

الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département de Mathématiques et de l'Informatique



Projet de Fin d'Etudes
Dans le cadre de l'arrêté ministériel 1275
« Un diplôme, une startup / micro entreprise ou brevet
d'invention »

Pour l'obtention du diplôme de Master
Filière : Informatique
Spécialité : Réseaux et Ingénierie des Données

E-Carte Universitaire

Présenté Par :

- | | | |
|--|----|------------------------------------|
| 1/ M. Garrouddji Mohammed Salah Eddine | M2 | Mathématiques et de l'Informatique |
| 2/ M. Meftahi Mohamed Amine | M2 | Mathématiques et de l'Informatique |
| 3/ Mme. Sedjai Bouchra | M2 | Mathématiques et de l'Informatique |

Devant le jury composé de :

Dr.MESSAOUDI Mohamed Amine	U.A.T	Président
Mme. SAIDI Samira	U.A.T	Examinateur
Dr. BOUTOUBA Mohammed Ben Abdelaziz	U.A.T	Examinateur
Dr. BELGRANA Fatima Zohra	U.A.T	Encadrante
Dr. GHERBI Sabah	U.A.T	Co-encadrante
Dr. BEMMOUSSAT Chems eddine	U.A.T	Représentant de l'incubateur
Pr. ZIADI Abdelkader	U.A.T	Partenaire socioéconomique
Pr. BELARBI Lahcene	U.A.T	Partenaire socioéconomique

Année Universitaire 2022/2023

Remerciements

Nous souhaitons exprimer notre profonde reconnaissance à **ALLAH** tout-puissant, dont la grâce et la volonté nous ont soutenus et guidés tout au long de cette période.

Nous tenons à adresser nos sincères remerciements à notre Encadrante, **Dr Belgrana Fatima Zohra**, pour sa présence constante, son soutien inconditionnel et ses précieux conseils tout au long de ce travail. Sa disponibilité et son expertise ont été essentielles à notre réussite.

Nous exprimons également notre gratitude envers notre Co-Encadrante, **Dr Gherbi Sabah**, pour sa présence continue, ses encouragements et ses précieux commentaires qui ont contribué à l'amélioration de notre mémoire.

Nous souhaitons adresser nos sincères remerciements à tous les membres du jury monsieur le directeur de l'université **Pr Ziadi Abdelkader**, monsieur le doyen de faculté Science et Technologie **Pr Belarbi Lahcene**, monsieur le directeur de l'incubateur **Dr Benmoussat Chems eddine**, **Dr**

Messaoudi Mohammed Amine, **Mme Saidi Samira** et **Dr Boutouba Mohammed Ben Abdelaziz** pour avoir accepté d'évaluer notre travail. Leur temps, leur expertise et leurs commentaires constructifs ont été d'une grande importance pour notre développement académique.

Nous tenons à exprimer toute notre gratitude envers tous les enseignants du département d'informatique qui ont contribué à notre formation tout au long de notre parcours universitaire. Leurs connaissances partagées et leurs efforts dévoués ont joué un rôle essentiel dans notre développement intellectuel.

Nos remerciements vont également à nos parents, qui nous ont soutenus et encouragés tout au long de cette aventure. Leur amour, leur soutien indéfectible et leurs encouragements constants ont été une source d'inspiration et de motivation.

Enfin, nous souhaitons adresser nos remerciements à tous ceux et celles qui, de près ou de loin, ont contribué à notre projet et à notre réussite. Nous sommes conscients qu'il y a probablement des personnes que nous avons omises, mais nous tenons à exprimer notre reconnaissance sincère à tous ceux qui ont apporté leur aide et leur soutien.

Que tous ceux et toutes celles dont nous sommes redevables trouvent ici nos remerciements les plus sincères.

Que **ALLAH** vous bénisse tous.

Résumé

Ce mémoire explore les avantages de l'utilisation des outils numériques dans le campus universitaire et présente une application qui vise à résoudre les différents défis rencontrés. En mettant en avant les avantages tels que l'efficacité, l'accessibilité, la sécurité, la conservation des données et l'amélioration de l'expérience des étudiants, ce projet cherche à transformer et à optimiser les services offerts par l'université. Notre projet proposé est une E-carte universitaire visant cinq principaux objectifs : le paiement électronique des services au sein du campus, la géolocalisation interne, la gestion numérique des présences, le stockage des informations personnelles ainsi que la planification des tâches.

Ce manuscrit décrit les différentes étapes du projet, en commençant par une analyse de l'existant et en suivant des spécifications formelles et techniques. Nous mettons notamment l'accent sur les systèmes d'exploitation mobiles, les outils utilisés ainsi que les étapes de conception et de développement de notre application. Les connaissances dans les domaines juridiques, marketing et financier sont des éléments clés ayant contribué à l'intégration réussie des différents aspects du projet sous forme d'un BMC.

Notre prototype met en évidence les impacts positifs de l'utilisation des outils numériques sur la vie étudiante, en offrant une meilleure accessibilité aux informations, une communication plus efficace avec les enseignants et le personnel administratif, ainsi qu'une amélioration générale des services universitaires. Notant toutes, qu'il est également question d'assurer une évolution continue de notre E-carte universitaire grâce à l'utilisation des technologies numériques, dans le but de répondre aux besoins changeants de la communauté étudiante

Abstract

This thesis explores the benefits of using digital tools on the university campus and presents an application that aims to address various challenges. By emphasizing advantages such as efficiency, accessibility, security, data preservation, and improved student experience, this project seeks to transform and optimize the services offered by the university. Our proposed project is an E-campus card with five main objectives: electronic payment for campus services, internal geolocation, digital attendance management, storage of personal information, and task planning.

This manuscript describes the different stages of the project, starting with an analysis of the current situation, followed by formal and technical specifications. We specifically focus on mobile operating systems, tools used, as well as the design and development steps of our application. Knowledge in legal, marketing, and financial domains are key elements that have contributed to the successful integration of different aspects of the project in the form of a Business Model Canvas (BMC).

Our prototype highlights the positive impacts of using digital tools on student life by providing better accessibility to information, more effective communication with teachers and administrative staff, and overall improvement of university services. It is worth noting that ensuring the continuous evolution of our E-campus card through the use of digital technologies is also important in order to meet the changing needs of the student community.

ملخص

هذه المذكرة تستكشف فوائد استخدام الأدوات الرقمية في الحرم الجامعي وتقدم تطبيقاً يهدف إلى التغلب على التحديات المختلفة. من خلال التركيز على مزايا مثل الكفاءة، وإمكانية الوصول، والأمان، والحفاظ على البيانات، وتحسين تجربة الطلاب، يسعى هذا المشروع لتحويل وتحسين الخدمات التي تقدمها الجامعة. مشروعنا المقترح هو بطاقة جامعية إلكترونية تهدف إلى تحقيق خمسة أهداف رئيسية : الدفع الإلكتروني للخدمات في الحرم الجامعي، والتحديد المكاني الداخلي، وإدارة الحضور الرقمية، وتخزين المعلومات الشخصية، وتخطيط المهام.

تصف هذه المذكرة المراحل المختلفة للمشروع، بدءاً من تحليل الوضع الحالي، مروراً بالموصفات الرسمية والفنية. وضعنا تركيزاً خاصاً على أنظمة التشغيل المحمولة، والأدوات المستخدمة، بالإضافة إلى خطوات تصميم وتطوير تطبيقنا. إضافة لاكتساب المعرفة في المجالات القانونية والتسويق والمالية و التي هي عناصر أساسية ساهمت في الاندماج الناجح لمختلف جوانب المشروع في شكل مخطط نموذج العمل التجاري.

يسلط نموذجنا التجريبي الضوء على الآثار الإيجابية لاستخدام الأدوات الرقمية في حياة الطلاب، من خلال توفير إمكانية الوصول إلى المعلومات بشكل أفضل، وتحسين التواصل مع الأساتذة والموظفين الإداريين، وتحسين الخدمات الجامعية بشكل عام. يجب ملاحظة أن ضمان تطوير بطاقةنا الجامعية الإلكترونية بشكل مستمر باستخدام التكنولوجيا الرقمية مهم أيضاً لتلبية الاحتياجات المتغيرة لمجتمع الطلاب.

Table des matières

Chapitre 1	Introduction générale	1
Chapitre 2	Contexte et Objectifs	4
2.1	Introduction	5
2.2	Contexte	5
2.3	Objectifs	7
2.3.1	Objectifs principale	7
2.3.2	Objectifs informatique	8
Chapitre 3	Outils de développement,modélisation et no- tion de multiplateforme	10
3.1	Introduction	11
3.2	Type d'application	12
3.2.1	Application mobile	12
3.2.2	Application de bureau	13
3.2.3	Application web	15
3.3	Système d'exploitation	15
3.3.1	Les systèmes d'exploitation mobiles	16
3.3.2	Les systèmes d'exploitation pour ordinateurs	18
3.4	Les serveurs	19
3.5	Outils développement	20

3.5.1	Choix des outils de développement	20
3.6	Outil de modélisation : UML	24
3.6.1	Types des diagrammes	25
3.7	Conclusion	27
Chapitre 4	Analyse et conception	29
4.1	Introduction	30
4.2	Analyse de l'existant	30
4.2.1	La description détaillée de la carte	30
4.2.2	L'historique de la carte	31
4.3	Spécification des besoins	32
4.3.1	Les besoins fonctionnels	32
4.3.2	Les besoins non fonctionnels	33
4.4	Spécification techniques de l'IHM d'ECU	34
4.4.1	Les spécifications Conceptuelles	34
4.4.2	Les spécifications syntaxiques	35
4.4.3	Les spécifications lexicales	37
4.5	Conclusion	38
Chapitre 5	Présentation du prototype	39
5.1	Introduction	40
5.2	Architecteur de l'application	40
5.3	Présentation de l'application	41
5.3.1	Fenêtre « Authentification »	41
5.3.2	Fenêtre « S'inscrire »	43
5.3.3	Fenêtre « Page d'accueil »	43
5.3.4	Fenêtre « Plan de campus universitaire »	45
5.3.5	Fenêtre « Profile »	46

5.3.6	Fenêtre « La liste des tâches »	47
5.3.7	Fenêtre « E-présence »	48
5.3.8	Fenetre « E-Paiement »	51
5.4	Conclusion	54
Chapitre 6	Conclusion générale	56
Bibliographie		59

Table des figures

3.1	Quelques applications mobiles : (a) sur un smartphone, (b) sur Google Play store.	12
3.2	Quelques représentant les applications de bureau (Linux mint).	14
3.3	Quelques applications dans la boutique des logicieles en linux mint.	14
3.4	L'icône représentant le système Android.	17
3.5	L'icône représentant le système iOS.	17
3.6	L'icône représentant le système Windows Phone.	17
3.7	BlackBerry OS	18
3.8	L'icône représentant le système Windows OS.	18
3.9	L'icône représentant le logo de système mac OS.	18
3.10	L'image représentant le logo de LinuxOS.	19
3.11	Chrome OS.	19
3.12	L'icône représentant le logo de Flutter.	21
3.13	Firebase	22
5.1	Fenêtre "Authentification" : (a) Mode sombre version mobile en français,(b) Mode clair version mobile en anglais,(c) version bureau en arabe.	42
5.2	Fenêtre d'inscription version mobile : (a) en français en mode clair, (b) en anglais en mode sombre.	43
5.3	Fenêtre d'accueil	44

5.4	Fenêtre du plan du campus universitaire (géolocalisation intra-campus).	45
5.5	Fenêtre du profile affichant les informations personnelles. . . .	46
5.6	Fenêtre de planification des tâches ,version en anglais ToDo list : (a) Liste des tâches, (b) planification de la tâche, (c)Moyen de suppression d'une tâche.	47
5.7	Gestion des modules : (a) Ajout d'un module, (b) la liste des modules.	48
5.8	Choix de la méthode de gestion de la présence des étudiants. .	49
5.9	Liste des étudiants(méthode manuelle).	50
5.10	La fenêtre indique que la présence est marquée.	51
5.11	Envoi de crédits : (a) choix du montant, (b) utilisation du code QR pour recevoir les crédits.	52
5.12	Achat de services : (a) saisir le montant d'achat, (b) choix du service.	53
5.13	Fenêtre d'affichage des transactions.	54

Liste des tableaux

4.1	Spécification des besoins fonctionnels de notre projet.	33
4.2	Extrait de spécifications conceptuelles : présentation de l'objet menu principale.	35

Chapitre 1

Introduction générale

Depuis l'avènement de l'informatique, de nombreux aspects de la vie quotidienne ont été simplifiés. Aujourd'hui, les logiciels et les réseaux informatiques facilitent le traitement de l'information, permettant aux individus de démontrer et de mettre en valeur leurs talents et leurs compétences. Cette évolution a touché de nombreux secteurs d'activité et est devenue une caractéristique essentielle des grandes entreprises.

Les smartphones et les ordinateurs ont largement remplacé les papiers, les calculatrices, les radios et même les boîtes aux lettres. L'objectif des professionnels de l'informatique est de développer cette technologie dans tous les domaines de la vie quotidienne. L'informatique offre de nombreux outils pour la transmission de l'information, permettant aux utilisateurs d'accéder rapidement et facilement à des informations sur des sujets variés. Dans l'ère de la numérisation et de la technologie, avec le développement de l'intelligence artificielle et de l'apprentissage automatique, nous pouvons nous attendre à des avantages et à des améliorations continues dans nos vies et nos affaires. Nous pouvons envisager des applications plus avancées dans des domaines tels que la robotique, la médecine, les transports, l'agriculture, et bien d'autres,

qui auront un impact significatif sur nos vies et nos communautés. Autant qu'étudiant, nous avons constaté durant nos cinq années d'études l'existence de certains problèmes au niveau du campus universitaires, tels que :

1. Problème des paiements en espèces pour les services au sein du campus.
2. Difficulté à gérer et à surveiller la présence des étudiants lors des cours, des travaux dirigés et des travaux pratiques, surtout lorsque le nombre des étudiants est élevé.
3. Difficulté à localiser des endroits spécifiques sur le campus.
4. Dépendance excessive aux documents papier, avec des risques de perte, de détérioration et de difficulté de récupération.
5. Affluence élevée et temps d'attente importants afin d'acquérir des services universitaires.

Nous avons décidé de profiter de cette évolution technologique et numérique, alors nous avons proposé une solution à ces problèmes-là, en développant une application multiplateforme (Android, iOS, Bureau) offrant plusieurs services regroupés dans une seule E-Carte Universitaire, dans le but de faciliter la vie étudiante sur le campus universitaire. L'objectif est de concevoir et de réaliser une solution qui permettra :

- Paiement électronique des services au sein du campus (porte-monnaie électronique).
- La gestion numérique de la présence des étudiants en classe.
- La géolocalisation des salles, des amphithéâtres, des départements et d'autres lieux sur le campus.
- Le stockage des documents importants au format numérique.
- Nous proposons également d'autres outils, tels que la planification des tâches.

En intégrant ces solutions, le personnel universitaire ainsi que les étudiants bénéficieront d'une meilleure expérience au sein du campus universitaire grâce aux services proposés par notre E-carte, qui offrent une certaine rapidité, sécurité et efficacité.

Ce mémoire est composé de quatre chapitres : Le premier chapitre expose le contexte et les objectifs du projet, quant au deuxième chapitre, il traite le concept de multiplateforme et expose des outils utilisés. Le troisième chapitre aborde les phases d'analyse et de conception de notre application. Enfin, le quatrième et dernier chapitre est dédié à la présentation du prototype développé. Le mémoire contient également notre BMC comme annexe.

Chapitre 2

Contexte et Objectifs

Contenu

2.1	Introduction	5
2.2	Contexte	5
2.3	Objectifs	7
2.3.1	Objectifs principale	7
2.3.2	Objectifs informatique	8

2.1 Introduction

Notre projet de E-Carte Universitaire "**ECU**" a de grandes potentialités pour améliorer la vie des étudiants de différentes façons. La carte peut inclure les informations personnelles de l'étudiant, telles que le nom, l'adresse, la date de naissance, ect, ainsi que les informations académiques, les présences, les absences, etc. En outre, cette carte peut être intégrée à divers systèmes universitaires pour permettre aux étudiants d'accéder à des services de manière plus efficace et pratique. La carte peut également être utilisée pour les transactions financières, telles que les frais de scolarité, les dépenses en cafétéria, etc. La mise en œuvre d'une telle carte d'étudiant électronique peut également aider les universités à collecter des données sur les étudiants et à améliorer leur expérience en fournissant des services personnalisés en fonction de leurs besoins et de leurs intérêts. Il est important de noter que la sécurité des données est une préoccupation majeure lors de la mise en œuvre d'une telle carte, il est donc important de prendre des mesures de sécurité adéquates pour protéger les informations personnelles et financières des étudiants. En résumé, une carte d'étudiant électronique peut améliorer considérablement la vie des étudiants en leur fournissant un moyen pratique et efficace d'accéder à divers services universitaires et en aidant les universités à fournir des services de meilleure qualité .

2.2 Contexte

Au cours de nos années d'études universitaires, la carte d'étudiant se limiter hélas à identifier les étudiants afin par exemple de passer des examens, mais avec les avancées technologiques actuelles, les cartes étudiantes

à l'échelle international ne sont plus seulement des moyens d'identification, elles sont devenues de plus en plus sophistiquées et intègrent des fonctionnalités électroniques (numériques) avancées.

Les technologies de l'information et de la communication ont permis aux universités étrangères de proposer des cartes étudiantes intelligentes qui offrent une multitude de fonctionnalités pour améliorer la vie des étudiants.

Ces cartes universitaires électroniques permettent aux étudiants de gérer de manière plus efficace leurs transactions et leurs interactions avec l'université et ses services universitaires. Grâce à la technologie mobile et à Internet, les étudiants peuvent facilement accéder à des informations d'ordre pédagogique tels que les emplois du temps, leurs horaires de cours et d'autres données importantes.

Les cartes universitaires électroniques doivent offrir essentiellement une sécurité accrue pour les étudiants et les enseignants universitaires et le personnel de l'université, les données collectées à partir de ces cartes peuvent également être utilisées pour améliorer la planification et de gestion des tâches pédagogiques.

En somme, les cartes universitaires électroniques peuvent transformer la façon dont les étudiants interagissent avec l'université et faciliter ainsi leur vie quotidienne. Les universités peuvent utiliser ces cartes pour offrir des services de qualité supérieure et améliorer leur efficacité, tout en offrant aux étudiants une expérience universitaire plus sûre et plus agréable. Notant toutefois que cette carte pourra notamment être utilisée par les œuvres universitaires qui subissent à leur tour une numérisation.

2.3 Objectifs

2.3.1 Objectifs principale

La carte étudiante est généralement émise par une institution d'enseignement dans une université pour identifier les étudiants inscrits et leur fournir l'accès à différents services .Elle est souvent considérée comme une preuve d'identité pour les étudiants et peut être requise pour accéder aux bâtiments universitaires, aux salles de cours, aux bibliothèques, aux centres de loisirs, etc.

Les cartes étudiantes traditionnelles étaient généralement en papier avec la photo de l'étudiant et d'autres informations de base imprimées dessus. Cependant, avec l'avancée de la technologie, de nombreuses institutions éducatives ont commencé à utiliser des cartes universitaires électroniques qui offrent une gamme de fonctionnalités supplémentaires, comme nous l'avons mentionné dans la section précédente.

Les cartes électroniques universitaires ont une utilité bien plus large que simplement identifier les étudiants, les enseignants et le personnel administratif. Elles offrent la possibilité de régler les services proposés par l'établissement ainsi que les activités culturelles et associatives qui y sont proposées, ainsi que tout autre produit ou service disponible en interne. En outre, elles permettent le contrôle d'accès et offrent des avantages spéciaux. Ces cartes améliorent l'expérience des étudiants en leur offrant un accès plus rapide et plus efficace à une gamme de services variés.

2.3.2 Objectifs informatique

Afin d'atteindre notre objectif principal, il est question de développer une application multiplateforme rattachée à cette dite **E-Carte Universitaire**. C'est une application multiplateforme pour l'université destinée aux étudiant, enseignant, administrateur, l'agent, et l'agent transactionne, dont l'objectif est de fournir une plateforme pratique et accessible afin de leur permettre d'accéder à leurs informations académiques et administratives et pédagogique. L'objectif est de simplifier et d'optimiser l'expérience universitaire pour les étudiants, en les aidants à rester informés en temps réel, tout en réduisant la charge de travail du Les enseignants universitaires et Le personnel administratif universitaire. Certains des objectifs spécifiques de cette application mobile dédiée à l'étudiant ainsi qu'à l'université peuvent inclure :

1. **Améliorer la communication** : l'application vise à améliorer la communication entre les étudiants et le personnel, avec des fonctionnalités telles que les notifications et le fil d'actualités pour tenir les étudiants informés des mises à jour importantes.
2. **Améliorer la productivité** : en fournissant un accès rapide et facile à des informations académiques et administratives, l'application aide les étudiants à rester organisés et à gérer leur temps de manière plus efficace.
3. **Gestion numérique de la présence des étudiants en classe** : fait référence à l'utilisation de technologies numériques pour suivre et enregistrer la présence des étudiants lors des cours. Cela permet aux établissements d'enseignement de collecter des données précises sur la présence des étudiants, de manière automatisée et efficace, en remplaçant les méthodes traditionnelles telles que l'appel nominal manuel.

4. **Paiements électroniques** : la carte peut être utilisée comme une carte de paiement électronique, permettant aux étudiants de payer pour les repas, les services de transport, etc.
5. **Soutenir le succès des étudiants** : En proposant des ressources telles que le suivi des présences, ainsi qu'un GPS académique.
6. **Contrôle d'accès** : La carte peut être utilisée pour accéder aux bâtiments de l'université ou pour vérifier la présence des étudiants lors des cours ou des examens.
7. **Réductions** : la carte peut également offrir des réductions ou des avantages spéciaux pour les étudiants auprès de certains magasins ou restaurants.

Chapitre 3

Outils de développement, modélisation et notion de multiplateforme

Contenu

3.1	Introduction	11
3.2	Type d'application	12
3.2.1	Application mobile	12
3.2.2	Application de bureau	13
3.2.3	Application web	15
3.3	Système d'exploitation	15
3.3.1	Les systèmes d'exploitation mobiles	16
3.3.2	Les systèmes d'exploitation pour ordinateurs	18
3.4	Les serveurs	19
3.5	Outils développement	20
3.5.1	Choix des outils de développement	20
3.6	Outil de modélisation : UML	24
3.6.1	Types des diagrammes	25
3.7	Conclusion	27

3.1 Introduction

Le développement multiplateforme est une approche de développement logiciel qui vise à créer des applications qui peuvent fonctionner sur plusieurs systèmes d'exploitation et/ou sur plusieurs types de matériel. Les développeurs utilisent cette approche pour économiser du temps et de l'argent en créant une seule application qui fonctionne sur plusieurs plates-formes, plutôt que de développer des applications séparées pour chaque plate-forme.

Pour développer des applications multiplates-formes, les développeurs utilisent des outils qui offrent des fonctionnalités telles que des bibliothèques de code partagé, des API compatibles avec plusieurs plates-formes, des compilateurs et des émulateurs pour tester les applications sur différentes plates-formes.

Le développement multi-plateforme est devenu de plus en plus important ces dernières années en raison de la croissance du marché des smart phones et tablettes, il est devenu alors nécessaire pour les développeurs de créer des applications qui peuvent fonctionner sur plusieurs plates-formes pour atteindre un public plus large et économiser du temps et de l'argent, effectivement cela permet une réduction des coûts de développement en n'ayant à créer qu'une seule application, il permet également une facilité de maintenance en ayant un seul code base pour toutes les plates-formes

Par conséquent, ce chapitre sera dédié à présenter quelques notions sur les principaux types d'application, mobile, de bureau et web, nous verrons également plusieurs systèmes d'exploitation, ainsi que différents outils de développement, nous clôturons ce chapitre par une présentation de l'outil de modélisation que nous avons opté pour, qui est l'UML en illustrant ses différents types de diagramme.

3.2 Type d'application

3.2.1 Application mobile

Une application mobile est un logiciel conçu pour être exécuté sur un appareil mobile, tel qu'un smart phone ou une tablette. Ce type d'application est souvent téléchargé à partir d'une boutique en ligne, telle que l'App Store d'Apple ou Google Play pour Android. Ces applications sont conçues pour fournir des fonctionnalités spécifiques à l'utilisateur, telles que les médias sociaux, les jeux, les outils de productivité, les services bancaires, etc. Les applications mobiles sont souvent utilisées pour interagir avec le monde qui nous entoure, en utilisant les fonctionnalités intégrées de l'appareil mobile, telles que l'appareil photo, la géolocalisation, les capteurs de mouvement, etc.

(7)

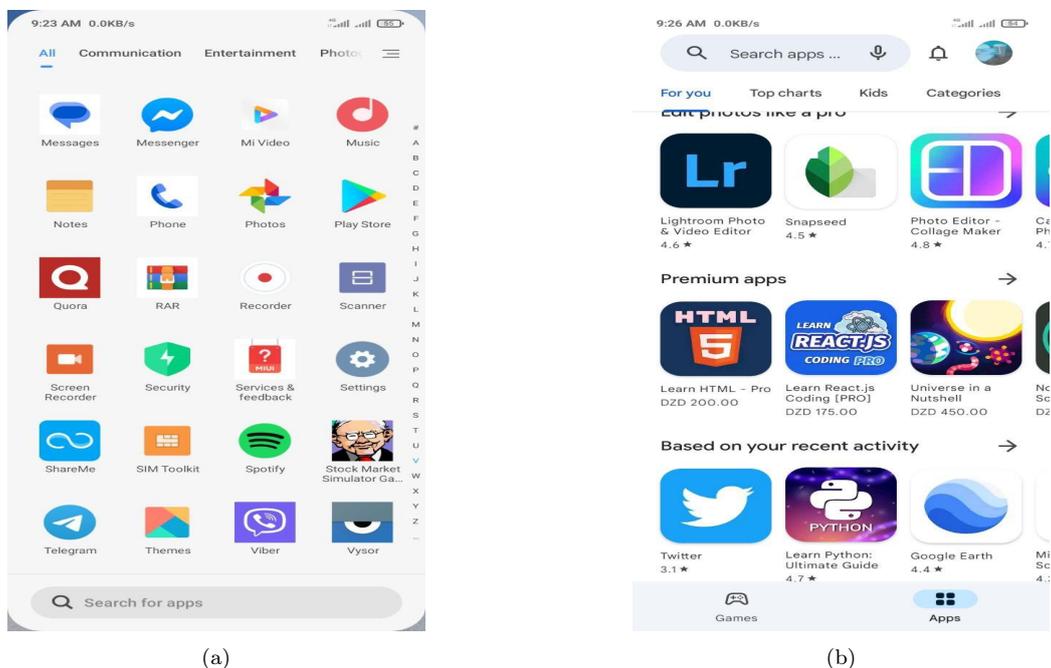


FIGURE 3.1 – Quelques applications mobiles : (a) sur un smartphone, (b) sur Google Play store.

Certaines applications mobiles sont gratuites, tandis que d'autres peuvent être payantes, cependant les développeurs peuvent également inclure des publicités dans leurs applications mobiles pour générer des revenus supplémentaires. L'architecture de ces applications doit prendre en compte un certain nombre de contraintes de conception, telles que les ressources limitées, les problèmes de connectivité, les modèles de saisie des données et les différentes résolutions d'affichage des appareils mobiles.

En résumé, une application mobile est un logiciel spécialement conçu pour fonctionner sur un appareil mobile, offrant des fonctionnalités adaptées aux utilisateurs ainsi qu'un accès facile via des plateformes de téléchargement d'applications, certaines peuvent même fonctionner en mode déconnecté.

3.2.2 Application de bureau

Une application de bureau est un logiciel qui est installé sur un ordinateur local, plutôt que d'être exécuté dans un navigateur web. Les applications de bureau sont souvent téléchargées à partir d'un site web de l'éditeur ou d'une boutique d'applications, et sont conçues pour fournir des fonctionnalités spécifiques à l'utilisateur, telles que le traitement de texte, la gestion de projets, l'édition de photos, les jeux, etc. (22)

Les applications de bureau peuvent être développées pour une plateforme spécifique, telle que Windows, macOS ou Linux, et sont conçues pour fonctionner de manière autonome sur un ordinateur. Elles ont généralement une interface utilisateur graphique (GUI) pour faciliter l'interaction avec l'utilisateur et peuvent utiliser les fonctionnalités de l'ordinateur local, telles que les fichiers locaux, l'imprimante, les dispositifs de stockage externes, etc.

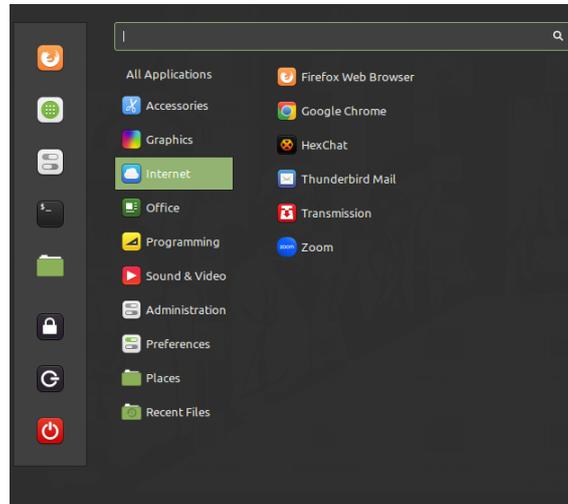


FIGURE 3.2 – Quelques représentations des applications de bureau (Linux mint).

Les applications de bureau ont généralement des IHMs (Interface Homme Machine) plus riches et des fonctionnalités plus avancées que les applications web. Elles sont conçues pour être utilisées sur des ordinateurs locaux plutôt que dans un navigateur web. Les applications de bureau peuvent être téléchargées à partir de sites web de l'éditeur ou de plateformes de distribution d'applications telles que le Microsoft Store ou le Software Manager de linux. Il convient de noter que les applications de bureau sont généralement destinées à un usage individuel et ne sont pas conçues pour être exécutées sur plusieurs appareils en même temps, contrairement aux applications web ou mobiles.

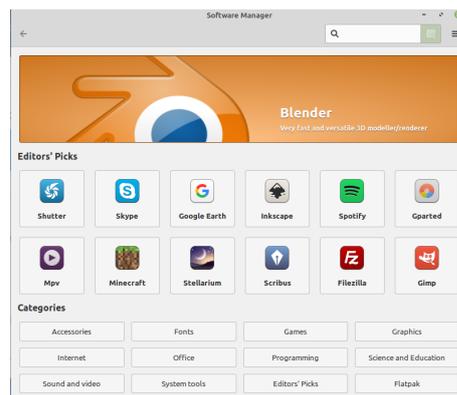


FIGURE 3.3 – Quelques applications dans la boutique des logiciels en linux mint.

3.2.3 Application web

Une application web est un logiciel accessible via un navigateur web, qui permet aux utilisateurs d'effectuer différentes tâches et d'interagir avec des données ou des services en ligne. Contrairement aux applications traditionnelles qui nécessitent une installation sur un ordinateur ou un appareil mobile, les applications web sont hébergées sur des serveurs distants et sont accessibles à partir de n'importe quel navigateur web, offrant ainsi une grande flexibilité d'accès.

Les applications web sont généralement développées en utilisant des technologies web telles que HTML (HyperText Markup Language), CSS (Cascading Style Sheets) et JavaScript, qui permettent de créer des interfaces utilisateur interactives et dynamiques. Elles peuvent également utiliser des frameworks et des bibliothèques populaires tels que React, Angular ou Vue.js pour faciliter le développement.⁽²⁷⁾

3.3 Système d'exploitation

Un système d'exploitation est un programme qui agit comme un intermédiaire entre l'utilisateur d'un ordinateur et le matériel de l'ordinateur. Il fournit des services aux applications (ou logiciel) et gère les ressources du système, y compris les processeurs, la mémoire, les périphériques d'entrée/sortie et le stockage, et ceci de façon contrôlée et ordonnée. Il permet également à plusieurs utilisateurs et programmes d'accéder aux ressources du système simultanément, en garantissant que les opérations ne se chevauchent pas et ne causent pas d'interférences.

Les systèmes d'exploitation sont responsables de la gestion de toutes les ressources du système, telles que les processeurs, la mémoire, les périphé-

riques d'entrée/sortie et le stockage. Ils fournissent également des services tels que la gestion, des dossiers, des fichiers, la gestion des utilisateurs et des comptes ainsi que de la sécurité et la protection contre les virus et les logiciels malveillants. (28)

Les systèmes d'exploitation sont utilisés sur divers types de périphériques informatiques, tels que les ordinateurs de bureau, les ordinateurs portables, les serveurs, les smart phones, les tablettes et les appareils embarqués. Ils sont alors divisés en deux catégories, mobile et ordinateur.

3.3.1 Les systèmes d'exploitation mobiles

Les systèmes d'exploitation mobiles sont des logiciels qui permettent de gérer les ressources matérielles et logicielles d'un appareil mobile tel qu'un smartphone, une tablette ou une montre connectée. Ils offrent un environnement d'exécution pour les applications mobiles et permettent aux utilisateurs d'interagir avec leur appareil.

Les systèmes d'exploitation mobiles les plus courants sont Androïde, iOS, Windows Phone et BlackBerry OS. Chacun de ces systèmes d'exploitation est conçu pour fonctionner sur des matériels spécifiques, bien qu'Android soit disponible sur un large éventail de périphériques. Les systèmes d'exploitation mobiles offrent des services tels que la gestion des applications, la messagerie, la navigation, les appels téléphoniques, la gestion des contacts, la prise de le système photos et vidéos, et bien d'autres fonctionnalités.

Les systèmes d'exploitation mobiles sont développés avec des contraintes particulières, comme la gestion des ressources limitées, la durée de vie de la batterie, l'utilisation des réseaux mobiles et la sécurité. Les développeurs d'applications mobiles doivent tenir compte de ces contraintes pour créer des applications qui fonctionnent efficacement sur ces plateformes. (15)

Android

Développé par Google, Android est l'un des systèmes d'exploitation mobiles les plus utilisés dans le monde. Il est open source, personnalisable et est disponible sur une large gamme de smartphones et de tablettes. (10)



FIGURE 3.4 – L'icône représentant le système Android.

iOS

Développé par Apple, iOS est exclusif aux appareils Apple tels que l'iPhone et l'iPad. Il est connu pour son interface utilisateur élégante, sa sécurité élevée et sa simplicité d'utilisation.(3)



FIGURE 3.5 – L'icône représentant le système iOS.

Windows Phone

Développé par Microsoft, Windows Phone a une interface utilisateur unique et est disponible sur une gamme de téléphones Nokia Lumia. Cependant, sa part de marché est relativement faible par rapport à Android et iOS.



FIGURE 3.6 – L'icône représentant le système Windows Phone.

BlackBerry OS

Développé par BlackBerry, ce système d'exploitation mobile est populaire dans les entreprises en raison de ses fonctions de sécurité renforcées et de son clavier physique. Cependant, il a également une part de marché relativement faible. (6)



FIGURE 3.7 – BlackBerry OS

3.3.2 Les systèmes d'exploitation pour ordinateurs

Windows

Développé par Microsoft, c'est l'un des systèmes d'exploitation les plus populaires au monde. Il est utilisé par la plupart des ordinateurs personnels et professionnels. (20)



FIGURE 3.8 – L'icône représentant le système Windows OS.

macOS

Développé par Apple, c'est le système d'exploitation utilisé sur les ordinateurs Mac. Il est connu pour sa stabilité, sa sécurité. (4)



FIGURE 3.9 – L'icône représentant le logo de système mac OS.

Linux

Il s'agit d'un système d'exploitation open source, basé sur le noyau Linux. Il est populaire pour les serveurs, les supercalculateurs et les ordinateurs personnels. Il existe de nombreuses distributions de Linux, telles que Ubuntu, Fedora, Debian, etc.(16)



FIGURE 3.10 – L'image représentant le logo de LinuxOS.

Chrome OS

Développé par Google, il est basé sur le noyau Linux et est conçu pour être utilisé avec les ordinateurs portables Chromebook.(12)



FIGURE 3.11 – Chrome OS.

Il existe également d'autres systèmes d'exploitation mobiles moins connus, tels que Sailfish OS, Ubuntu Touch et Firefox OS.

3.4 Les serveurs

Un serveur est un ordinateur qui fournit des services à d'autres ordinateurs qui sont appelés clients dans un réseau. Le serveur gère l'accès à ses services et fournit des informations ou des fonctions en réponse à des demandes des clients.(28)

Les serveurs sont souvent gérés par des administrateurs de système qui veillent à leur bon fonctionnement et à leur sécurité. Ces administra-

teurs peuvent configurer les serveurs pour répondre aux besoins spécifiques de l'organisation ou de l'utilisateur, et s'assurer que les données stockées sur les serveurs sont sauvegardées et protégées contre les intrusions ou les pannes.

Les technologies concernant les serveurs ont évolué au fil des ans, avec des améliorations constantes de la puissance de traitement, de la capacité de stockage et de la sécurité. Les serveurs modernes sont souvent équipés de multiples processeurs, de grandes quantités de mémoire vive et de disques durs à haute performance, ainsi que de logiciels de virtualisation pour maximiser l'utilisation des ressources.

3.5 Outils développement

Les outils de développement sont des logiciels utilisés par les développeurs pour créer, tester, déboguer et déployer des applications logicielles. Ces outils sont essentiels pour le développement de logiciels efficaces et de haute qualité, en faisant recours à une variété de langages de programmation et d'environnements de développement.

3.5.1 Choix des outils de développement

Le choix de la solution pour le développement d'une application dépend en grande partie des exigences et des contraintes spécifiques du projet. Par exemple, si l'application doit être déployée sur plusieurs plateformes, il peut être plus judicieux de choisir des langages de programmation et des outils de développement multiplateformes pour garantir la portabilité de l'application. De même, si l'application nécessite des fonctionnalités spécifiques qui ne sont pas disponibles dans le système d'exploitation choisi, il peut être nécessaire de considérer d'autres options.

Il est important de prendre en compte tous les facteurs pertinents lors de la sélection d'une solution pour le développement d'une application. Cela peut inclure l'expérience et les compétences de l'équipe de développement, les ressources disponibles, les contraintes de temps, les exigences de performances, la sécurité, la maintenance, la documentation et bien plus encore.

En effet, le choix des outils de développement est une étape cruciale dans la création d'une application. Il est important de considérer les facteurs tels que les exigences du projet, la portabilité, les fonctionnalités nécessaires, l'expérience de l'équipe de développement et les ressources disponibles.

Flutter

Flutter est un cadre de développement d'applications mobiles open-source créé par Google pour créer des applications hautes performances pour iOS et Android dans une seule base de code. Cet outil



FIGURE 3.12 – L'icône représentant le logo de Flutter.

offre aux développeurs un moyen rapide et expressif de créer des applications natives. Flutter est sorti en 2016 et écrit en utilisant le langage C, C++, Dart.(13)

Flutter est basé sur un moteur de rendu personnalisé qui permet une grande flexibilité et des performances élevées. Il offre également une riche bibliothèque de widgets pour la création d'interfaces utilisateur, ainsi que des outils de développement puissants tels que le hot reload pour un développement rapide et itératif. Avec Flutter, il est possible de créer des applications mobiles élégantes et performantes avec une seule base de code pour plusieurs plateformes, ce qui permet de réduire les coûts et les délais de développement.

Firebase



FIGURE 3.13 – *Firebase*

Firebase est une plateforme mobile de Google qui fournit des services back end pour les applications mobiles, telles que le stockage de données, l'authentification des utilisateurs, l'hébergement de fichiers, les notifications push et les analyses. Firebase peut être utilisé avec Flutter pour ajouter des fonctionnalités back end à une application mobile. Grâce à son intégration transparente avec Flutter, Firebase permet aux développeurs de se concentrer sur la création d'une expérience utilisateur de haute qualité sans avoir à se soucier de la gestion de l'infrastructure back end. Aider les développeurs à créer des applications hautement évolutives. Cette plateforme offre une base de données en temps réel, une authentification des utilisateurs, un hébergement de fichiers, une analyse d'utilisation, des notifications push, etc. Firebase est souvent utilisés avec Flutter car les deux sont développés par Google et sont donc étroitement intégrés, ce qui permet aux développeurs de créer des applications avec une connectivité cloud facilement et rapidement.[\(11\)](#)

Autres outils

React Native : Un Framework pour créer des applications mobiles natives en utilisant React - la bibliothèque JavaScript open-source développée par Facebook.[\(17\)](#)

Kotlin : Un langage de programmation créé par JetBrains pour fonctionner sur la machine virtuelle Java. C'est un langage à typage statique, concis, sûr

et interopérable avec Java.(14)

Electron : Un Framework pour créer des applications de bureau natives avec des technologies web comme JavaScript, HTML et CSS. Il utilise Chromium et Node.js et les package en applications de bureau.(9)

Java : Un langage de programmation orienté objet populaire, optimisé pour la portabilité et la scalabilité.(23)

Django : Un framework web Python de haut niveau qui encourage le développement rapide et un design pragmatique et clair.(8)

Back4App : Une plateforme Back end a le service (BaaS) pour alimenter les applications mobiles et les sites web. Elle offre de nombreuses fonctionnalités telles qu'une API REST, une authentification, une base de données, un hébergement que les développeurs peuvent utiliser pour construire leurs applications sans avoir à coder leur propre backend.(2)

Spring Boot : Un framework construit sur le framework Spring qui aide à créer des applications Spring autonomes et de qualité de production avec un minimum d'effort. Il vise à rendre le développement d'applications plus rapide avec une configuration automatique, des serveurs intégrés et des starters pour les technologies courantes.(25)

MongoDB : Une base de données de documents, initialement développée en 2007 dans le cadre de la pile MEAN. Elle utilise des documents au format JSON pour stocker les données. (21)

RethinkDB : Une base de données de documents open-source qui stocke des documents JSON et permet des mises à jour en temps réel.(26)

Meteor : Une plateforme JavaScript complète pour créer des applications web, mobiles et de bureau. Elle inclut un langage de requête de base de données, une bibliothèque d'interface utilisateur réactive et s'intègre parfaitement à de nombreuses bibliothèques tierces. (18)

Parse : Une plateforme open source BaaS (Backend-as-a-Service) axée sur la fourniture d'outils aux développeurs pour construire et exploiter des applications mobiles et web.(24)

3.6 Outil de modélisation : UML

La modélisation est une étape importante qui s'inscrit dans de nombreux processus d'ingénierie, son objectif est de mieux comprendre ou construire le système à réaliser. L'UML pour « **Unified Modeling Language** » est alors un langage unifié de modélisation objet, ce n'est pas une méthode, mais un ensemble d'outils permettant la modélisation des futures applications informatiques, bureau, mobile ou même web.

C'est un ensemble de notations graphiques s'appuyant sur des diagrammes et permettant de spécifier, visualiser et de documenter les systèmes logiciels orientés-objet. UML permet de décrire les différentes composantes d'un système logiciel, telles que les classes, les objets, les relations entre les classes, les cas d'utilisation, les activités, les séquences et les états. Les diagrammes UML sont des représentations graphiques de ces composants et permettent de visualiser les différentes interactions et les comportements du système.

Les diagrammes UML sont utilisés tout au long du cycle de vie d'un projet logiciel, depuis l'analyse des besoins jusqu'à la mise en œuvre du système. Ils permettent aux membres de l'équipe de développement de comprendre facilement les exigences du système, de concevoir et de planifier les fonctionnalités du système, et de documenter le code.

3.6.1 Types des diagrammes

Il existe deux grands types de diagrammes UML : les diagrammes de structure et les diagrammes de comportement (ces catégories sont elles-mêmes divisées en plusieurs sous-catégories). Ces variétés représentent les nombreux types de scénarios et de diagrammes que différents types de personnes utilisent.⁽¹⁹⁾

Les diagrammes de structure

Diagramme de classes : ce modèle, le plus courant dans le développement de logiciels, est utilisé pour décrire la conception logique et physique d'un système et pour présenter ses classes. Il ressemble à un organigramme vu que ces dernières sont représentées par des cases.

Diagramme de composants : également connu sous le nom de diagramme de flux des composants, il montre les groupes logiques d'éléments et leurs relations. En d'autres termes, il offre une vue plus simplifiée d'un système complexe en le désossant en plus petits composants. Chacun des éléments est représenté par une case rectangulaire comportant son nom. Les connecteurs définissent la relation/les dépendances entre les différents composants.

Diagramme de structure composite : il est rarement utilisé en dehors du développement de logiciels. En effet, bien qu'il soit similaire à un diagramme de classes, il va bien plus loin en décrivant la structure interne de plusieurs classes et en présentant leurs interactions. Pour les non-développeurs, une vue de haut niveau est probablement suffisante.

Diagramme de déploiement : il affiche les composants matériels (nœuds) et logiciels (artefacts) ainsi que leurs relations. Il apporte une représentation visuelle de l'endroit exact où chaque composant logiciel est déployé.

Diagramme des paquetages : il est utilisé pour représenter les dépendances entre les paquetages qui composent un modèle. L'objectif principal est de montrer la relation entre les différents grands composants qui forment un système complexe.

Les diagrammes de comportement

Diagramme d'activité : il s'agit d'un processus pas-à-pas comportant un début et une fin clairs. Ce diagramme comprend un ensemble d'activités qui doivent se produire pour atteindre un but. Il montre comment chaque activité mène à la suivante et la façon dont elles sont toutes liées.

Diagramme de cas d'utilisation : il décrit ce que fait un système, mais pas la manière dont il y parvient. Un cas d'utilisation correspond à un ensemble d'événements qui se produisent lorsqu'un acteur utilise un système pour terminer un processus. Un acteur est défini comme toute personne ou tout élément qui interagit avec le système (personne, organisation ou application) de l'extérieur du système.

Diagramme de temps : il est utilisé lorsque le temps est primordial, aussi connu sous le nom de diagramme de séquence ou d'événement, il ne montre pas comment les objets interagissent ou se modifient entre eux. Sur le plan fonctionnel, il montre comment les objets et les acteurs agissent dans une ligne du temps. L'accent est mis ici sur la durée des événements et sur les changements qui surviennent en fonction des contraintes de durée.

Diagramme états-transitions : aussi appelé « statechart », ce diagramme s'applique lorsque le comportement d'un objet est complexe et que le souci du détail est essentiel. Il permet de décrire le comportement d'un objet (ou parfois d'un opérateur) et la façon dont il change en fonction d'événements internes et externes.

Diagramme de communication : un diagramme de communication ou de collaboration est similaire à un diagramme de séquence, mettant toutefois l'accent sur la communication entre les objets. Il présente l'organisation des objets qui participent à une interaction et reprend des itérations ainsi que des ramifications plus complexes.

3.7 Conclusion

Ce chapitre nous a permis de survoler les différents types d'applications possibles, à savoir, mobile, web et bureau, dans l'objectif de choisir celle, voir celles qui sont plus appropriées dans le cas de notre problématique. Nous avons notamment parcouru quelques systèmes d'exploitation, car nous voulons répondre à l'exigence multiplateforme, afin de faire fonctionner notre prototype sur différents appareils et par conséquent augmenter le nombre de clients possibles.

Par la suite nous avons illustré quelques outils de développement tels que Flutter et Firebase, ainsi que l'outil de modélisation le plus répandu UML.

Dans le chapitre suivant, nous allons exposer une étape primordiale dans la réalisation de notre projet, il s'agit de la phase analyse et conception, qui consiste dans un premier temps à étudier les systèmes existants afin de pouvoir situer les besoins issus de notre projet et de procéder à la conception.

Chapitre 4

Analyse et conception

Contenu

4.1	Introduction	30
4.2	Analyse de l'existant	30
4.2.1	La description détaillée de la carte	30
4.2.2	L'historique de la carte	31
4.3	Spécification des besoins	32
4.3.1	Les besoins fonctionnels	32
4.3.2	Les besoins non fonctionnels	33
4.4	Spécification techniques de l'IHM d'ECU	34
4.4.1	Les spécifications Conceptuelles	34
4.4.2	Les spécifications syntaxiques	35
4.4.3	Les spécifications lexicales	37
4.5	Conclusion	38

4.1 Introduction

Pour assurer la réussite d'un projet, il est crucial de démarrer avec une phase d'analyse et de conception de qualité. Cela implique de formaliser les besoins des utilisateurs afin de développer une application qui leur correspond parfaitement. Cette approche contribue grandement au bon déroulement du projet. Lors de ce chapitre, nous allons voir les deux premières étapes d'un processus de génie logiciel, à savoir l'analyse des besoins et la conception, nous commençons d'abord par analyser et définir les besoins fonctionnels (les fonctionnalités spécifiques attendues) et non fonctionnels (les contraintes et critères de qualité) du système, nous verrons également la conception du point de vue IHM. Ensuite nous ferons appel à la modélisation UML, en présentant quelques diagrammes, tels que les diagrammes de cas d'utilisation, de séquence et de classes, afin d'avoir une vision globale de notre application.

4.2 Analyse de l'existant

L'analyse de l'existant est une phase essentielle, qui permet d'étudier les solutions informatiques existantes, elle doit être menée d'une façon claire et efficace pour parvenir à déterminer les besoins de la solution cible qui est notre projet de carte numérique universitaire.

4.2.1 La description détaillée de la carte

La carte étudiante en Algérie est un document officiel qui est délivré par le ministère de l'Enseignement Supérieur et de la Recherche Scientifique aux étudiants inscrits dans des établissements d'enseignement supérieur publics ou privés algériens. La carte étudiante algérienne comporte généralement les

informations suivantes :

- Le nom complet de l'étudiant
- La date de naissance
- La photo de l'étudiant
- Le numéro d'identification national de l'étudiant (NIN)
- Le nom de l'établissement dans lequel l'étudiant est inscrit
- La filière d'études de l'étudiant
- La date d'expiration de la carte

Il est important de noter que la carte étudiante en Algérie est souvent nécessaire pour accéder aux salles de cours, aux bibliothèques, et à d'autres installations universitaires. Elle est sur tous indispensable durant les examens. Les étudiants doivent donc la conserver sur eux en tout temps lorsqu'ils se trouvent sur le campus universitaire. Elle peut également être utilisée comme preuve de statut d'étudiant pour bénéficier de réductions sur les transports en commun, les événements culturels et autres avantages liés à la vie étudiante.

4.2.2 L'historique de la carte

L'origine de la carte étudiante en Algérie remonte à l'époque postindépendance en 1962, lorsque le pays a commencé à mettre en place son propre système d'enseignement supérieur qui a connu un développement au fil des années. Le ministère de l'Enseignement supérieur et de la Recherche Scientifique (MESRS) a émis des cartes d'étudiant algériennes pour les étudiants inscrits dans les universités algériennes, qui a à son tour évolué pour devenir plus moderne, sécurisée et multifonctionnelle. Cette carte est devenue un

outil essentiel pour les étudiants, leur permettant d'accéder à différents services universitaires et même à des services publics en dehors du campus. Les méthodes utilisées pour collecter et traiter les données sur les cartes des étudiantes en Algérie varient en fonction des périodes et des évolutions de la carte elle-même. Au départ (1960), les données étaient principalement collectées à travers des formulaires papier que les étudiants remplissaient à la main. Ces formulaires comprenaient des informations de base telles que le nom, l'adresse, le niveau d'étude, la filière, etc. Les méthodes de collecte de données ont également évolué, ainsi, dans les années 1990, les cartes étudiantes étaient devenues des cartes à code-barres, ce qui permettait de stocker des informations sur les étudiants directement sur la carte elle-même. Les données collectées incluaient alors des informations telles que l'identité de l'étudiant, son inscription dans une filière, la date d'obtention du diplôme, etc. Aujourd'hui, les données sur les cartes des étudiantes en Algérie sont collectées et traitées de manière électronique à l'aide de systèmes informatisés. Les informations collectées peuvent inclure les mêmes informations de base que par le passé, ainsi que des données plus avancées telles que des informations biométriques. (1)

4.3 Spécification des besoins

On trouve deux types de besoin : les besoins fonctionnels et les besoins non fonctionnels.

4.3.1 Les besoins fonctionnels

Il s'agit d'un ensemble de fonctionnalité et d'actions spécifiques que le système doit être capable d'accomplir pour répondre aux attentes d'un

acteur, la table 4.1 suivante représente un résumé des besoins fonctionnels de notre application.

TABLE 4.1 – *Spécification des besoins fonctionnels de notre projet.*

Acteur	Besoins
Utilisateur	<ol style="list-style-type: none"> 1. S’inscrire. 2. S’authentifier. 3. Consulter la page d’accueil. 4. Consulter les actualités et les évènements. 5. Consulter la carte de campus universitaire. 6. Rechercher une salle. 7. Spécifier son position. 8. Consulter le menu latéral. 9. Modifier des paramètres. 10. Consulter des notifications. 11. Consulter le profil personnel.

4.3.2 Les besoins non fonctionnels

Il s’agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Les besoins non fonctionnels de notre application se résument à :

- ▷ **Architecture** : client/serveur.
- ▷ **Interopérabilité** : Notre E-carte universitaire est compatible avec différentes plates-formes (Systèmes d’exploitation) : Android, iOS, Windows, Linux, MacOS.
- ▷ **Sécurité** : étant donné que le système gère des données confidentielles, comme les informations personnelles des étudiants, la sécurité est pri-

mordiale. L'authentification et la gestion des accès doivent être rigoureuses pour éviter toute violation de la confidentialité des données.

- ▷ **Performance** : La fluidité du logiciel et la minimisation du temps de réponse sont essentielles, les utilisateurs doivent pouvoir accéder rapidement aux différents services liés à leur carte universitaire, notre application a comme objectif d'offrir une expérience utilisateur efficace et rapide.
- ▷ **Maintenance et évolutivité** : Nous tenons à maintenir notre E-carte universitaire à jour, grâce à son architecture extensible nous pouvons ajouter de nouvelles fonctionnalités à l'avenir.

4.4 Spécification techniques de l'IHM d'ECU

L'interface homme Machine(IHM) de notre application est primordiale, elle doit être ergonomique, afin de faciliter son utilisation et son apprentissage, pour notamment élargir son spectre d'utilisation, pour cela nous avons procédé aux spécifications techniques tout en appliquant les règles ergonomiques de base.

4.4.1 Les spécifications Conceptuelles

Les spécifications conceptuelles représentent la première étape dans le processus de développement d'une IHM. Elles visent à décrire les objets manipulés par l'utilisateur, leurs attributs, propriétés, relations, et les opérations exécutables sur ces objets. (La table 4.2) présenté ci-dessous est un extrait des spécifications conceptuelles de notre application :

TABLE 4.2 – *Extrait de spécifications conceptuelles : présentation de l'objet menu principale.*

Objets	Menu principal latéral
Attributs et propriétés	<ul style="list-style-type: none"> ▷ Groupe : Organisation <ul style="list-style-type: none"> • Item : Liste des tâches • Item : Mes fichiers ▷ Groupe : Générale <ul style="list-style-type: none"> • Item : Paramètres • Item : À propos de nous • Item : Déconnexion
Relations	Le menu latéral utilise des icônes représentant chaque option principale, accompagnées de libellés textuels descriptifs. Les icônes sont placées à gauche des libellés.
Opérations exécutables	Accès aux items via un simple clic (via souris ou tactile) Afficher ou cacher le menu en glissant ce dernier.

4.4.2 Les spécifications syntaxiques

Elle concerne le dialogue utilisateur-machine, il faudrait prendre en compte des normes de qualité syntaxique lors du choix du type d'IHM. Notre application multiplateforme est dédiée à différents acteurs du secteur universitaire, cela signifierait que les menus, boutons et autres éléments d'interface

doivent être conçus de manière à être facilement identifiables et compréhensibles pour les utilisatrices de tous les âges et niveaux d'expérience. Les textes et les instructions doivent être clairs et concis, sans utiliser de jargon ou de termes techniques difficiles à comprendre par les utilisateurs universitaires. Les champs de saisie doivent être conçus de manière à minimiser les erreurs de saisie et à aider l'utilisateur à entrer les données correctement. Par exemple, les formats de date et d'heure doivent être clairement indiqués, les options de sélection doivent être présentées de manière claire et ordonnée, et les messages d'erreur doivent être précis et indiquer comment résoudre le problème.

- ▷ **Choix du style de dialogue** : il est important de choisir le style de dialogue approprié pour chaque opération afin d'optimiser l'expérience utilisateur. Le menu latéral, les boutons et les champs de saisie sont des éléments d'interface courants et conviviaux pour les opérations simples et fréquentes. Les formulaires détaillés, en revanche, sont utilisés à titre de besoin comme au niveau de l'inscription et ils doivent être bien structurés.
- ▷ **Règles ergonomiques générales** : nous nous sommes basés sur les critères de base de Bastien et Scapin (5) que nous avons la chance de voir de notre cursus, voici quelques règles que nous avons prises en considération :
 - Utiliser des icônes et des images pour améliorer la compréhension de l'utilisateur et rendre l'interface plus intuitive (signifiants des codes et dénomination).
 - Placer les éléments les plus importants (les plus fréquents) en haut de la page ou dans des endroits facilement accessibles pour l'utilisateur (guidage), et en contrepartie ne pas mettre en pers-

pective ceux qui sont risqués (gestion des erreurs).

- Éviter les éléments clignotants ou animés qui pourraient distraire ou perturber l'utilisateur (guidage).
- Proposer des messages d'erreur clairs et concis en cas de saisie incorrecte ou d'opération impossible (gestion des erreurs et feedback immédiat).
- Adapter l'interface en fonction du type d'appareils utilisé pour garantir une expérience homogène sur toutes les plateformes (adaptabilité).
- Offrir des options de personnalisation pour permettre à l'utilisateur de modifier l'interface en fonction de ses préférences et de ses besoins (adaptabilité).

4.4.3 Les spécifications lexicales

Les spécifications lexicales définissent la représentation graphique et la présentation externe de l'interface utilisateur. Voilà ce que nous avons considéré lors de notre conception.

- **Dénomination** : choisir des noms clairs et concis pour les éléments de l'interface utilisateur.
- **Codage** : utiliser un encodage approprié pour les caractères spéciaux et les symboles.
- **Guide de couleurs** : créer une hiérarchie visuelle en utilisant des couleurs pour faciliter la compréhension.
- **Typographie** : choisir une police et une taille de police adaptées pour les différents éléments de l'interface utilisateur afin d'améliorer la lisibilité et la compréhension.

- **Mnémoniques** : utiliser des symboles, des icônes ou des images pour représenter les fonctions de l'interface utilisateur et faciliter leur reconnaissance et leur utilisation.

4.5 Conclusion

Dans n'importe quel projet informatique, il faudrait passer par une démarche génie logiciel, afin de cerner la conception formelle et technique du projet. Par conséquent, nous avons commencé d'abord dans ce chapitre, par l'étape de spécifications des besoins fonctionnels et non fonctionnels, nous avons par le site procédé à la présentation des différentes spécifications techniques de l'IHM de notre E-carte universitaire, un aspect important et primordial lors de la réalisation d'application à un usage destiné à des acteurs avec différents profile (étudiant, enseignant, administrateur, agent (de sécurité ou autres), agent transactionnel). Nous avons également identifié les principaux objectifs à prendre en compte. Dans le chapitre suivant nous allons présenter notre application de E-carte universitaire.

Chapitre 5

Présentation du prototype

Contenu

5.1	Introduction	40
5.2	Architecteur de l'application	40
5.3	Présentation de l'application	41
5.3.1	Fenêtre « Authentification »	41
5.3.2	Fenêtre « S'inscrire »	43
5.3.3	Fenêtre « Page d'accueil »	43
5.3.4	Fenêtre « Plan de campus universitaire »	45
5.3.5	Fenêtre « Profile »	46
5.3.6	Fenêtre « La liste des tâches »	47
5.3.7	Fenêtre « E-présence »	48
5.3.8	Fenetre « E-Paiement »	51
5.4	Conclusion	54

5.1 Introduction

Dans ce chapitre, nous allons voir notre application de E-Carte Universitaire de plus près, ou nous allons présenter en détails les différents sur les composants et les interactions entre eux. Nous donnerons également un aperçu détaillé des différentes fenêtres de l'application, mettant en évidence leurs fonctionnalités et leur rôle dans l'ensemble du système.

5.2 Architecteur de l'application

Nous avons procédé à une séparation le front-end du back-end, afin de permettre une évolution de notre application, une flexibilité technologique, et une meilleur collaboration entre notre les membres de notre équipe.

Le front-end : est la partie de l'application qui est visible pour l'utilisateur. Il comprend toutes les fonctionnalités et tâches que l'utilisateur peut exécuter, comme la gestion des tâches ou des activités. L'interface utilisateur et les exigences d'authentification sont également incluses pour identifier et sécuriser l'accès des utilisateurs.

Le back-end : représente les opérations qui se déroulent en arrière-plan de l'application et qui ne sont pas directement visibles ou accessibles à l'utilisateur. La gestion des données et les événements qui se produisent, tels que le stockage des informations dans la base de données et l'interaction avec le serveur connecté à l'application, font partie de cela.

Chaque action effectuée dans notre application nécessite une authentification et un accès à la base de données afin de garantir la sécurité et la confidentialité des utilisateurs.

5.3 Présentation de l'application

Notre application prend en charge 4 principales fonctionnalités, elle peut être utilisée comme un guide dans le campus universitaire, elle permet de gérer la présence des étudiants, et de procéder à des paiements électroniques des services dans le campus universitaire, ainsi que de planifier des tâches à faire. Pour arriver à nos fins nous avons développé Huit fenêtres dans notre application, à savoir :

- ▷ Fenêtre "Authentification".
- ▷ Fenêtre "S'inscrire".
- ▷ Fenêtre "Page d'accueil".
- ▷ Fenêtre "Plan de campus universitaire".
- ▷ Fenêtre "Profile".
- ▷ Fenêtre "E-présence".
- ▷ Fenêtre "E-Paiement".
- ▷ Fenêtre "La liste des tâches".

5.3.1 Fenêtre « Authentification »

Lorsque vous ouvrez l'application, vous verrez une fenêtre où vous devriez vous connecter. Vous pouvez choisir la langue (Anglais, Français ou Arabe) et le thème de l'interface (clair ou sombre) voir figure 5.1. Pour vous identifier, vous devriez entrer votre adresse e-mail de l'université (avec le nom du domaine de l'université) qui n'est délivrée qu'aux étudiants, aux enseignants ainsi que le personnel universitaire exclusivement, ceci est le premier niveau de sécurité, vous devriez également introduire votre mot de passe.

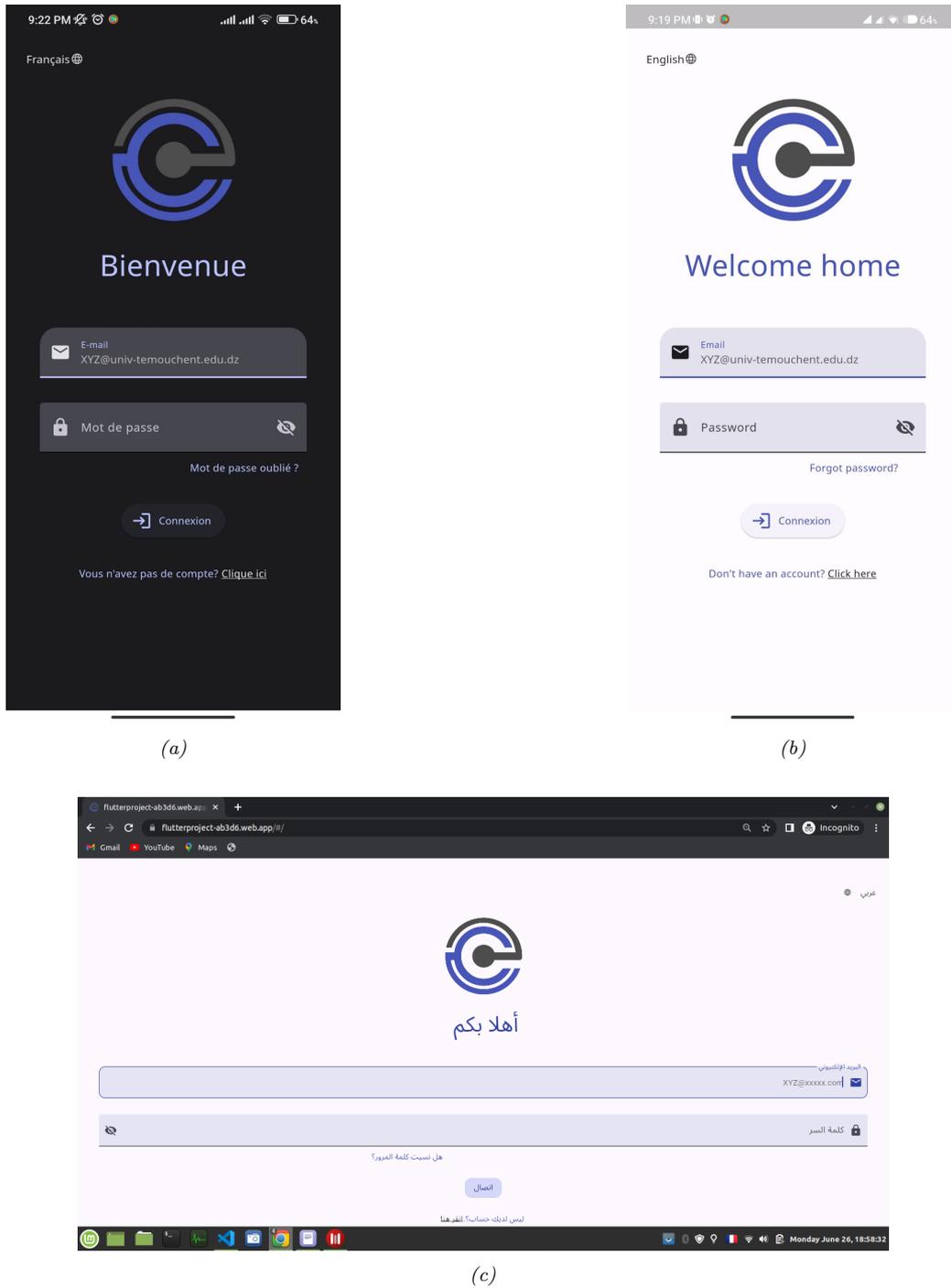


FIGURE 5.1 – Fenêtre "Authentification" : (a) Mode sombre version mobile en français, (b) Mode clair version mobile en anglais, (c) version bureau en arabe.

5.3.2 Fenêtre « S'inscrire »

Lorsque l'utilisateur clique sur le bouton "Cliquez ici" (voir figure 5.1) dans la fenêtre "Authentifier" pour créer un compte, l'application affiche une fenêtre dans laquelle l'utilisateur peut remplir ses informations (voir figure 5.2).

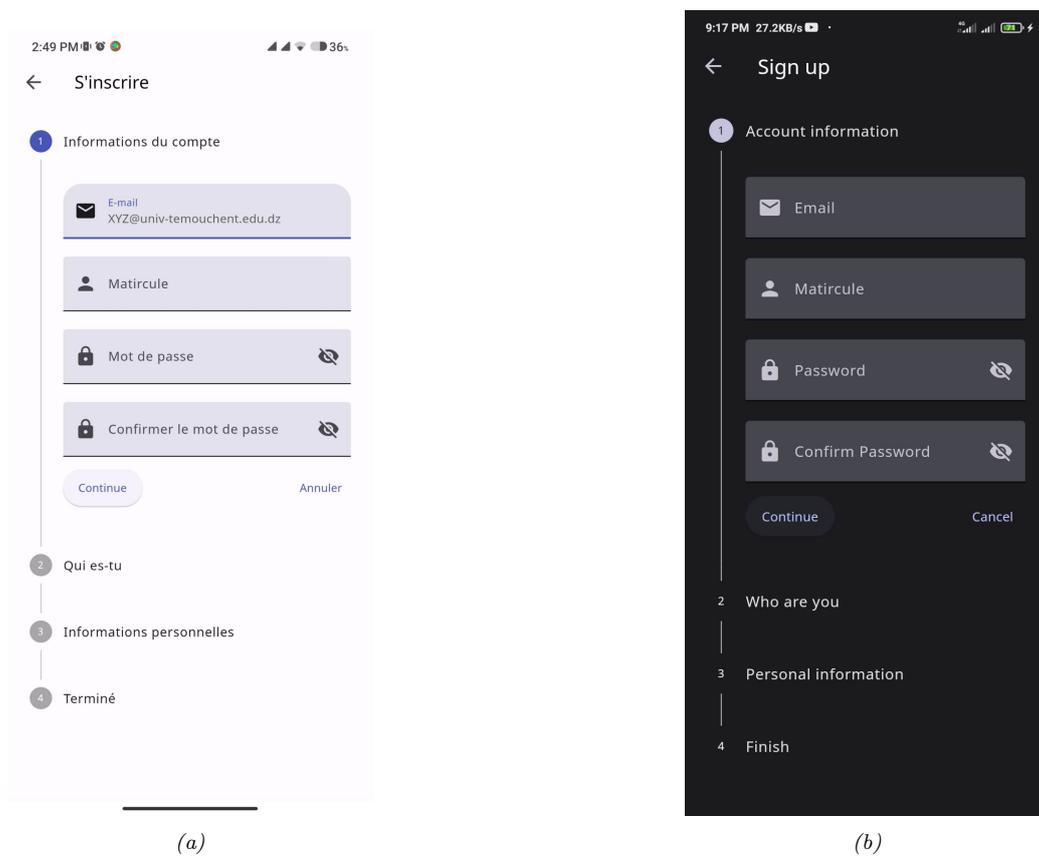


FIGURE 5.2 – Fenêtre d'inscription version mobile : (a) en français en mode clair, (b) en anglais en mode sombre.

5.3.3 Fenêtre « Page d'accueil »

Afin de garantir la sécurité, un e-mail de confirmation est envoyé pour activer le compte, c'est le deuxième niveau de protection. La vérification du

rôle de l'utilisateur est effectuée dans la base de données. Une fois le compte activé, l'utilisateur peut s'authentifier et accéder à la page d'accueil. Cette page d'accueil est dotée d'un menu latéral qui propose des services et des solutions à notre problématique. La figure 5.3 suivante présente un aperçu de la page d'accueil.



(a)

FIGURE 5.3 – Fenêtre d'accueil

5.3.4 Fenêtre «Plan de campus universitaire »

La fonctionnalité de géolocalisation utilise le plan du campus universitaire pour détecter automatiquement votre emplacement. Ensuite, vous pouvez spécifier votre destination à l'intérieur du campus, telle que les salles de cours, les laboratoires ou les locaux administratifs, par exemple. Cela vous permettra de trouver facilement votre chemin vers votre destination précise sur le campus, et vous évitera de perdre du temps ou même d'être frustré. Cette fenêtre est illustrée par la figure 5.4 présentée ci-dessous.

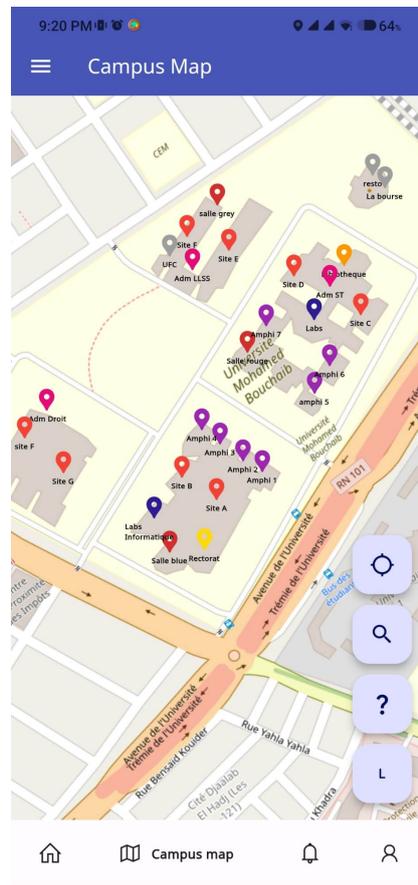


FIGURE 5.4 – Fenêtre du plan du campus universitaire (géolocalisation intra-campus).

5.3.5 Fenêtre « Profile »

La fenêtre du profil d'utilisateur illustré dans la figure 5.5 affiche les informations personnelles de l'utilisateur. Cela peut inclure des détails tels que le nom, l'adresse e-mail, le numéro de téléphone, la photo de profil, la date de naissance, etc. De plus, chaque utilisateur possède un code QR unique qui peut être affiché sur cette page.



FIGURE 5.5 – Fenêtre du profile affichant les informations personnelles.

5.3.6 Fenêtre « La liste des tâches »

Dans la liste des tâches (To-Do List), vous pouvez créer une liste en y ajoutant les différentes tâches que vous souhaitez accomplir. Pour chaque tâche, vous avez la possibilité de planifier une date de rappel, ce qui vous aidera à vous souvenir de la tâche à effectuer à une date spécifique. Vous pouvez également supprimer une tâche de la liste si vous l'avez terminée ou si vous ne souhaitez plus la suivre. Cela vous permet de garder une trace de vos tâches à faire et de les organiser selon vos préférences (voir figure 5.6).

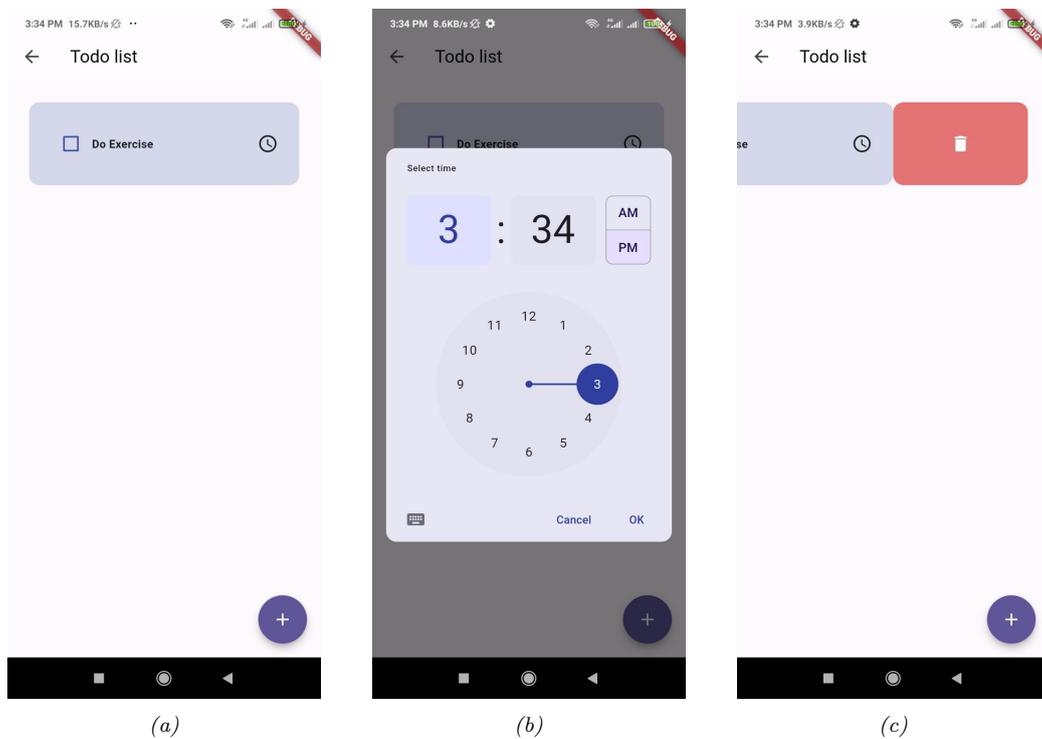


FIGURE 5.6 – Fenêtre de planification des tâches ,version en anglais *ToDo list* : (a) Liste des tâches, (b) planification de la tâche, (c)Moyen de suppression d'une tâche.

5.3.7 Fenêtre « E-présence »

La gestion de la présence comporte deux aspects, du côté de l'enseignant et du côté de l'étudiant.

a) Du côté de l'enseignant L'enseignant doit spécifier les matières qu'il enseigne et fournir certaines informations supplémentaires. Ensuite, il doit sélectionner la matière pour laquelle il souhaite marquer la présence (voir figure 5.7).

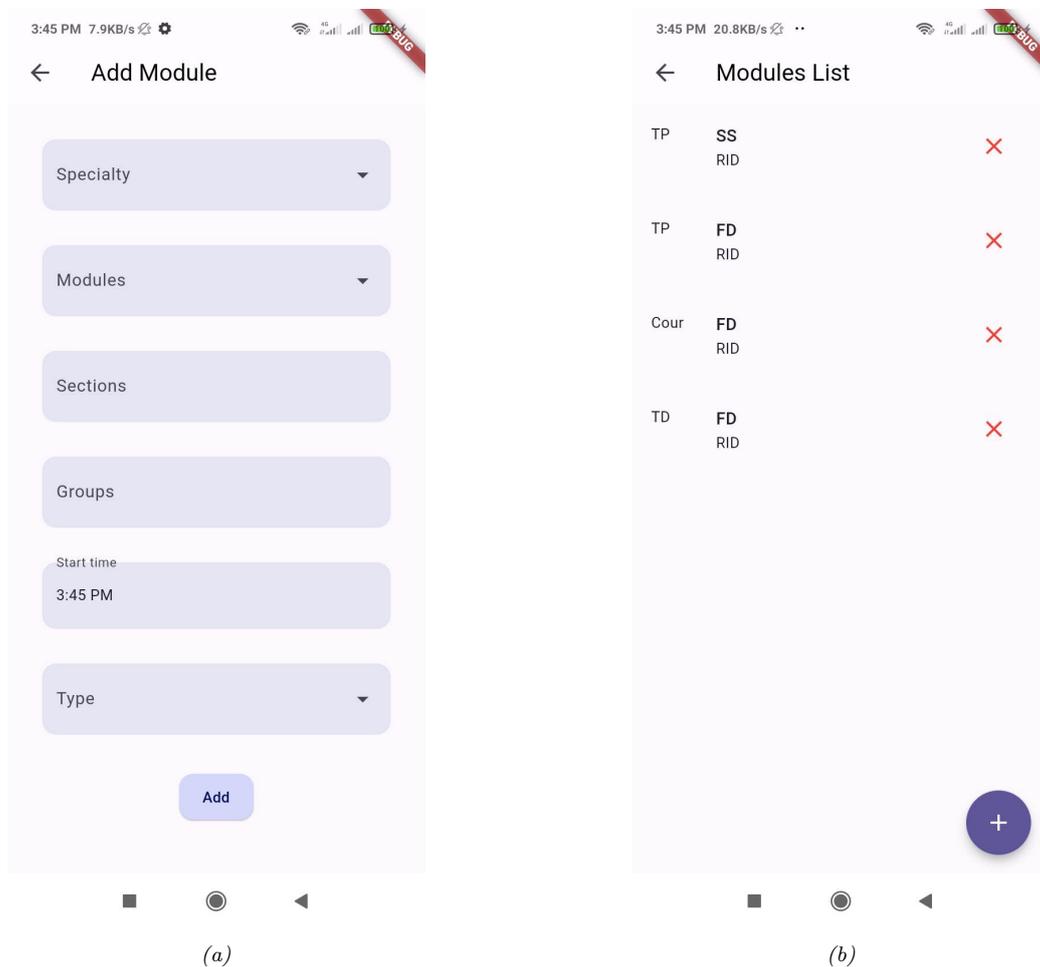


FIGURE 5.7 – Gestion des modules : (a) Ajout d'un module, (b) la liste des modules.

Il a alors trois méthodes possibles pour réaliser cette tâche, l'enseignant choisit celle qui convient. Ces méthodes sont les suivantes :

1. **Partager un code QR via un projecteur (data show)** : l'enseignant peut afficher un code QR sur l'écran de projection, que les étudiants pourront ensuite scanner à l'aide de leur appareil mobile pour indiquer leur présence (voir figure 5.8)

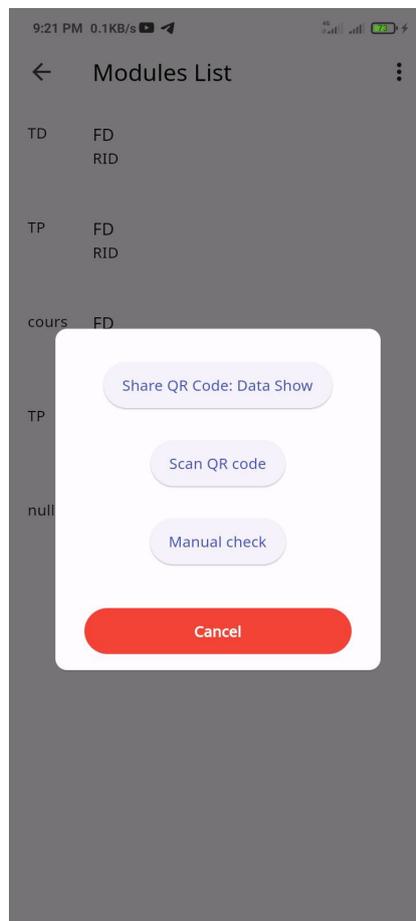
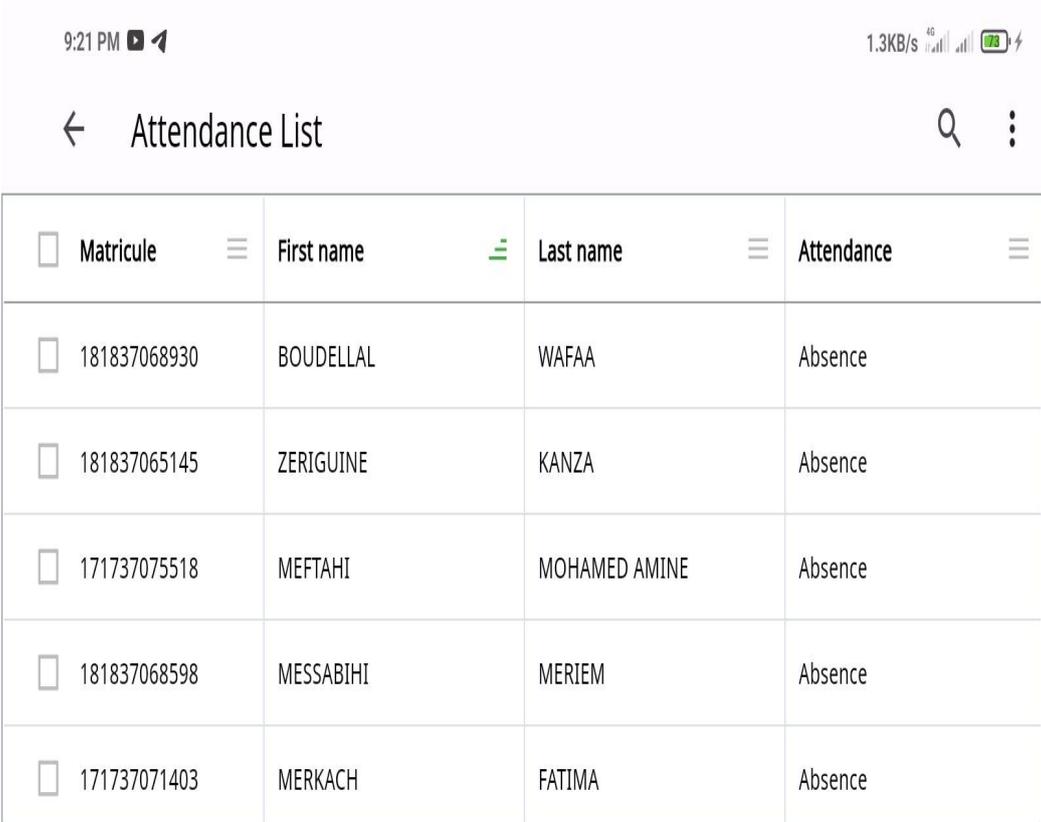


FIGURE 5.8 – *Choix de la méthode de gestion de la présence des étudiants.*

2. **Scanner le code QR de l'étudiant** : l'enseignant peut utiliser son propre appareil pour scanner le code QR affiché sur l'appareil mobile de l'étudiant, confirmant ainsi sa présence.
3. **Marquer la présence manuellement** : l'enseignant peut également choisir de marquer manuellement la présence des étudiants en cochant leur nom sur une liste(voir figure 5.9).



<input type="checkbox"/> Matricule	First name	Last name	Attendance
<input type="checkbox"/> 181837068930	BOUDELLAL	WAFAA	Absence
<input type="checkbox"/> 181837065145	ZERIGUINE	KANZA	Absence
<input type="checkbox"/> 171737075518	MEFTAHI	MOHAMED AMINE	Absence
<input type="checkbox"/> 181837068598	MESSABIHI	MERIEEM	Absence
<input type="checkbox"/> 171737071403	MERKACH	FATIMA	Absence

FIGURE 5.9 – Liste des étudiants(méthode manuelle).

Ces options offrent à l'enseignante différente possibilité pour enregistrer la présence des étudiants de manière pratique et efficace, ceci est illustré dans la figure 5.8.

b) Du côté de étudiant Une fois que l'étudiant appuie sur le bouton "E-présence", le scanner s'allume. L'étudiant peut alors utiliser son smart-

phone pour scanner le code QR partagé par l'enseignant. Cela permet à l'étudiant de signaler sa présence de manière rapide et pratique. Cependant (voir figure 5.10), si l'étudiant ne dispose pas d'un smartphone ou n'est pas en mesure de scanner le code QR, la méthode manuelle est utilisée par l'enseignant.

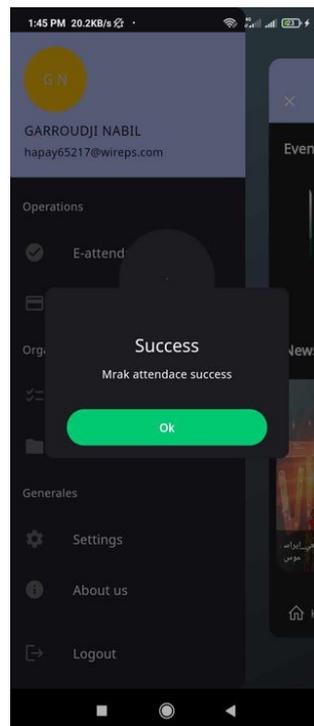


FIGURE 5.10 – La fenêtre indique que la présence est marquée.

5.3.8 Fenetre « E-Paiement »

Dans le système de paiement électronique, les utilisateurs (étudiants, enseignants ou personnel de l'université) ont la possibilité de :

1. Recharger leurs comptes :

- l'utilisateur ne possède aucune de ces deux cartes, il peut utiliser l'option "Recevoir crédits"

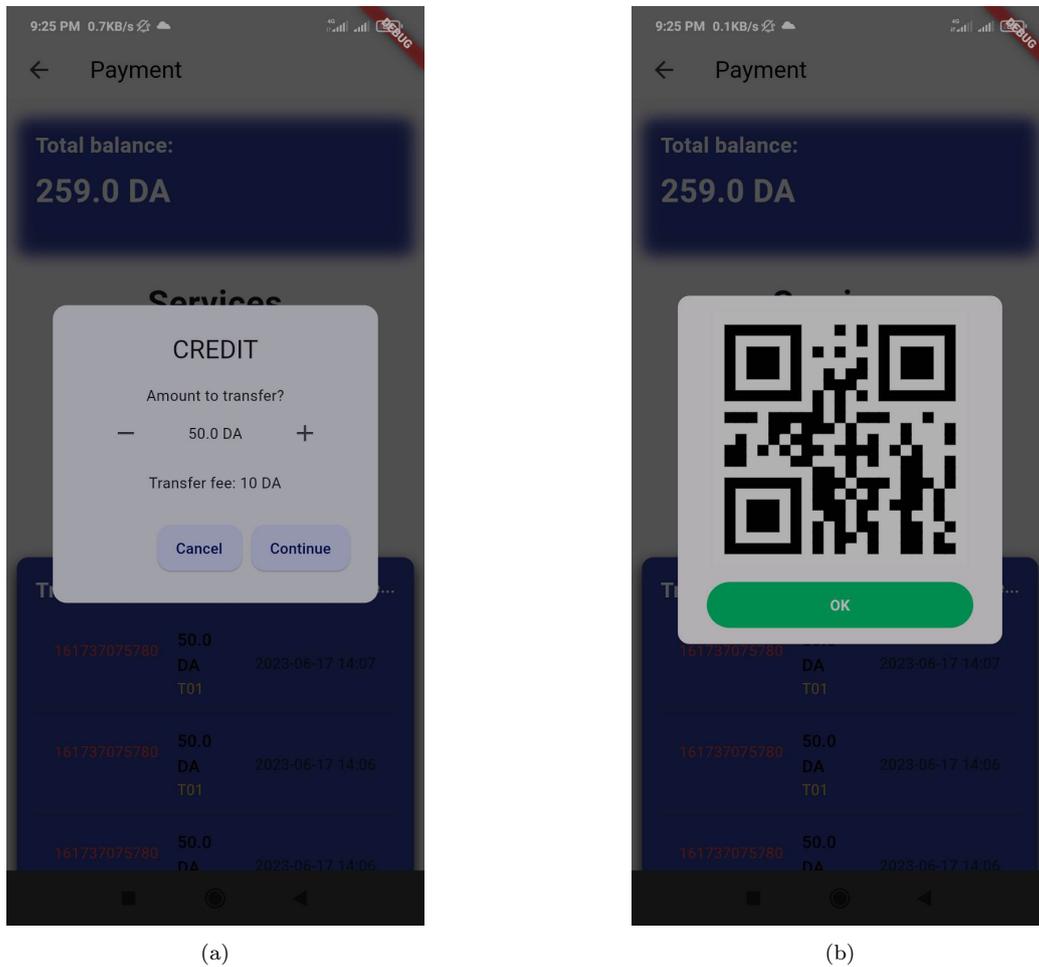


FIGURE 5.11 – Envoi de crédits : (a) choix du montant, (b) utilisation du code QR pour recevoir les crédits.

2. Transférer des crédits :

L'utilisateur peut utiliser l'option "Envoyer crédits" pour transférer des crédits de son compte vers le compte d'une autre personne. Pour ce faire, il doit fixer le montant qu'il souhaite envoyer (voir figure 5.11), puis scanner le code QR du destinataire.

3. Acheter des services :

Lorsqu'ils souhaitent acheter un service universitaire, que ce soit au niveau des restaurants, des cafétérias, de l'hébergement, du transport, etc., les utilisateurs ont la possibilité d'utiliser

le bouton "Acheter" pour effectuer la transaction correspondante. Pour cela, l'utilisateur doit entrer le montant requis, sélectionner le service souhaité, puis partager leur code QR (voir figure 5.12)

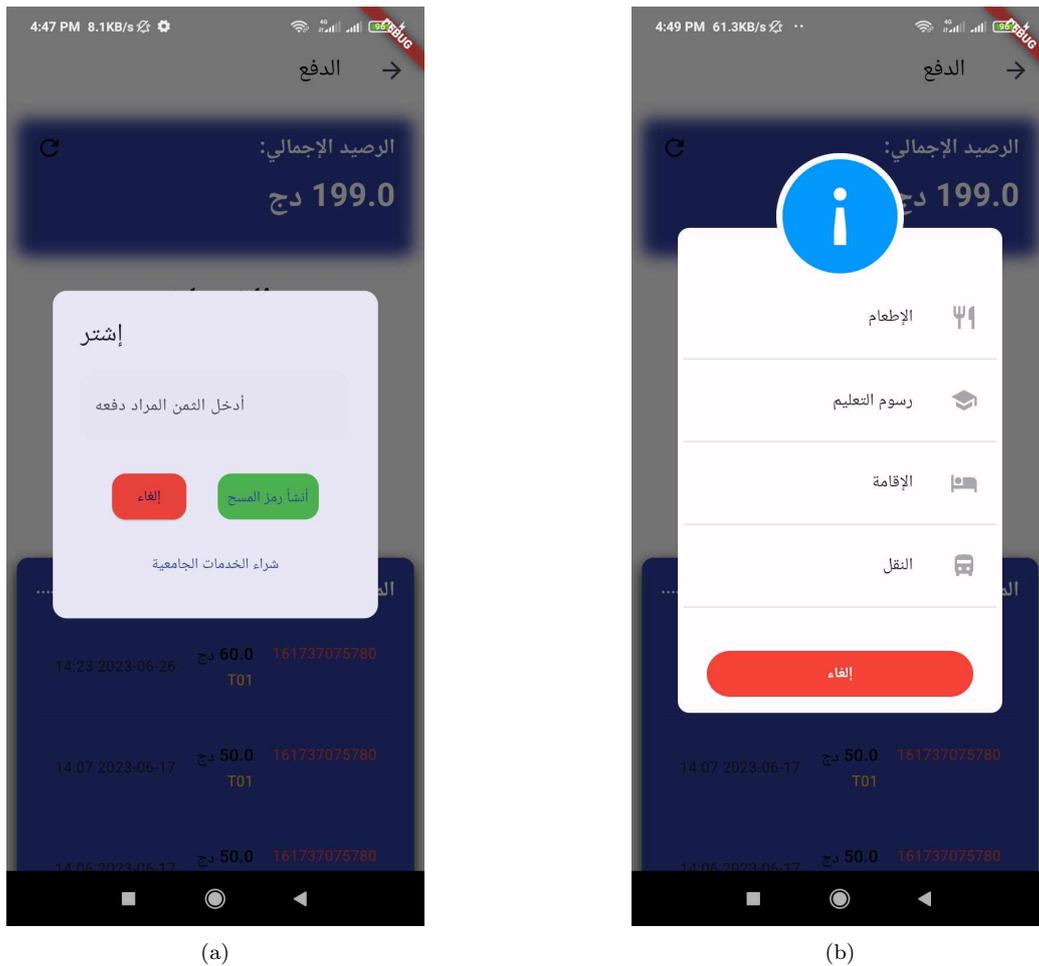


FIGURE 5.12 – Achat de services : (a) saisir le montant d'achat, (b) choix du service.

4. **Consulter l'historique des transactions** : Les utilisateurs ont la possibilité de voir l'historique de toutes les transactions effectuées, ce qui leur permet de suivre leurs dépenses et leurs recharges précédentes. Ces fonctionnalités offrent aux utilisateurs une expérience pratique et sécurisée pour effectuer des transactions électroniques, recharger leur

compte, transférer du crédit et acheter des services selon leurs besoins (voir figure 5.13).



FIGURE 5.13 – Fenêtre d’affichage des transactions.

5.4 Conclusion

La phase de réalisation qui inclut l’implémentation et le développement est la dernière étape dans le cycle de vie d’une application. Dans ce dernier

chapitre, nous avons présenté l'architecture globale de notre prototype, qui est une application E-Carte Universitaire. Nous avons également illustré ses principales fenêtres. L'architecture globale permet de définir la structure de l'application, ses composants et leurs interactions. Les fenêtres sont les interfaces graphiques qui permettent aux utilisateurs d'interagir avec l'application, tous au long de la réalisation de notre projet nous étions à l'écoute de quelques acteurs, nous avons essayé de faire une évaluation au fur et à mesure du développement, afin de combler les lacunes, cependant d'autres tests utilisateurs seraient bénéfiques pour l'amélioration de notre prototype.

Chapitre 6

Conclusion générale

La numérisation offre de nombreux avantages dans différents domaines, tels que l'efficacité, l'économie d'espace, l'accessibilité, la conservation des données, la recherche avancée et la durabilité environnementale. Sur un campus universitaire, la numérisation peut résoudre des problèmes tels que la gestion de l'argent liquide, l'affluence et la perte de temps, les difficultés pédagogiques et le manque d'orientation pour les nouveaux étudiants ainsi qu'au visiteur. Avec l'ère de numérisation et grâce à l'internet et les technologies de réseaux sans fil tel que LTE(4G), ces problèmes peuvent être efficacement résolus en proposant une solution adéquate.

Notre application favorise une meilleure préservation des données en éliminant les risques de perte, de détérioration et de difficulté de récupération, liés aux documents papiers. Elle offre une fonctionnalité de géolocalisation permettant aux utilisateurs de trouver les lieux de destination plus rapidement, évitant ainsi les pertes de temps lors de la recherche d'une localisation particulière (salle, administration, etc.). De plus, elle améliore l'expérience des étudiants, les enseignants et le personnel universitaire, en leur offrant un outil pratique et intuitif pour naviguer dans les services universitaires, assu-

rant ainsi une expérience fluide et personnalisée. En intégrant ces solutions, l'université peut améliorer l'accessibilité et l'efficacité de ces services, simplifier les processus pédagogiques tels que la gestion numérique de la présence avec diverses options, tout en considérant les différentes contraintes, offrir une meilleure expérience aux utilisateurs, et résoudre les problèmes liés aux paiements, aux documents papier et à l'affluence des services.

Dans le cadre de ce projet, nous avons suivi plusieurs étapes, en commençant par le contexte et les objectifs afin de bien cerner notre projet, puis nous avons donné un aperçu sur les systèmes d'exploitation les plus utilisés actuellement et plus particulièrement dans le domaine mobile, nous avons exposé ensuite quelques technologies surtout celle qui nous avons utilisé afin de créer notre applications multiplateforme. Dans une démarche génie logiciel, nous avons procédé à l'analyse et Et puis finalement, nous avons présenté un prototype complet de l'application.

Pour réaliser ce projet, nous avons acquis des connaissances dans divers domaines techniques, juridiques, marketing et financiers. Sur le plan technique, Nous nous sommes également familiarisés avec des aspects juridiques tels que les réglementations commerciales, les impôts et la propriété intellectuelle et les aspects de marketing et financiers telque le BMC.

Dans notre perspective future, nous prévoyons d'être à jour en matière de technologie, d'assurer une gestion efficace, de favoriser l'innovation et d'être à l'écoute de nos clients potentiels. Nous souhaitons exploiter les réseaux sociaux pour augmenter la portée et l'engagement de notre application, élargir et diversifier nos partenariats commerciaux, améliorer l'efficacité de l'application et proposer de nouvelles fonctionnalités et services. Nos objectifs ultimes sont d'assurer une évolution exponentielle de notre l'application, d'accroître les revenus, la rentabilité et la valeur globale de l'entreprise,

d'investir dans les infrastructures et les ressources humaines pour soutenir l'expansion du marché, et de maintenir une croissance et un succès à long terme.

Bibliographie

- [1] Historique de l'enseignement supérieur en Algérie. <http://www.mesrs.dz/fr/historique-de-lenseignement-superieur-en-algerie>. Consulté le 29/06/2023.
- [2] Anthropic Ltd. Back4app. <https://www.back4app.com>. Consulté le 29/06/2023.
- [3] Apple Inc. Apple. <https://www.apple.com/>. Consulté le 29/06/2023.
- [4] Apple Inc. Apple macOS Ventura. <https://www.apple.com/macOS/ventura/>. Consulté le 29/06/2023.
- [5] J. C. Bastien and D. L. Scapin. Evaluating a user interface with ergonomic criteria. *International Journal of Human-Computer Interaction*, 7(2) :105–121, 1995.
- [6] BlackBerry Ltd. BlackBerry. <https://www.blackberry.com/>. Consulté le 29/06/2023.
- [7] J. Clark. *Tapworthy : Designing Great iPhone Apps*. O'Reilly Media, 2010.

-
- [8] Django Software Foundation. Django. <https://www.djangoproject.com/>. Consulté le 29/06/2023.
- [9] GitHub, Inc. Electron. <https://electronjs.org/>. Consulté le 29/06/2023.
- [10] Google LLC. Android. <https://www.android.com>. Consulté le 29/06/2023.
- [11] Google LLC. Firebase. <https://firebase.google.com/>. Consulté le 29/06/2023.
- [12] Google LLC. Google Chrome OS. <https://www.google.com/chromebook/chrome-os/>. Consulté le 29/06/2023.
- [13] Google LLC. Flutter : Beautiful native apps in record time. <https://flutter.dev>, 2016. Consulté le 29/06/2023.
- [14] JetBrains s.r.o. The kotlin programming language. <https://kotlinlang.org/>. Consulté le 29/06/2023.
- [15] J. F. Kurose and K. W. Ross. *Computer Networking : A Top-Down Approach*. Pearson, 2017.
- [16] Linux Community. Linux. <http://www.linux.org>. Consulté le 29/06/2023.
- [17] Meta Platforms, Inc. React native. <https://reactnative.dev/>. Consulté le 29/06/2023.
- [18] Meteor Development Group. Meteor. <https://www.meteor.com/>. Consulté le 29/06/2023.

-
- [19] Microsoft Corporation. Guide de modélisation uml et de modélisation de bases de données. <https://www.microsoft.com/fr-fr/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>.
- [20] Microsoft Corporation. Microsoft Windows. <https://www.microsoft.com/fr-fr/windows/>. Consulté le 29/06/2023.
- [21] MongoDB Inc. Mongoddb. <https://www.mongodb.com/>. Consulté le 29/06/2023.
- [22] J. P. Mueller. *Beginning Programming with Python For Dummies*. John Wiley & Sons, 2019.
- [23] Oracle Corporation. The java language environment. <https://www.java.com/en/>. Consulté le 29/06/2023.
- [24] Parse Community. Parse. <https://parseplatform.org/>. Consulté le 29/06/2023.
- [25] Pivotal Software. Spring boot. <https://spring.io/projects/spring-boot>. Consulté le 29/06/2023.
- [26] RethinkDB Community. Rethinkdb. <https://www.rethinkdb.com/>. Consulté le 29/06/2023.
- [27] L. Shklar and R. Rosen. *Web Application Architecture : Principles, Protocols, and Practices*. Wiley, 2009.
- [28] A. S. Tanenbaum and H. Bos. *Modern Operating Systems*. Pearson Education, 4th edition, 2014.

