pdf> (dernière visite le 08/06/2020).
- [5] [https://www.memoireonline.com/05/13/7195/m_{mise en place d'une application webmapping de geolocalisation des points d'interet de la ville}.html](https://www.memoireonline.com/05/13/7195/m_mise_en_place_dune_application_webmapping_de_geolocalisation_des_points_dinteret_de_la_ville.html) (dernière visite le 08/06/2020).
- [6] <https://fr.venngage.com/blog/modeles-diagrammes-gantt-projet/> : :text=un (dernière visite le 29/05/2020).
- [7] [https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-conception-d'un-système-informatique](https://www.ionos.fr/digitalguide/sites-internet/developpement-web/uml-un-langage-de-modelisation-pour-la-conception-d-un-systme-informatique/) (dernière visite le 29/05/2020).
- [8] https://fr.wikipedia.org/wiki/diagramme_de_séquence (dernière visite le 08/05/2020).
- [9] http://projet.eu.org/pedago/sin/isn/8-client_serveur.pdf (dernière visite le 08/05/2020).
- [10] <https://lerjen.me/mean-stack/> (dernière visite le 08/05/2020).
- [11] <https://www.javatpoint.com/architecture-of-mean-stack> (dernière visite le 08/05/2020).
- [12] [https://www.tutorialspoint.com/meanjs/meanjs-architecture](https://www.tutorialspoint.com/meanjs/meanjs_architecture) (dernière visite le 08/05/2020).
- [13] <https://fr.wikipedia.org/wiki/mongodb> (dernière visite le 30/06/2020).
- [14] <https://lerjen.me/mean-stack/> (dernière visite le 08/05/2020).

- [15] <https://fr.wikipedia.org/wiki/node.js> (dernière visite le 30/06/2020).
- [16] <https://fr.wikipedia.org/wiki/jsonwebtoken> (dernière visite le 02/05/2020).
- [17] <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/> (dernière visite le 30/06/2020).
- [18] <https://www.nuxeo.com/fr/blog/get-a-closer-look-at-your-nuxeo-data-with-mongodb-compass> (dernière visite le 30/06/2020).
- [19] https://fr.wikipedia.org/wiki/visual_studio_code (dernière visite le 30/06/2020).