

Dédicaces

En témoignage d'amour et d'affection nous dédions ce travail :

À nos très chers parents. Leurs considérables sacrifices, leurs encouragements et leur confiance ont été la source de réussite.
À nos amis qui nous n'oublierons jamais et qui seront toujours
dans nos pensées.

À tous nos enseignants et nos amis à l'ISIMM, que vous
trouviez dans ce travail l'expression de nos sentiments les plus
sincères.

À tous ceux qui comptent pour nous et à qui nous avons senti
essentiel de leur dédier ce simple travail.

Remerciements

C'est avec un grand plaisir que nous réservons cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui nous ont aidé et ont contribué de près ou de loin à la réalisation de ce travail.

Nous adressons nos sincères remerciements à Monsieur Akil Elkamel pour son précieuse assistance, son sincère encadrement, sa disponibilité, ses conseils précieux et l'intérêt qu'il a manifesté pour ce travail.

Pour la même occasion, nous exprimons nos profonds respects à toute l'équipe NITS pour sa disponibilité et ses conseils et aide précieux.

Nous remercions vivement tous les cadres de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir ainsi que tous les enseignants pour leurs judicieux conseils, leurs précieuses directions et pour la qualité de la formation qu'ils nous ont fournie.

Nous remercions les membres du jury de nous avoir fait l'honneur d'accepter d'évaluer ce modeste travail, et nous espérons que ce travail sera à la hauteur de la confiance qu'ils nous ont donné.

Résumé

Aujourd'hui la dispersion spatiale de l'habitat et des activités de ces dernières décennies a fortement contribué à une évolution constante de la demande en déplacement. Cette dispersion a entraîné une utilisation croissante de l'automobile et une remise en cause de l'offre de transports publics réguliers. Le véhicule particulier est devenu le mode de déplacement privilégié, offrant une simplicité, une efficacité et une rapidité bien supérieure au transport en commun.

Cependant, son utilisation massive et abusive dans les déplacements domicile-travail a entraîné de graves problèmes de congestion dans les villes, et une pollution accrue des milieux urbains. Chaque jour, aux heures de pointe, les entrées des villes sont soumises à une augmentation du nombre de véhicules.

Ces problèmes ont contraint les collectivités locales à envisager et à promouvoir de nouvelles pratiques moins onéreuses que les transports en commun, plus respectueuses de l'environnement et plus simples à mettre en œuvre.

C'est dans ce cadre-là que le covoiturage qui consiste à partager son véhicule avec d'autres usagers de la route pour effectuer tout ou une partie d'un trajet en commun, peut être une des solutions.

La mise en place de telles solutions est le plus souvent associée à des innovations technologiques qui jouent un rôle nécessaire pour leur bon fonctionnement et contribuent à leur donner une image attractive et dynamique.

Dans ce contexte nous avons eu la chance dans le cadre de notre projet de fin d'étude de concevoir et de développer un système de gestion de covoiturage qui permet de mettre facilement et rapidement en relation les conducteurs et les passagers.

Table des matières

Table des figures	vii
Liste des tableaux	viii
Introduction	1
Contexte de stage	1
Problématique	1
Présentation de l'organisme d'accueil	2
Structuration du rapport	3
1 Etude préalable	4
1.1 Présentation générale du projet	4
1.1.1 C'est quoi le covoiturage ?	4
1.1.2 Pourquoi faire du covoiturage ?	4
1.2 Etude de l'existant	5
1.2.1 Exemples des sites web de covoiturage	6
1.2.2 Exemples des applications mobiles de covoiturage	7
1.2.3 Le covoiturage via les réseaux sociaux	9
1.3 Généralités sur les téléphones mobiles intelligents	10
1.4 Les différents systèmes d'exploitation pour mobiles	11
1.5 Technologie iOS	13
1.6 Le langage : Objective C	14
1.7 Planning prévisionnel	14
2 Analyse et spécification des besoins	16
2.1 Spécification des besoins	16
2.1.1 Besoins fonctionnels	16
2.1.2 Besoins non fonctionnels	18
2.2 Spécification semi-formelle des besoins	19
2.2.1 Identification des acteurs	19
2.2.2 Diagramme global des cas d'utilisation	20
2.2.3 Descriptions textuelles des cas d'utilisation	21
3 Conception	24
3.1 Patron de conception MVC	24
3.1.1 Généralités sur MVC	24
3.1.2 L'architecture MVC et la plateforme iOS	25
3.2 Le langage de modélisation orienté objet : UML	26
3.3 La conception UML	26

3.3.1	Diagramme de composant	26
3.3.2	Diagramme de classe	27
3.3.3	Diagrammes des séquences	29
3.4	Conception graphique Maquette	34
4	Réalisation	38
4.1	Environnement de développement	38
4.1.1	Environnement matériel	38
4.1.2	Environnement logiciel	38
4.2	Architecture de système	39
4.3	Description du cycle de vie	40
4.4	Composition des modules	41
4.5	Réalisation de l'application mobile	43
4.5.1	Interface de lancement de l'application :	43
4.5.2	Interface de détail d'un trajet :	43
4.5.3	Interface de la carte de géolocalisation d'un trajet	44
4.5.4	Interface de consultation de profil :	44
4.5.5	Interface de recherche d'un trajet :	45
4.5.6	Interface de création d'un trajet :	45
4.5.7	Interface de gestion d'un trajet :	46
4.5.8	Interface de connexion :	46
4.5.9	Interface d'inscription :	47
4.5.10	Interfaces de réservation et de notification :	47
4.6	Réalisation de site web	48
	Conclusion et perspectives	52
	Bilan	52
	Perspectives	52
	Conclusion	52
	Bibliographie	53
A	Outils de développement	I
A.1	Rational Rose	I
A.2	ArgoUML	I
A.3	MySQL	II
A.4	MAMP	II
A.5	Joomla	III
A.6	AdsManager	IV
A.7	OmniGraffle	IV
A.8	Xcode	V
A.9	Photoshop	VI
A.10	TeXnicCenter	VII

Table des figures

1	Logo de NITS	2
1.1	Réduction du nombre des véhicules en fonction du nombre de covoitureurs.	5
1.2	Page d'accueil de site web "www.tunisiecovoiturage.com"	6
1.3	Page d'accueil de site web "www.blablacar.fr"	7
1.4	Logo de Carpooling	8
1.5	Des imprimes écran de l'application carpooling	8
1.6	Logo de Grand Lyon	9
1.7	Des imprimes écran de l'application covoiturage "Grand Lyon"	10
1.8	Un groupe Facebook de covoiturage Sfax-Tunis/Tunis-Sfax	11
1.9	La tendance du monde entier vers les OS mobiles	13
1.10	Les parts du marché des systèmes d'exploitation en Tunisie	13
1.11	Planning prévisionnel de déroulement du projet	14
2.1	Les acteurs du système	19
2.2	Diagramme général des cas d'utilisation	20
3.1	Présentation de modèle MVC en iOS	24
3.2	Architecture MVC de notre application iOS	25
3.3	Diagramme de composant	27
3.4	Diagramme de classe	28
3.5	Diagramme de séquence de cas d'utilisation "rechercher"	29
3.6	Diagramme de séquence de cas d'utilisation "s'inscrire"	30
3.7	Diagramme de séquence de cas d'utilisation "s'authentifier"	31
3.8	Diagramme de séquence de cas d'utilisation "proposer un trajet"	32
3.9	Diagramme de séquence de cas d'utilisation "Gérer une réservation"	33
3.10	Interface d'accueil	34
3.11	Interface d'un détail d'un trajet	35
3.12	Interface de la carte de géolocalisation	35
3.13	Interface d'authentification	36
3.14	Interface de recherche	36
3.15	Interface de proposition d'un trajet	37
4.1	Architecture matérielle de la solution proposée de notre système	40
4.2	Cycle de vie en spirale	41
4.3	Composition des modules	42
4.4	Interface d'accueil sur un iphone 4	43
4.5	Interface d'accueil sur un iphone 5	43
4.6	Interface de détail d'un trajet	43
4.7	Interface de la carte dynamique	44

4.8	Interface de profil	44
4.9	Interface de recherche	45
4.10	Interface de proposition d'un trajet	45
4.11	Formulaire de proposition d'une demande	45
4.12	Formulaire de proposition d'une offre	45
4.13	Interface de gestion des trajets	46
4.14	Interface d'authentification	46
4.15	Formulaire d'inscription de l'étape un	47
4.16	Formulaire d'inscription de l'étape deux	47
4.17	Interface de choix de la photo de profil	47
4.18	Interface des réservations reçues	47
4.19	Interface des notifications reçues	47
4.20	Interface d'accueil de site web	48
4.21	Formulaire d'inscription	49
4.22	Interface d'authentification	49
4.23	Formulaire de publication d'un trajet	50
4.24	Profil d'un utilisateur	50
A.1	Rational Rose	I
A.2	ArgoUML	I
A.3	MySQL	II
A.4	MAMP	II
A.5	CMS : Joomla	III
A.6	Qui utilise Joomla ?	III
A.7	AdsManager	IV
A.8	OmniGraffle	IV
A.9	Xcode	V
A.10	Photoshop	VI
A.11	TeXnicCenter	VII

Liste des tableaux

2.1	Description textuelle de cas d'utilisation "S'inscrire"	21
2.2	Description textuelle de cas d'utilisation "S'authentifier"	21
2.3	Description textuelle de cas d'utilisation "Rechercher un trajet"	22
2.4	Description textuelle de cas d'utilisation "Proposer un trajet"	22
2.5	Description textuelle de cas d'utilisation "Gérer une réservation"	23

Introduction

Contexte de stage

Le présent travail s'inscrit dans le cadre du projet de fin d'étude en vue de l'obtention du diplôme de Licence Appliquée en Informatique de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir. Notre projet est réalisé au sein de la société Nearshore IT Solutions. Ce stage a duré quatre mois à partir du 04/02/2013.

Problématique

Le covoiturage représente une pratique en cour de développement qui constitue un mode de transport nouveau et complémentaire aux autres modes de transport. Le covoiturage, en toute généralité, est un mode de déplacement qui consiste l'utilisation conjointe et organisée d'un véhicule privé par plusieurs personnes, dans le but d'effectuer toute ou une partie d'un trajet commun occasionnel ou régulier.

Cette pratique est encouragée aux Etats-Unis depuis le début des années 1970 lors de la première crise pétrolière, elle bénéficie parfois de véhicules coups de pouce de la part des gouvernements où des multiples facilités sont offertes aux usagers (tarifs préférentiels, places de parking et voies de circulation réservées, etc.). Des associations ont également contribué au développement de cette pratique depuis les années 1980 en Allemagne, en Belgique, et au Canada[1].

Selon les estimations du Ministère du transport en 2011, 50 % des Tunisiens utilisent leurs voitures pour se rendre au travail. 1% de nos concitoyens seulement partagent régulièrement un véhicule[2]. Nous pouvons expliquer l'absence de cette pratique dans notre pays par l'indisponibilité des moyens et des outils fiables et efficaces pour mettre en contact les conducteurs et les passagers.

Le but de ce projet est de concevoir et développer une application mobile et un site web qui assurent la communication entre les voyageurs.

Présentation de l'organisme d'accueil

NITS est une société de service NearShore fondée en 2009 spécialisée dans le développement des solutions informatiques. Elle propose à ses clients des solutions sur mesure ou d'intégration open source, clé en main et qui respectent les plus hauts standards de développements informatiques. Les logiciels qu'elle développe pour ses clients sont mis en œuvre suivant les meilleurs standards tel que la méthodologie agile. Nearshore IT Solutions offre des services de développement, de maintenance, de test, de conseil et d'intégration dans les domaines suivants :

- Développement Web
- Développement mobile
- Développement Open Source

NITS est une start-up spécialisée dans la conception des applications mobiles. Elle a pour vocation d'accompagner ses clients dans la définition et la mise en place d'une stratégie mobile innovante. Elle a pour ambition d'aider ses clients à définir une stratégie pour la conquête et la fidélisation des clients, de la mettre en œuvre en créant des véritables produits attrayants, ainsi que l'accompagnement dans toute la vie des produits développés. Ses fondateurs ont acquis une expérience de plusieurs années dans l'innovation technologique, le développement d'applications mobiles et les services aux entreprises dans l'informatique.



FIGURE 1 – Logo de NITS

Afin d'offrir une solution efficace en termes de coût à ses clients, Nearshore IT Solutions a noué un partenariat stratégique avec un centre de développement offshore. Ce centre existe depuis 2008 et compte des designers, plusieurs dizaines de développeurs, de business analysts, d'ingénieurs en qualité de logiciel.

Structuration du rapport

Le présent rapport est organisé comme suit. Nous commençons par l'étude préalable dans le premier chapitre, le deuxième chapitre est consacré pour l'analyse, la spécification des besoins fonctionnels et non fonctionnels et la spécification semi formelle, le troisième chapitre présente la conception de l'application mobile pour mettre en œuvre les principales fonctionnalités du système, le dernier chapitre est réservé pour la réalisation dans laquelle nous présentons la technologie utilisée, enfin nous clôturons ce rapport par une conclusion dans laquelle nous présentons le bilan de notre travail et nous proposons quelques perspectives.

Chapitre 1

Etude préalable

Introduction

Dans ce chapitre, nous mettons en œuvre notre projet dans son contexte général. Nous commençons par une présentation générale, ensuite nous étudions l'existant et les solutions disponibles sur le marché afin de bien effectuer notre choix pour s'échapper des limites des applications et des méthodes existantes et améliorer leurs points forts dans notre solution.

1.1 Présentation générale du projet

1.1.1 C'est quoi le covoiturage ?

Le covoiturage est l'utilisation conjointe et organisée d'un véhicule, par un conducteur et un ou plusieurs passagers, dans le but d'effectuer un trajet commun. Le conducteur propose de partager sa voiture pour un trajet donné avec la date et l'heure précisées d'avance. Les frais sont laissés à son appréciation. Le plus classique est de calculer le coût du trajet comprenant les frais de carburants et de péages s'il y en a et de diviser le montant par le nombre de personnes au moment du départ.

1.1.2 Pourquoi faire du covoiturage ?

Le covoiturage permettra aux utilisateurs de réduire leurs dépenses liées au déplacement. Cette pratique permet de développer le sens de la solidarité puisque les covoitureurs prennent l'habitude de s'aider mutuellement.

Avec le covoiturage, le risque d'accident de la route est également statistiquement réduit, puisqu'il est proportionnellement lié au nombre de voitures particulières en circulation. En plus le partage des voitures permet de réduire la pollution.

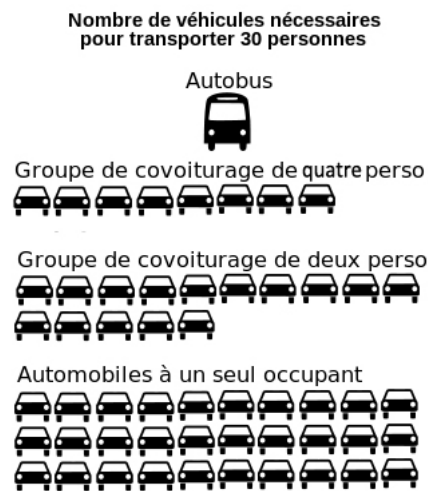


FIGURE 1.1 – Réduction du nombre des véhicules en fonction du nombre de covoitureurs.

Le transport routier étant la source de 20% des émissions globales de gaz, dont la moitié provient des voitures. Le covoiturage est alors une manière conviviale, économique et écologique de voyager. L'économie sur le plan financier et sur le plan des émissions de CO₂ peut être considérable, pour faciliter la coordination entre les différents covoitureurs, il faut avoir un système qui permet d'offrir des fonctionnalités permettant d'aboutir à la coordination entre les covoitureurs.

Dans ce cadre, il s'agit de créer un système de covoiturage qui dispose de deux parties à réaliser, une partie Web et une partie Mobile. Le site web doit répondre à un besoin devenu indispensable aujourd'hui, nous comptons sur ce site web pour créer une communauté libre autour du covoiturage, et puisque les utilisateurs ciblés de notre solution sont caractérisés par leurs mobilités, une application mobile pour les smartphone est envisageable, pour satisfaire leurs besoins n'importe où et à tous moment.

1.2 Etude de l'existant

Dans cette section, nous menons une étude basée sur les observations des différents sites web d'actualité ainsi que leurs applications mobiles similaires à notre application et lister leurs points forts et leurs points faibles. Ces observations permettront de bien définir les grandes lignes de notre solution. Ainsi le but de cette étude est de déduire les fonctionnalités que nous devons intégrer dans notre application. Pour pouvoir percevoir les besoins, nous nous sommes basés principalement sur la diversité, le contenu et la qualité des applications étudiées.

1.2.1 Exemples des sites web de covoiturage

Tunisiecovoiturage

Description

C'est le premier site internet de covoiturage en Tunisie, lancé en novembre 2011, Ce site met en relation des conducteurs et des passagers ayant une destination commune et désirant partager un véhicule par souci d'économie ou tout simplement pour de la compagnie. Le site permet à ces derniers d'accéder à des centaines d'offres en Tunisie, déposées par des étudiants et des particuliers membres. Et en chiffres, 1 000 membres sont déjà inscrits sur le site et 450 trajets y sont proposés en seulement 9 mois d'existence[3].



FIGURE 1.2 – Page d'accueil de site web "www.tunisiecovoiturage.com"

Les points forts

Le majeur avantage de ce site est qu'il contient des interfaces conviviales et ergonomiques exploitables par l'utilisateur d'où il permet une utilisation simple et facile. La recherche par les villes et les gouvernorats représente un point fort pour ce site. Aussi il dispose d'une carte des trajets qui précise les lieux de départ et d'arrivée pour chaque annonce.

Les points faibles

Le site de covoiturage tunisiecovoiturage.com n'offre pas un système de réservation pour faciliter la communication entre les passagers et les conducteurs. Aussi il ne permet pas la publication des demandes d'où les passagers ne peuvent pas poster leurs trajets demandés.

BlaBlaCar

Description

BlaBlaCar, anciennement Covoiturage.fr, est une plateforme communautaire payante de covoiturage conçue en 2004 par Frédéric Mazzella puis lancée officiellement en 2006 par l'intermédiaire de l'entreprise Comuto. La mise en relation entre usagers, gratuite pendant de nombreuses années, est devenue payante en 2011, de l'ordre de 20% du prix du covoiturage. Avec 10 millions d'utilisateurs en 2014, BlaBlaCar est leader du covoiturage en Europe .[4].

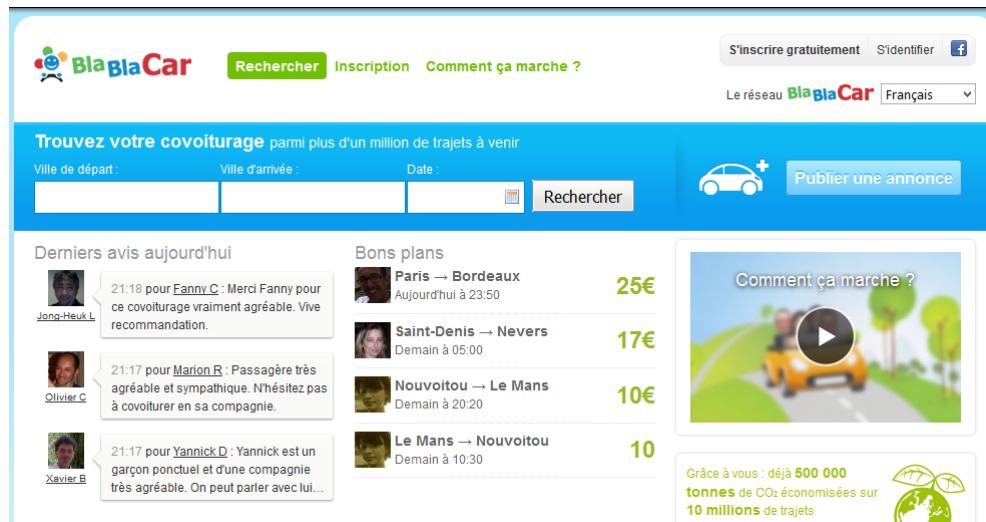


FIGURE 1.3 – Page d'accueil de site web "www.blablacar.fr"

Les points forts

Ce web site offre plusieurs fonctionnalités, telles que : la carte du territoire, la cartographie routière avec l'itinéraire détaillé du trajet, la réservation. En plus, L'inscription est facile. Il est même possible de le faire directement via son compte Facebook

Les points faibles

Le temps calculé lorsqu'on indique des villes intermédiaires pour prendre des passagers ne tient pas compte des arrêts pour les pauses. En plus tout contact direct entre conducteurs et passagers n'est pas possible avant de payer.

1.2.2 Exemples des applications mobiles de covoiturage

Carpooling

Description

Carpooling gère le transport de plus d'un million de personnes par mois à travers 45

pays. Avec plus de 650 000 offres de covoiturage en France et en Europe. C'est le leader européen.



FIGURE 1.4 – Logo de Carpooling

Carpooling permet de mettre en relation les conducteurs et les passagers, cette application est gratuite et présente des nombreuses fonctionnalités permettant à ses utilisateurs de profiter d'une expérience de covoiturage fiable, agréable et économique. Carpooling possède un site web qui offre à ses utilisateurs plusieurs fonctionnalités. Ce site est présent dans des nombreux pays : France, Belgique, Allemagne, Suisse, Autriche, Espagne, Italie, Royaume-Uni, Pologne et Grèce, et est traduit en 7 langues[5].

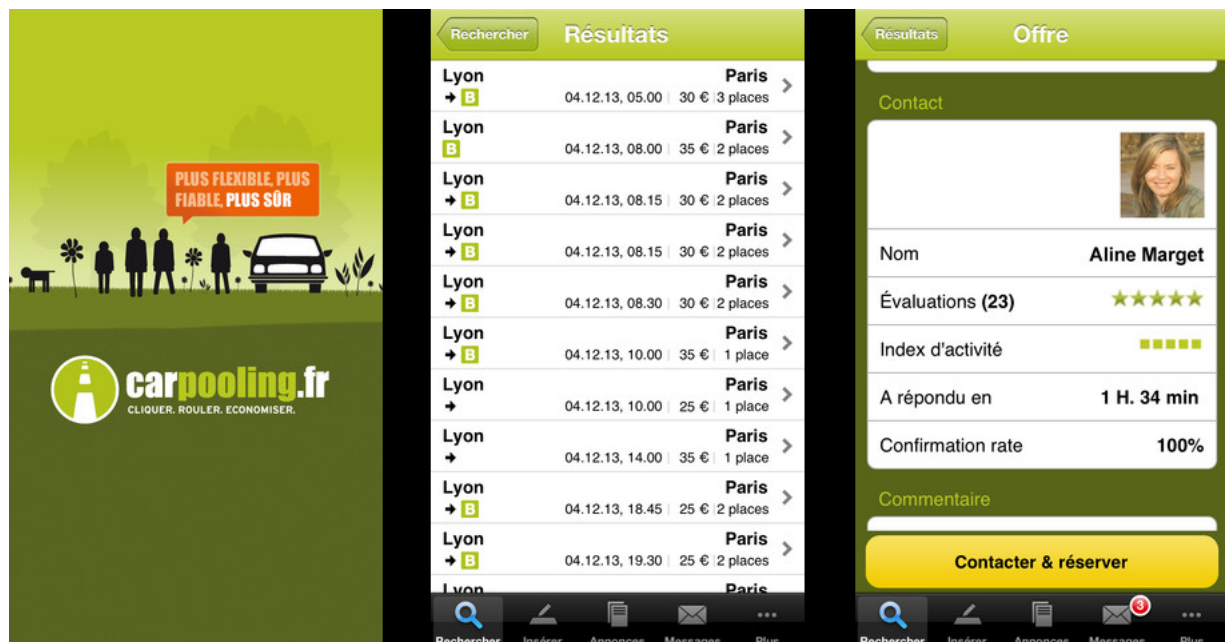


FIGURE 1.5 – Des imprimées écran de l'application carpooling

Les points forts

Carpooling offre un système de notification pour avertir les membres du système de toutes les nouveautés et les actualités, elle dispose aussi d'un système de feedback qui permet l'évaluation du conducteur pour assurer la sincérité de l'application.

Les points faibles

L'application Carpooling est la plus utilisée en Europe, malgré qu'elle contient plusieurs limites, d'une part elle n'offre pas la possibilité de s'inscrire à partir de l'application mobile, la création d'un nouveau compte se fait uniquement à partir du site web, d'autre part

l'application ne dispose pas d'une carte de localisation pour visualiser les trajets. En plus les interfaces ne sont pas ergonomiques et la navigation est parfois un peu compliquée.

Covoiturage Grand Lyon

Description

Cette plateforme de covoiturage est proposée par Le Grand Lyon depuis 2009 dans le cadre de sa politique pour une mobilité durable. Le portail Covoiturage Grand Lyon est accessible à l'ensemble des salariés des zones économiques du Grand Lyon. Il permet de mettre en relation et de trouver rapidement des covoitureurs pour des trajets domicile-travail. L'application covoiturage Grand Lyon est 100% gratuite et disponible pour iPhone, iPad et Android[6].



FIGURE 1.6 – Logo de Grand Lyon

Les points forts

L'application de Covoiturage Grand Lyon propose plusieurs fonctionnalités pratiques telles que la recherche rapide d'un trajet, la déposition des offres, la réception des alertes et le repérage des covoitureurs sur une carte dynamique, etc. L'application offre une interface conviviale et ergonomique exploitable par l'utilisateur. Elle est rapide et simple d'où elle facilite son usage par les utilisateurs.

Les points faibles

Cette application ne dispose pas d'un système de réservation qui a pour but d'assurer une mise en contact gratuite entre les membres de covoiturage. En plus nous remarquons l'indisponibilité d'un système de notification.

1.2.3 Le covoiturage via les réseaux sociaux

Aujourd'hui, l'utilisation de covoiturage, via les réseaux sociaux comme Facebook, permet d'augmenter le nombre des covoitureurs. Pour cela, plusieurs groupes sur ces réseaux sont fournis afin de permettre aux membres de s'organiser pour voyager ensemble.



FIGURE 1.7 – Des imprimés écran de l'application covoiturage "Grand Lyon"

Les points forts

L'utilisation des réseaux sociaux Twitter, Facebook comme plateforme de covoiturage dispose d'un potentiel de plus de 5 millions de membres qui peuvent être intéressés par du covoiturage entre amis ou connaissances.

Les points faibles

Cette technique de covoiturage n'est pas fiable pour plusieurs raisons, tout d'abord l'absence d'un système de réservation. En plus l'indisponibilité d'un système de recherche, ce qui provoque une grande difficulté pour trouver l'offre souhaitée. Aussi le manque des fonctionnalités utiles et importantes comme la carte de localisation des trajets et les outils de communications entre les covoitureurs courrier, SMS, etc. Le système de notification dans les réseaux sociaux comme Facebook n'est pas ciblé essentiellement pour le covoiturage.

1.3 Généralités sur les téléphones mobiles intelligents

Le Smartphone ou "mobile intelligent" révolutionne le marché de la téléphonie mobile. C'est un téléphone intelligent, initialement conçu pour gérer les activités des professionnels lors de leurs déplacements. Il est devenu un objet dont sa fonction première, téléphoner, est devenue quasiment accessoire avec la présence de plusieurs fonctionnalités comme l'appareil photo, chaînes hi-fi, GPS, et lecteur MP3 etc. ; l'intégration des plusieurs technologies tels que le Bluetooth, wifi, 3G ; sans oublier ses capacités énormes de stockage et l'implémentation des processeurs puissants aussi les divers systèmes d'exploitation qui le rendent très performant au point qu'il peut remplacer les autres appareils High Tech dans plusieurs fonctionnalités.



FIGURE 1.8 – Un groupe Facebook de covoiturage Sfax-Tunis/Tunis-Sfax

Le Smartphone a su se faire apprécier de tout le monde avec ses formes coulissant, tactile, à clapet ou monobloc, et encore, les multiples marques qu'on trouve sur le marché. Les marchés mobiles ont aujourd'hui atteint un niveau de maturité concernant le développement de la téléphonie mobile, son accessibilité, sa convivialité et surtout, sa croissance ascendante, ce qui permet un meilleur développement pour les technologies de l'information et de la communication. Les applications mobiles sont devenues un moyen essentiel de création des nouveaux services à destination des utilisateurs. Les applications sur les Smartphones et les tablettes sont un moyen d'atteindre efficacement le consommateur.

1.4 Les différents systèmes d'exploitation pour mobiles

L'écosystème des applications mobiles est essentiellement composé des deux acteurs Apple avec iOS et Google avec Android. Il reste d'autres acteurs relativement importants comme RIM ou Windows avec leurs OS respectifs BlackBerry et Windows Phone. Aujourd'hui, ces dernières plateformes sont marginales sur le marché. Les entreprises désirent développer une application sur ces plateformes sont déjà convaincues de la pertinence de

ces OS pour leurs segments de marché. Les principaux systèmes d'exploitation dédiés aux téléphones mobiles sont :

- iOS : Il était autrefois appelé iPhone OS, développé par Apple pour ses terminaux mobiles, ce système d'exploitation est dérivé du Mac OS X partageant ainsi son kernel Mach et plusieurs autres fondations. Chaque développeur désirant publier des applications tournant sous iOS doit s'adhérer au programme des développeurs Apple, en payant la somme de 99 \$ par an. Avec plus de 19,6 millions d'iPhone vendus iOS possède 25,2% des parts de marché de Smartphones au monde[7].
- Android : est un système d'exploitation open source utilisant le noyau Linux, pour Smartphones, PDA et terminaux mobiles conçu par Android, puis racheté par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs et des tablettes[8].
- Windows Phone : est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, Windows Phone est principalement destiné au grand public plutôt qu'au marché des entreprises. Cependant depuis Windows Phone 8, Microsoft propose des fonctions avancées pour les entreprises en offrant, par exemple, un espace d'applications réservé aux entreprises. Il a été lancé le 21 octobre 2010 en Europe, à Singapour, en Australie et en Nouvelle-Zélande, le 8 novembre 2010 aux États-Unis et au Canada[9].

La figure 1.9 représente les statistiques de la tendance du monde entier vers les systèmes d'exploitation des smartphones en avril 2013[10]. Ainsi la figure 1.10 montre les parts du marché des systèmes d'exploitation en Tunisie[11].

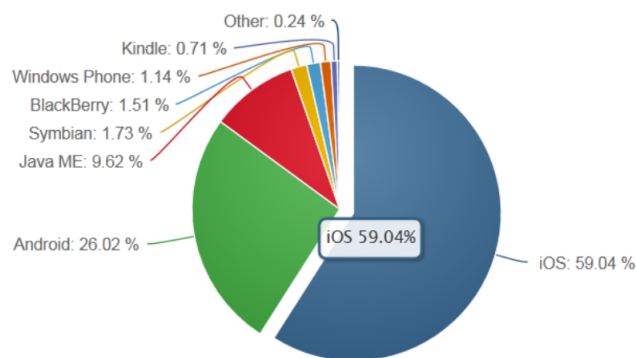


FIGURE 1.9 – La tendance du monde entier vers les OS mobiles

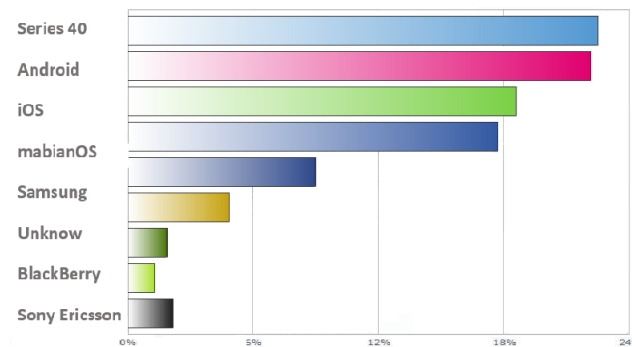


FIGURE 1.10 – Les parts du marché des systèmes d'exploitation en Tunisie

1.5 Technologie iOS

iOS est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch et l'iPad. Il est dérivé de Mac OS X.

iOS comporte quatre couches d'abstraction, similaires à celles de Mac OS X : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ». Le système d'exploitation occupe moins d'un demi-gigaoctet (Go) de la capacité mémoire totale de l'appareil. Ce système d'exploitation n'avait aucun nom officiel avant la publication du kit de développement iPhone (SDK) le 6 mars 2008. Jusqu'à cette date, Apple se contentait de mentionner que « l'iPhone tourne sous OS X », une référence ambiguë au système d'exploitation source d'iOS, Mac OS X. Ce n'est qu'à cette occasion que Scott Forstall présenta l'architecture interne du système d'exploitation, et dévoila alors le nom d'iPhone OS. Ce nom a été changé le 7 juin 2010 pour iOS.

Le kit de développement en question, disponible pour Mac OS X, propose les outils nécessaires à la création d'une application pouvant tourner sous iOS. Si son téléchargement et son utilisation sont gratuits, la publication de telles applications requiert d'adhérer au programme des développeurs Apple, pour la somme de 99 \$ par an. Il n'en demeure pas moins que cette offre peut s'avérer intéressante pour bon nombre de développeurs, étant donnée la taille du marché créé par iOS[12].

1.6 Le langage : Objective C

L'Objective-C est un langage de programmation orienté objet réflexif. C'est une extension du C, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Contrairement au C++, il ne permet pas l'héritage multiple mais il existe toutefois de moyens de combiner les avantages de C++ et d'Objective-C.

1.7 Planning prévisionnel

Le figure 1.11 montre le planning prévisionnel de déroulement du projet, nous possédons une période de deux semaines pour l'étude de l'existant et une semaine pour la documentation, ensuite nous allons commencer la conception qui durera environ un mois et à partir de mi-mars nous allons commencer à réaliser le projet, enfin nous allons accomplir notre travail par la rédaction du rapport.

	Février				Mars				Avril				Mai			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Etude de l'existant																
Documentation																
Conception																
Réalisation																
Rédaction de rapport																

FIGURE 1.11 – Planning prévisionnel de déroulement du projet

Conclusion

Au cours de ce chapitre, nous avons présenté le contexte du projet avant de citer et critiquer les méthodes et les applications qui sont encore utilisables malgré les points faibles qui les empêchent parfois de remplir leurs tâches convenablement. En plus nous avons jeté un coup d’œil sur les systèmes d’exploitations pour mobiles et défini en particulier la technologie iOS que nous allons l’exploiter pour le développement de l’application mobile. Nous présentons dans le chapitre suivant un aperçu sur l’analyse et la spécification des besoins.

Chapitre 2

Analyse et spécification des besoins

Introduction

L'analyse du problème et la spécification des besoins constituent une étape primordiale du développement de toute application. Dans ce chapitre, tout d'abord nous allons identifier les besoins fonctionnels et les besoins non fonctionnels de l'application mobile, par la suite nous allons détailler le diagramme des cas d'utilisation. Entre la spécification formelle et la spécification semi formelle nous devons penser à proposer une solution efficace qui répond aux besoins des utilisateurs.

2.1 Spécification des besoins

Notre objectif est de développer une application constituant la base d'un système de covoiturage qui permet de résoudre les limites et les inconvénients précédemment listés dans l'étude de l'existant.

2.1.1 Besoins fonctionnels

Cette application devrait avoir les fonctionnalités suivantes :

Consultation des trajets :

Un utilisateur peut consulter les trajets ainsi que leurs détails.

Recherche :

La recherche à travers le site web ou à travers l'application mobile doivent permettre aux utilisateurs de consulter et d'obtenir rapidement tous les trajets disponibles. Ainsi plusieurs critères doit être considérés pour faciliter la recherche à l'utilisateur selon la catégorie, la date, la ville de départ, la ville d'arrivée et le type de trajet (aller simple ou aller-retour).

Inscription :

L'inscription sur le site web ou à travers l'application mobile doit être simple et rapide. Dès la création de son compte, l'utilisateur doit pouvoir immédiatement commencer à publier ses annonces.

Authentification :

A travers un nom d'utilisateur et un mot de passe attribué à un compte déjà créé, l'utilisateur doit pouvoir se servir de toutes les fonctionnalités offertes par l'application.

Publication d'un trajet :

Le conducteur doit pouvoir proposer un trajet reliant une ville A à une autre B. Lors de la création, il doit indiquer le type de trajet, les nombres des places disponibles, la date et la période ainsi que d'autres informations selon la catégorie du trajet choisi (offre ou demande).

Gestion des trajets :

L'utilisateur doit pouvoir consulter ses annonces, par la suite il pourra les supprimer ou modifier leurs propriétés.

Réservation :

Un passager doit avoir la possibilité d'effectuer une réservation dans le trajet qui lui convient. Le conducteur doit être prévenu automatiquement par une notification.

Gestion des réservations :

Le conducteur doit pouvoir consulter toutes les réservations des passagers. Ensuite il pourra répondre à ces réservations par l'acceptation ou le refus.

Consultation du profil :

Un utilisateur authentifié au système disposera à tout moment la permission de consulter le profil d'un annonceur de trajet.

Géolocalisation :

La carte doit permettre aux utilisateurs de visualiser très rapidement les villes concernées par un trajet. La ville de départ et la ville d'arrivée doivent être respectivement identifiées.

Communication :

La prise de contact avec une personne ayant déposé une annonce de trajet doit être à partir de l'espace de présentation du trajet, pour effectuer une communication entre les différents membres du système de covoiturage l'utilisateur doit pouvoir appeler ou envoyer un message ou un email aux autres covoitureurs.

Notification :

Un système de notification doit être mis en place pour avertir l'utilisateur des nouvelles réservations ou des réponses du conducteur, ce système doit permettre l'utilisation rapide et faciliter la communication entre l'utilisateur et l'application.

2.1.2 Besoins non fonctionnels

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telles les contraintes liées à l'environnement et à l'implémentation et les exigences en matière de performance. L'application mobile doit être à la mesure de satisfaire l'utilisateur à travers les critères suivants :

Ergonomie :

L'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur, elle doit être simple pour faciliter leur usage par les utilisateurs.

Efficacité :

L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.

Performance :

Nous insistons à donner une application performante, coté temps de réponse, le chargement de l'application, l'ouverture d'écran, le temps de traitement.

Sécurité :

Le système demande à l'utilisateur un nom d'utilisateur et un mot de passe. Ce dernier est crypté au niveau de la base de données pour garantir plus de sécurité.

Simplicité :

L'application doit être simple et rapide. Ainsi, les utilisateurs doivent faire un minimum d'efforts pour publier leurs trajets et pour naviguer à travers les interfaces de

l'application. La navigation doit être intuitive, conviviale, simple à comprendre et facile à mémoriser.

Expérience utilisateur :

L'application doit se caractériser par des principes nécessaires : ergonomie, technologie ; l'objectif est de mixer des informations cohérentes de divers horizons de manière à qualifier au plus juste le ressenti de l'utilisateur vis-à-vis le service offert par l'application : notoriété, satisfaction, confiance, mémoriel.

Maintenabilité :

Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

2.2 Spécification semi-formelle des besoins

Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable et intéressant pour un acteur particulier. En effet, les cas d'utilisation sont des représentations fonctionnelles du système, ils permettent de modéliser les attentes des utilisateurs afin de réaliser une bonne délimitation du système et également d'améliorer la compréhension de son fonctionnement.

2.2.1 Identification des acteurs

La figure 2.1 représente les acteurs de notre système :



FIGURE 2.1 – Les acteurs du système

Un visiteur est une personne extérieure au système ne possédant pas un compte mais souhaitant se renseigner sur les annonces proposées.

Un covoitureur est l'acteur principal de notre système, il possède un compte et il peut consulter ou proposer des trajets, de plus il dispose d'autres fonctionnalités que nous allons détailler à la suite de ce chapitre.

2.2.2 Diagramme global des cas d'utilisation

Le diagramme des cas d'utilisation permet de structurer les besoins des utilisateurs et les objectifs correspondants d'un système et d'identifier les acteurs et leurs interactions avec le système.

La figure 2.2 présente le diagramme général des cas d'utilisation qui montre toutes les fonctionnalités de l'application.

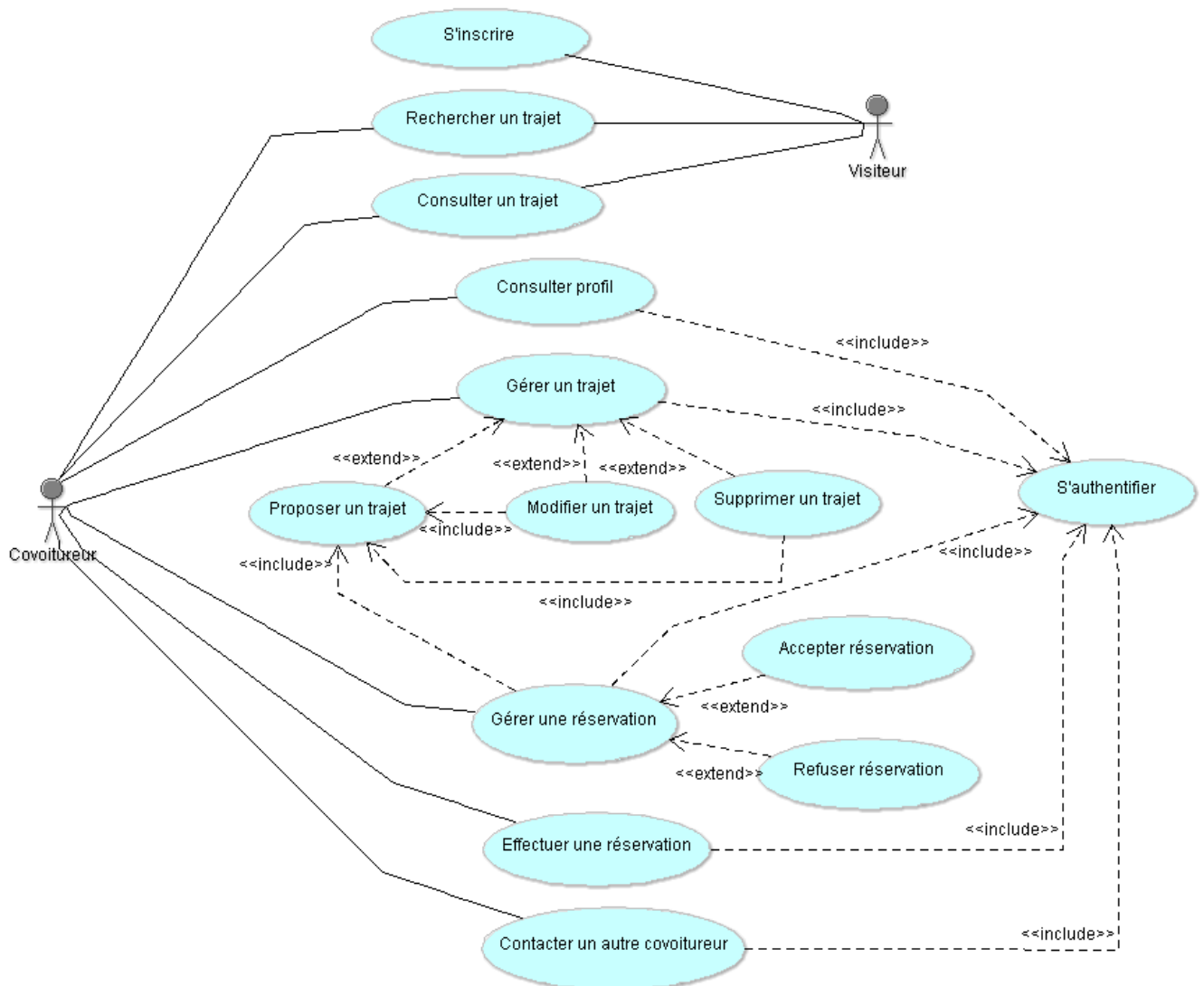


FIGURE 2.2 – Diagramme général des cas d'utilisation

Ce diagramme représente les cas d'utilisation sans en montrer les détails, chaque cas d'utilisation sera détaillé dans ce qui suit.

2.2.3 Descriptions textuelles des cas d'utilisation

Description textuelle de cas d'utilisation : S'inscrire

Acteur	Visiteur
Objectif	Créer un compte
Pré-conditions	Afficher l'interface de l'inscription
Scénario nominal	1. Saisir le nom d'utilisateur 2. Saisir l'email 3. Saisir le mot de passe 4. Confirmer le mot de passe 5. Remplir les autres champs de formulaire (nom, date de naissance, sexe, ville, téléphone) 6. Choisir la photo du profil
Scénario alternatif	Cas1 : Si le nom d'utilisateur existe déjà : retour à 1 Cas2 : Si le mot de passe n'est pas confirmé : retour à 3 puis à 4 Cas3 : Si l'une des informations entrées par l'utilisateur est syntaxiquement invalide : Il doit corriger le champ en erreur
Post-conditions	Utilisateur inscrit à l'application.

TABLE 2.1 – Description textuelle de cas d'utilisation "S'inscrire"

Description textuelle de cas d'utilisation : S'authentifier

Acteur	Covoitureur
Objectif	Se connecter au système
Pré-conditions	L'utilisateur possède déjà un compte
Scénario nominal	1. Saisir le nom d'utilisateur 2. Saisir le mot de passe 3. Valider les champs 4. Le système vérifie le remplissage des champs 5. Le système vérifie la validité des données
Scénario alternatif	Cas1 : Si le login est invalide : retour à 1 Cas2 : Si le mot de passe est invalide : retour à 1
Post-conditions	L'utilisateur est connecté et il est prêt à exploiter tous les services offerts par le système de covoiturage.

TABLE 2.2 – Description textuelle de cas d'utilisation "S'authentifier"

Description textuelle de cas d'utilisation :Rechercher un trajet

Acteur	Visiteur, covoitureur
Objectif	Rechercher les offres et les demandes des covoiturages proposés pour un trajet bien déterminé.
Préconditions	Aller à l'interface de recherche d'un trajet
Scénario nominal	1. Saisir la catégorie (offre ou demande) 2. Saisir la ville de départ 3. Saisir la ville d'arrivée 4. Saisir la date de départ 5. Saisir le type de trajet (aller ou aller-retour) 6. Valider les champs
Post-conditions	Affichage d'une liste des trajets qui satisfait les critères de recherche de l'utilisateur s'ils existent.

TABLE 2.3 – Description textuelle de cas d'utilisation "Rechercher un trajet"

Description textuelle de cas d'utilisation : Proposer un trajet

Acteur	Covoitureur
Objectif	Publier un trajet
Pré-conditions	Le covoitureur est déjà authentifié
Scénario nominal	1. Remplir le formulaire d'une demande ou d'une offre 2. Valider les champs 3. Publier l'annonce
Scénario alternatif	Si l'une des informations entrées par l'utilisateur est syntaxiquement invalide : Il doit corriger le champ en erreur
Post-conditions	Annonce publiée avec succès.

TABLE 2.4 – Description textuelle de cas d'utilisation "Proposer un trajet"

Description textuelle de cas d'utilisation : Gérer une réservation

Acteur	Covoitureur
Objectif	Accepter ou refuser une réservation de trajet
Pré-conditions	Le conducteur est déjà authentifié Le conducteur a déjà eu des réservations des trajets.
Scénario nominal	Un conducteur peut : – Accepter une réservation – Refuser une réservation
Scénario alternatif	Si l'annonce a été supprimée par son propriétaire toute réservation qui lui est associée sera automatiquement effacée.
Post-conditions	Réservation de trajet est acceptée ou refusée.

TABLE 2.5 – Description textuelle de cas d'utilisation "Gérer une réservation"

Conclusion

Cette description générale décrit les fonctionnalités de l'application. Lister les besoins fonctionnels et les besoins non fonctionnels est en fait une réponse à la question « quelles sont les fonctions du système ? ». Dans le chapitre suivant nous détaillerons la conception à travers le diagramme de composant, le diagramme de classe et les diagrammes de séquence.

Chapitre 3

Conception

Introduction

Après l'analyse et la spécification des besoins, nous présentons dans ce chapitre la conception détaillée, qui a pour principal objectif de structurer, de comprendre l'application et de modéliser le problème d'une façon orientée objet. Dans la première partie nous allons représenter le patron de conception MVC (Modèle, Vue, Contrôleur), dans la deuxième partie nous allons définir le langage de modélisation UML, ensuite nous détaillons la conception de l'application mobile à travers les principaux diagrammes, enfin nous allons présenter la conception graphique en figurant les maquettes de notre application.

3.1 Patron de conception MVC

3.1.1 Généralités sur MVC

MVC est un patron de conception ayant pour objectif d'organiser une application en trois parties distinctes :

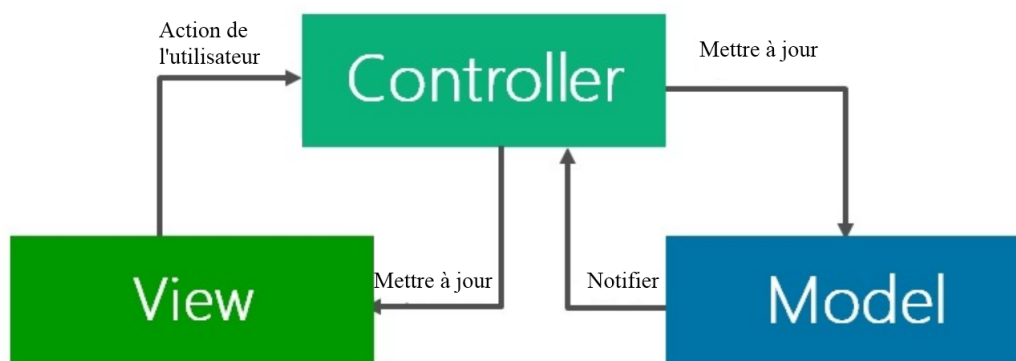


FIGURE 3.1 – Présentation de modèle MVC en iOS

- **Modèle** : il représente les données et les règles métiers (traitement des données, la persistance des données, etc.). C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données ou des services Web. La partie modèle définit également les méthodes d'accès.
- **Vue** : elle correspond à l'interface d'interaction avec l'utilisateur (l'IHM). Les résultats renvoyés par le modèle sont dénués de tout formatage mais sont présentés par les vues. Plusieurs vues peuvent afficher les informations d'un même modèle. La vue n'effectue aucun traitement, elle se contente d'afficher les résultats des traitements effectués par le modèle, et de permettre à l'utilisateur d'interagir avec elle.
- **Contrôleur** : il prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle.

3.1.2 L'architecture MVC et la plateforme iOS

Pour le développement de la partie mobile de notre application de covoiturage, nous adoptons le patron de conception MVC.

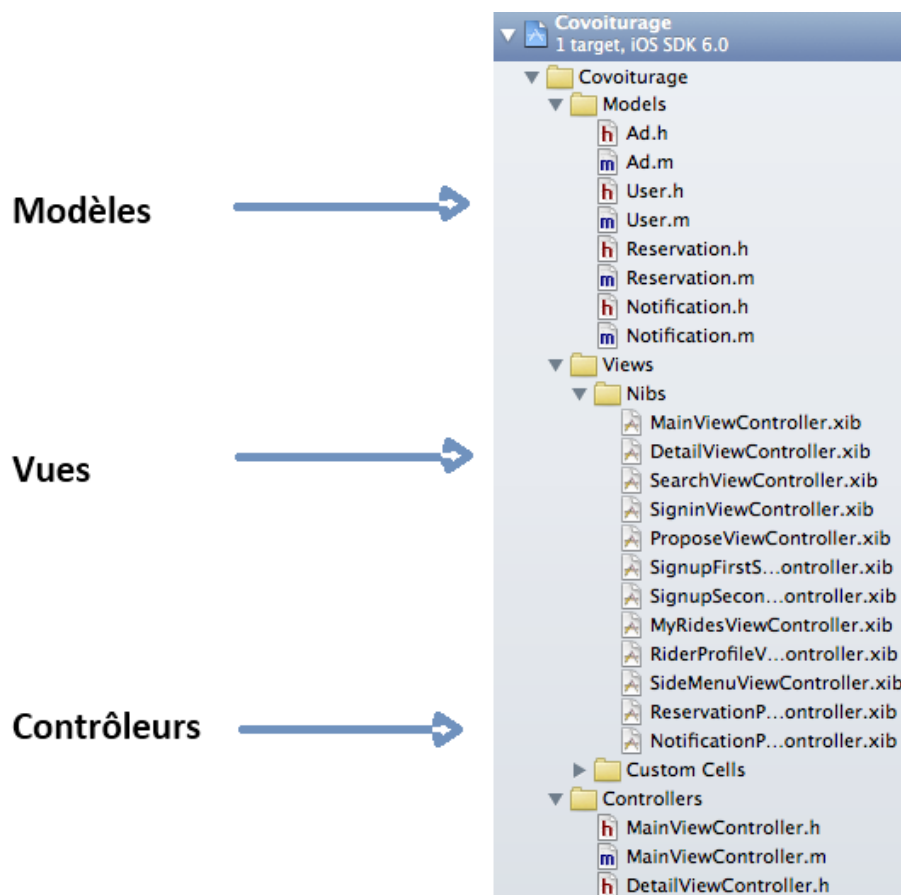


FIGURE 3.2 – Architecture MVC de notre application iOS

3.2 Le langage de modélisation orienté objet : UML

UML (Unified Modeling Language) est un langage de modélisation graphique. Il est apparu dans le monde du génie logiciel, dans le cadre de la conception orientée objet. Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Ce langage de modélisation utilise des diagrammes pour représenter chaque aspect d'un système : statique, dynamique et fonctionnel en s'appuyant sur la notion d'orienté objet qui est un véritable atout pour ce langage[13].

UML est un langage pseudo-formel qui est fondé sur un métamodèle, qui définit les éléments de modélisation et la sémantique de ces éléments, d'où sa notation graphique permet d'exprimer visuellement une solution objet et de simplifier l'abstrait. L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

Il vaut mieux modéliser un système avant de le réaliser, c'est pour cette raison nous prendrons UML afin d'obtenir une modélisation de très haut niveau indépendante des langages et des environnements et de faciliter la compréhension de représentations abstraites complexes.

3.3 La conception UML

3.3.1 Diagramme de composant

Le diagramme de composant permet de représenter les composants logiciels d'un système ainsi que les liens existant entre ces composants. Ce diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules et les bases de données[14].

La figure 3.3 présente les majeurs composants identifiés dans l'architecture générale de notre système.

- Côté client : Cette couche correspond à tous les objets spécifiques que nous allons manipuler. Typiquement dans notre cas, nous allons manipuler des clients. Elle contient le contenu graphique de l'application mobile ou notre site web ; on retrouve également dans cette couche tout ce qui compose une IHM (interface homme machine) d'un point de vue graphique et statistique. De plus on trouve la couche métier de l'application client, qui est chargée de se connecter au composant web service de l'application serveur et récupérer les données. Le rôle de cette partie est d'offrir une interface qui permet aux utilisateurs de consulter les données de l'application serveur.

- Côté serveur : Le serveur est constitué de plusieurs scripts effectuant le rôle des web services, ces scripts sont l'intermédiaire entre l'application client et la couche d'accès aux données. Ils contiennent les méthodes qui permettent de récupérer et traiter les données.
- Données : MySQL C'est le serveur de notre base de donnée centralisé ; il représente notre SGBD.

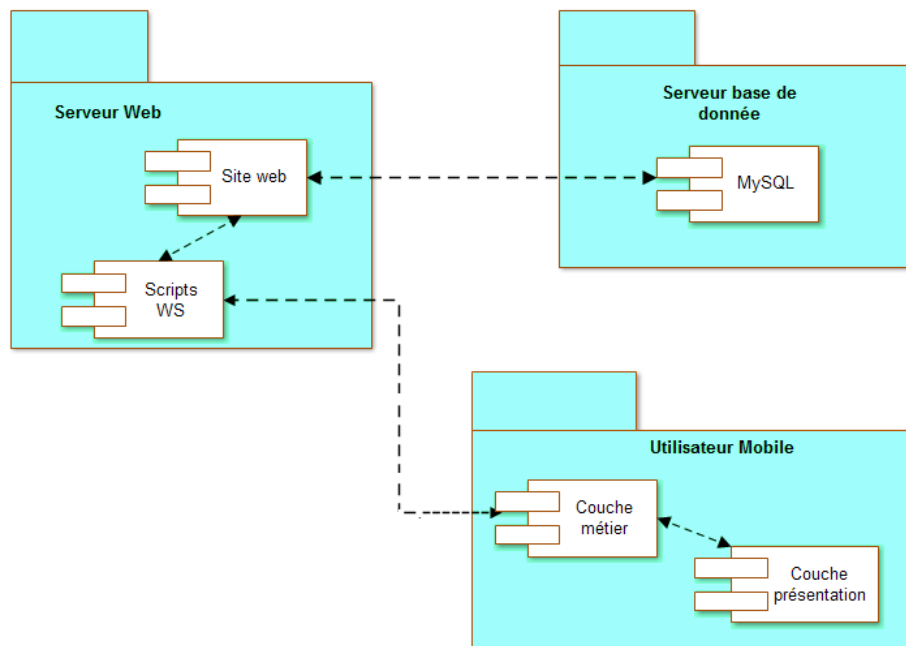


FIGURE 3.3 – Diagramme de composant

3.3.2 Diagramme de classe

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet, il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il exprime la structure statique du système en termes d'objet et de relations entre ses objets.

La figure 3.4 présente notre diagramme de classe qui montre qu'un utilisateur peut gérer ses annonces s'il possède tel que l'ajout, la modification et la suppression, un covoitureur en tant qu'un passager peut effectuer plusieurs réservations. De sa part le conducteur peut gérer ces réservations par l'acceptation ou le refus des passagers.

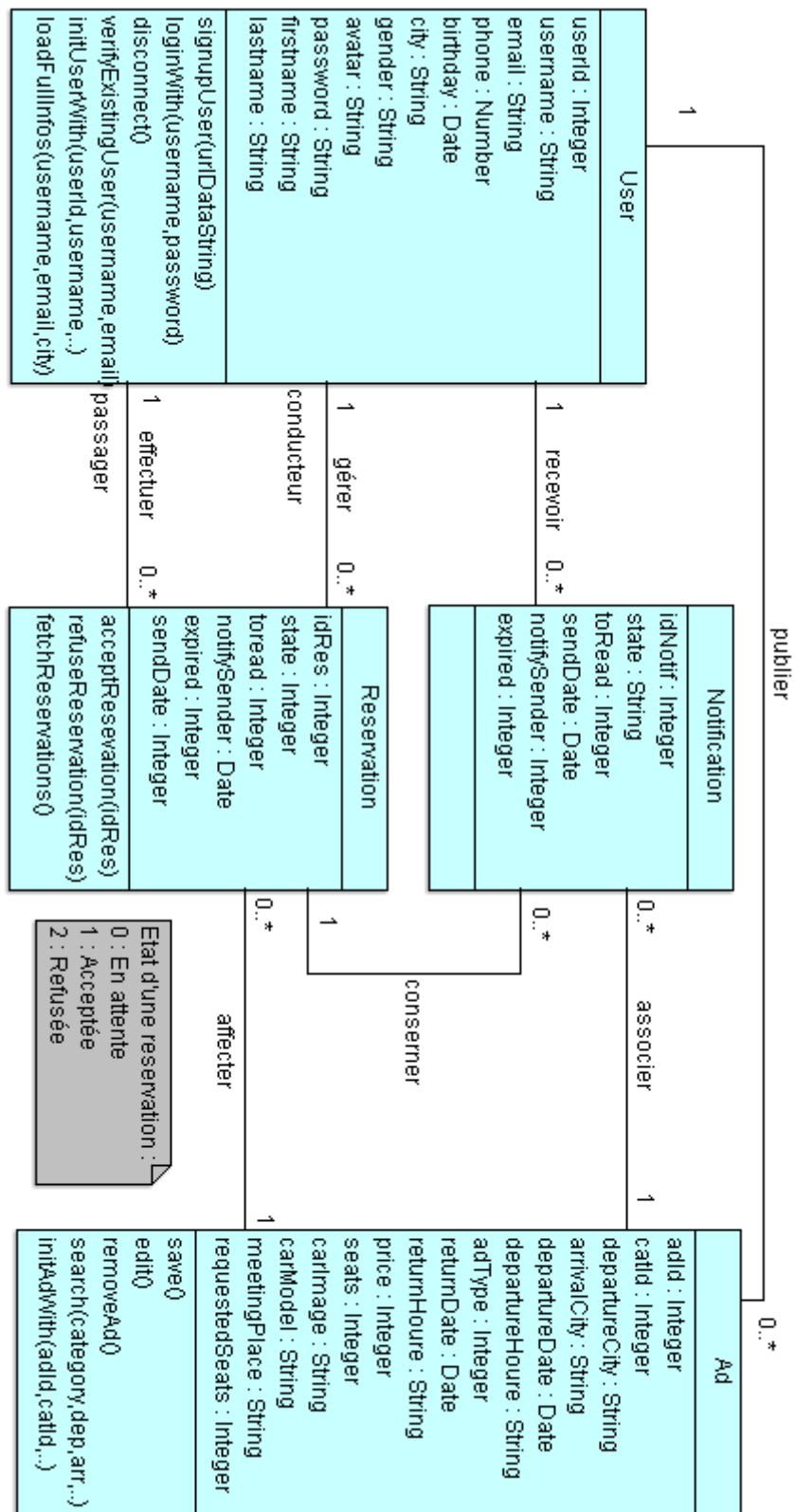


FIGURE 3.4 – Diagramme de classe

3.3.3 Diagrammes des séquences

Les diagrammes de séquences permettent de représenter des collaborations entre les objets selon un point de vue temporel, un diagramme de séquences peut servir à illustrer un cas d'utilisation.

La figure 3.5 présente le diagramme de séquence de recherche, l'utilisateur doit fournir les informations nécessaires pour trouver la liste des trajets qui satisfait les critères de recherche.

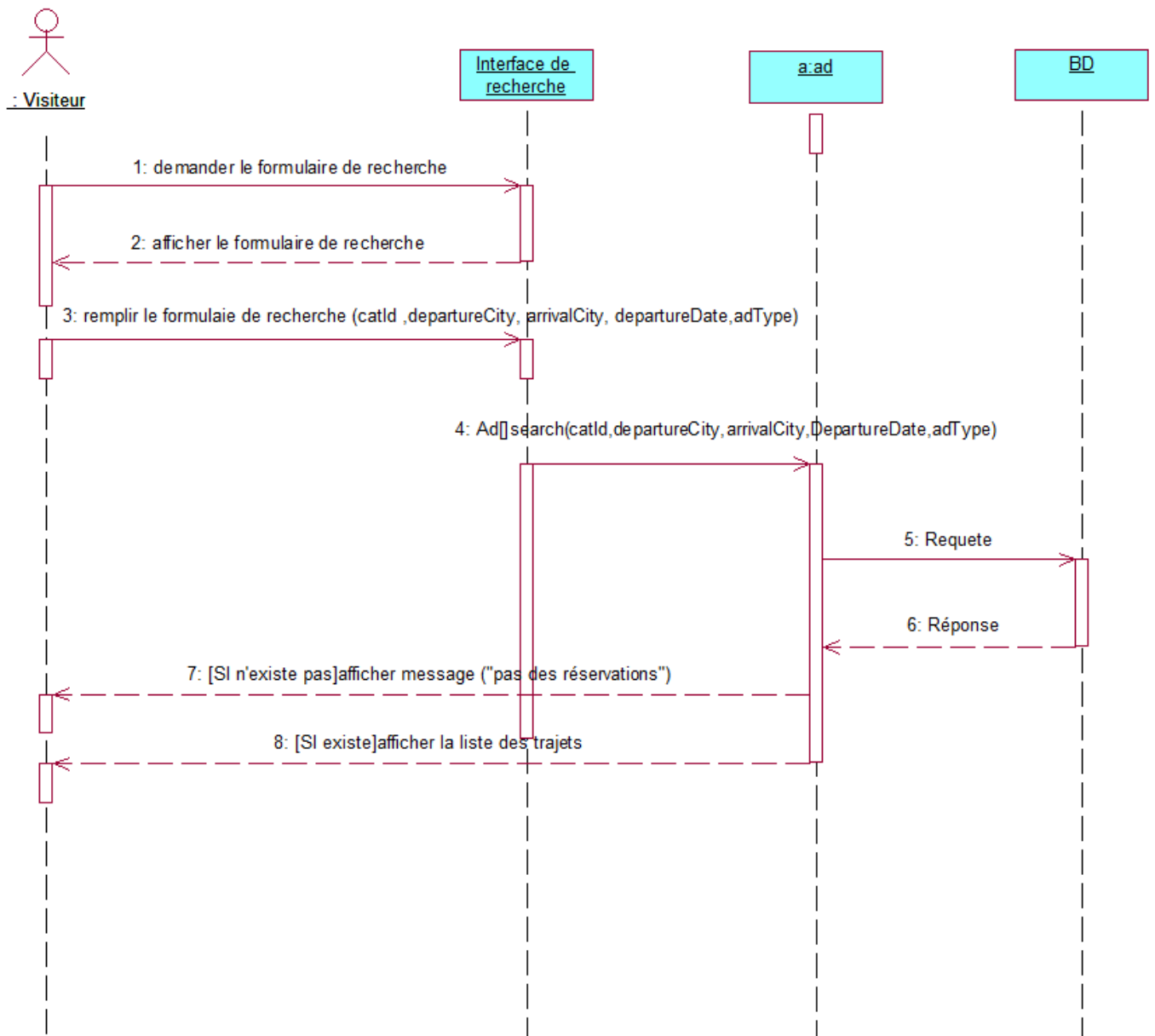


FIGURE 3.5 – Diagramme de séquence de cas d'utilisation "rechercher"

La figure 3.6 présente le scénario de l'inscription à l'application pour que l'utilisateur commence à l'utiliser et bénéficie de ses services.

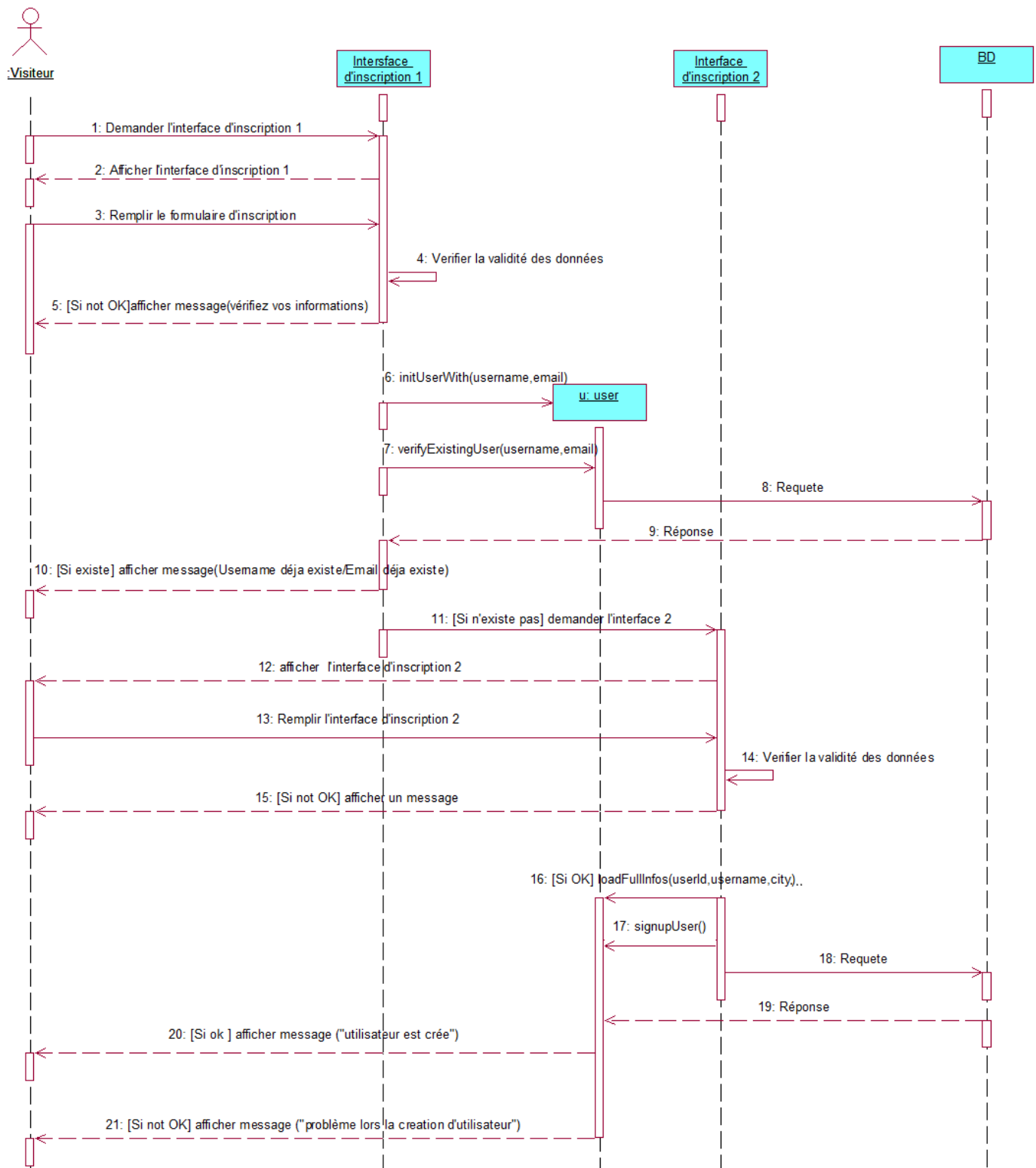


FIGURE 3.6 – Diagramme de séquence de cas d'utilisation "s'inscrire"

La figure 3.7 présente le diagramme de séquence de l'authentification, l'utilisateur doit saisir le nom d'utilisateur et le mot de passe pour accéder à l'application et se servir de ses fonctionnalités.

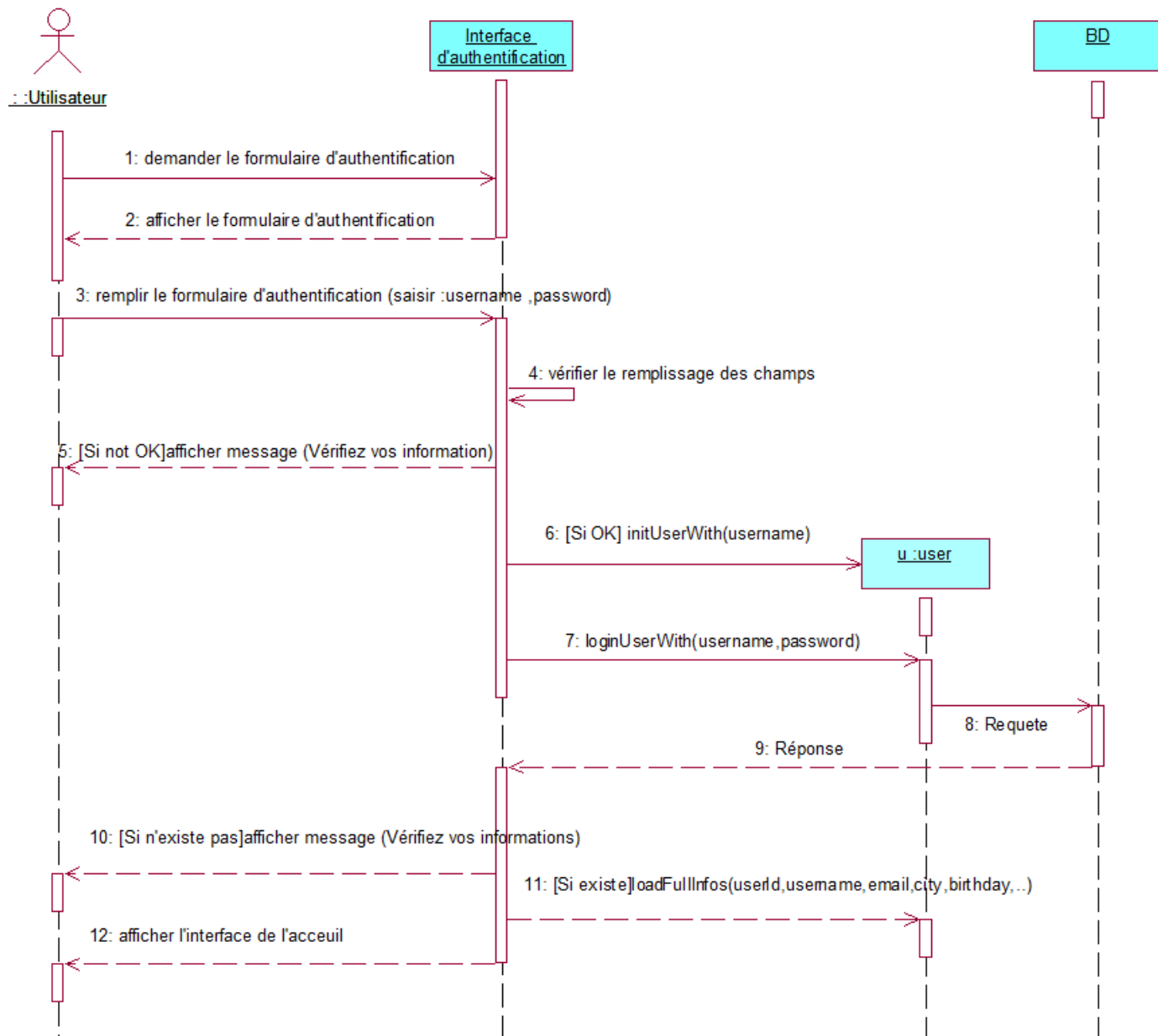


FIGURE 3.7 – Diagramme de séquence de cas d'utilisation "s'authentifier"

La figure 3.8 présente le diagramme de séquence de proposition d'un trajet, l'utilisateur doit indiquer le type de trajet, les nombres des places, la date, la ville de départ et la ville d'arrivée ainsi que d'autres informations selon la catégorie du trajet choisi (offre ou demande).

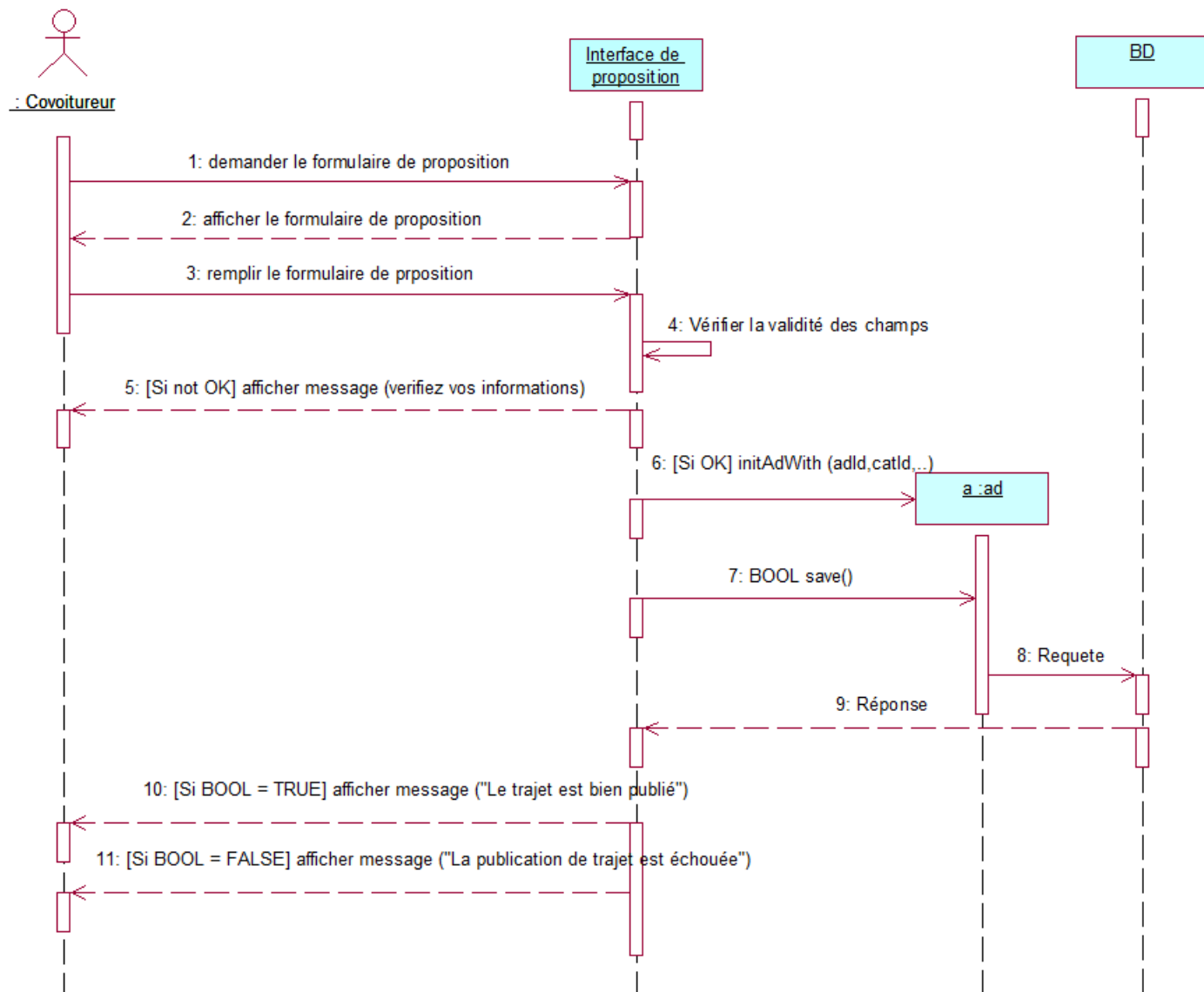


FIGURE 3.8 – Diagramme de séquence de cas d'utilisation "proposer un trajet"

La figure 3.9 présente le diagramme de séquence de la gestion des réservations.

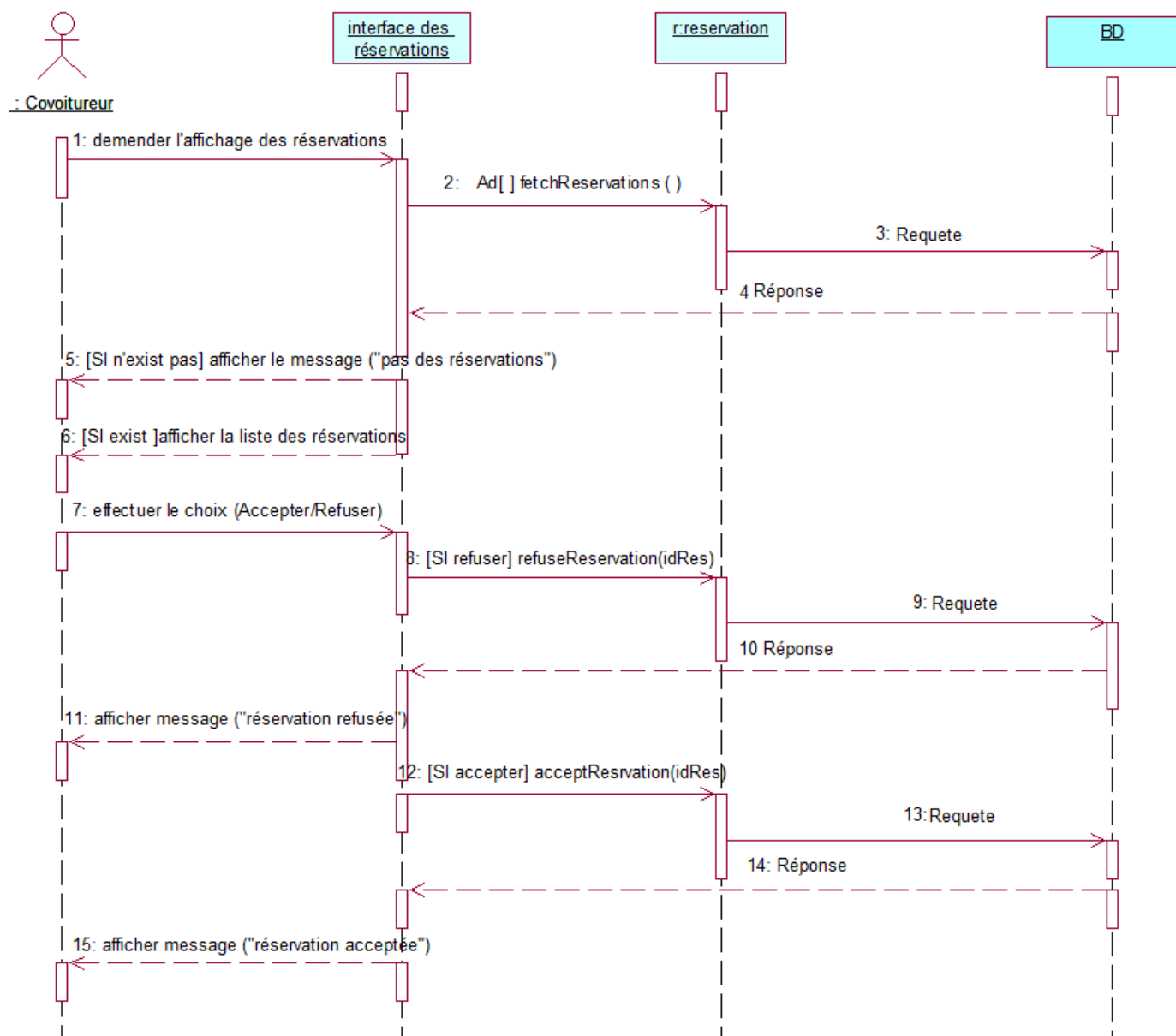


FIGURE 3.9 – Diagramme de séquence de cas d'utilisation "Gérer une réservation"

3.4 Conception graphique Maquette

Notre projet consiste à réaliser une application mobile iOS de covoiturage. Ce genre d'application doit contenir des interfaces qui doivent répondre à certains critères et doivent satisfaire le plus possible les besoins de l'utilisateur tout en gardant un esprit sérieux. Le graphisme et le design d'une application mobile (couleur, architecture, typographie, etc.) n'ont pas seulement un rôle esthétique mais ils doivent apporter aussi l'attention des utilisateurs sur les tâches, les boutons et les informations les plus importantes. Par ailleurs, la présentation des pages ne suffit pas. Il faut respecter les critères ergonomiques lors de la création de l'interface.

La figure 3.10 présente l'interface d'accueil de notre application, nous voulons permettre à l'utilisateur de pouvoir consulter tous les trajets de système.



FIGURE 3.10 – Interface d'accueil

Le clic sur l'un de ces trajets doit permettre à l'utilisateur de consulter le trajet avec plus des détails à l'aide de trois sections une pour le détail de trajet, l'autre pour les informations de covoitureurs et enfin une section consacrée pour les informations de la voiture dans le cas d'une offre.



FIGURE 3.11 – Interface d'un détail d'un trajet

Nous souhaitons à partir d'une carte localiser la ville de départ et d'arrivée comme l'indique la figure 3.12.

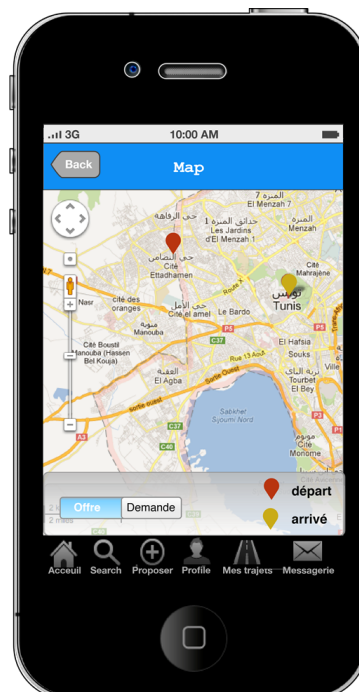


FIGURE 3.12 – Interface de la carte de géolocalisation

La figure de connexion présente l'interface qui doit permettre à l'utilisateur de s'authentifier au système avec un nom d'utilisateur et un mot de passe.



FIGURE 3.13 – Interface d'authentification

Notre expectation pour l'interface de recherche est comme l'indique la figure 3.14.



FIGURE 3.14 – Interface de recherche

La figure 3.15 présente le formulaire de proposition d'un trajet.



FIGURE 3.15 – Interface de proposition d'un trajet

Conclusion

Tout au long de ce chapitre, nous avons décrit le processus de conception de notre application de covoiturage. Nous l'avons accompli à travers les digrammes UML en l'occurrence avec le diagramme de composant, le diagramme de classe et les diagrammes de séquences qui nous ont permis de simplifier beaucoup l'abstrait et rendre les choses plus claires et plus compréhensibles. Ceci constitue une entrée essentielle dans le chapitre suivant dans lequel nous achevons notre travail.

Chapitre 4

Réalisation

Introduction

Ce chapitre a pour objet d'exposer le travail réalisé. Nous commençons par l'environnement de travail. Par la suite nous présentons les différentes technologies que nous avons adoptées pour la réalisation de notre application. Nous entamons après une description des différents outils ainsi que les composantes applicatives réalisées.

4.1 Environnement de développement

4.1.1 Environnement matériel

Pour la réalisation du projet, nous avons utilisé deux ordinateurs portables dans le développement ayant les caractéristiques suivantes :

- Processeur Intel Core i7 GHz.
- 8 Go de mémoire vive.
- Disque dur de capacité 500 Go.
- Systèmes d'exploitation : Windows 7 et OS X Mountain Lion version 10.8.

Pour tester notre application, nous avons utilisé nos appareils mobiles : un iPhone 4 version iOS 6.1.2 et un iPhone 4S version iOS 5.1.

4.1.2 Environnement logiciel

Conception :

ArgoUML : Pour élaborer les diagrammes des cas d'utilisation et de classe.

Rational rose : Pour tracer les diagrammes des séquences.

Les maquettes :

OmniGraffle : Pour concevoir les maquettes de l'application mobile.

Développent Web :

MAMP : Est une plateforme de développement Web sous Mac OS permettant de traiter et d'exécuter localement des requêtes HTTP et des scripts PHP. C'est un environnement comprenant deux serveurs Apache et MySQL, un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration des bases MySQL.

Développement mobile :

Xcode : Est un environnement de développement performant pour Mac OS X qui offre aux développeurs tous les outils dont ils ont besoin pour créer des applications pour Mac, iPhone et iPad.

Infographie et design :

Adobe Photoshop CS6 : Est un logiciel de traitement d'images et de retouche édité par adobe. Il est principalement utilisé pour le traitement de photographies numériques.

Rédaction du rapport :

TeXnicCenter : Est un logiciel libre sous la licence open source GNU GPL, qui se présente sous la forme d'un environnement de développement intégré pour le langage LaTeX sous Windows. Il est basé sur les distributions MiKTeX ou TeXLive.

4.2 Architecture de système

L'application mobile de covoiturage se connecte à un serveur de bases de données distant, afin d'insérer et récupérer des données .Ce qui nécessite aussi l'intégration d'un serveur web entre l'application client et le serveur de base de données.

D'où l'architecture trois-tires, elle est partagée entre :

Le client iOS ou web :

Conteneur d'application et demandeur de ressources.

Le serveur Web :

Vu que les données seront communiquées entre deux environnements hétérogènes, le rôle principal du serveur web est de gérer la communication entre le client iOS et le serveur de base de données à travers des scripts qui jouent le rôle de web services.

Le serveur de base de données :

Le serveur de base de données pour fournir les données au serveur web.

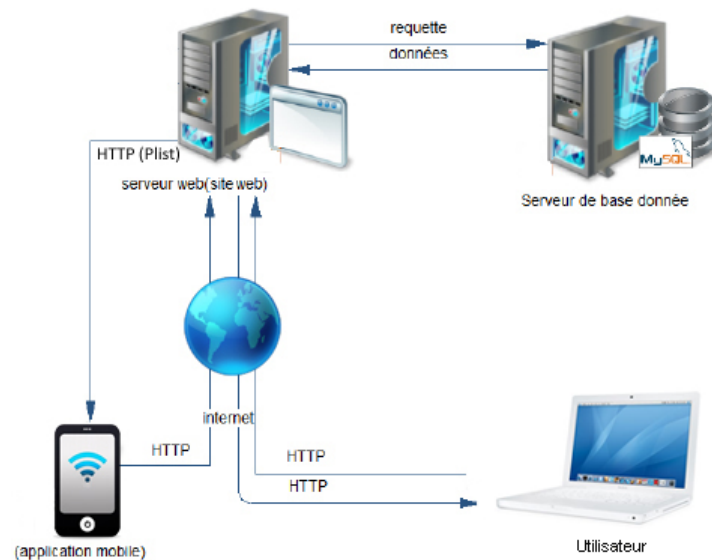


FIGURE 4.1 – Architecture matérielle de la solution proposée de notre système

4.3 Description du cycle de vie

Pour assurer une meilleure réalisation, nous avons abordé le modèle en spirale qui met l'accent sur l'activité d'analyse des risques. C'est le fait de contrôler au fur et à mesure les différents cycles du projet (analyse, spécification, conception). Ses principes fondamentaux sont :

- D'une part un découpage en modules qui suivent chacun son propre cycle de développement.
- D'autre part l'itération de cycles courts répétés (spécifications, développement, intégration) de manière indépendante sur chacune des modules.

Le modèle en spirale permet de mesurer l'avancement, et de maîtriser les régressions, tout particulièrement dans les incompatibilités qui pourraient apparaître entre les modules.

Ce modèle a également des aspects qu'on pourrait appeler « darwiniens », c'est-à-dire à tout instant un développeur peut donner naissance à une nouvelle version d'un module, qui pourra ou non remplacer la précédente selon qu'elle est considérée supérieure.

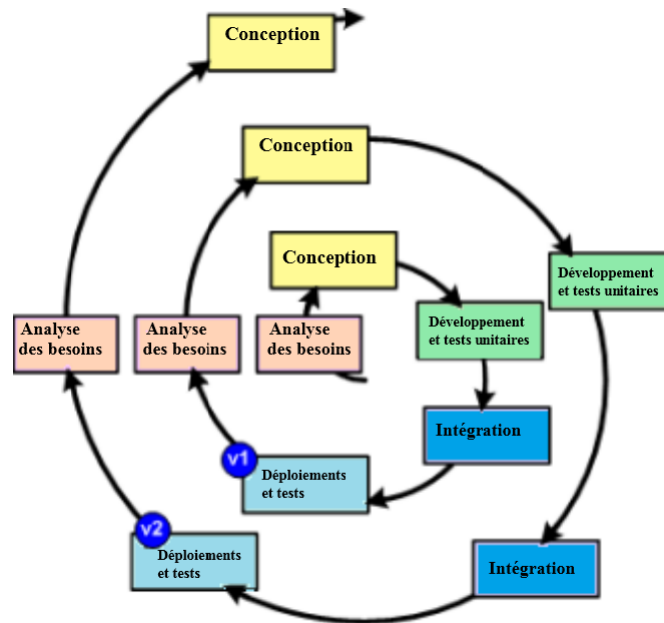


FIGURE 4.2 – Cycle de vie en spirale

4.4 Composition des modules

Au début du développement de site web et de l'application mobile, nous avons décomposé notre système en sept grands modules :

Inscription et connexion :

L'utilisateur peut se connecter ou créer un compte.

Profil d'utilisateur :

L'utilisateur peut consulter son profil et le gérer à partir de site web.

Gestion des trajets :

L'utilisateur peut ajouter, modifier ou bien supprimer un trajet.

Module de recherche des trajets :

L'utilisateur peut effectuer une recherche selon les critères de son choix.

Module de consultation des annonces :

L'utilisateur peut voir le détail de n'importe quel trajet, En plus dans l'application mobile il peut visualiser le trajet sur une carte.

Module de réservation :

Le système de réservation permet à l'utilisateur de réserver des trajets et gérer les réservations reçus par l'acceptation ou le refus.

Module de notification :

La notification permet d'avertir l'utilisateur des réponses du conducteur après la réservation d'un trajet.

La figure 4.3 présente les différents modules de notre système.

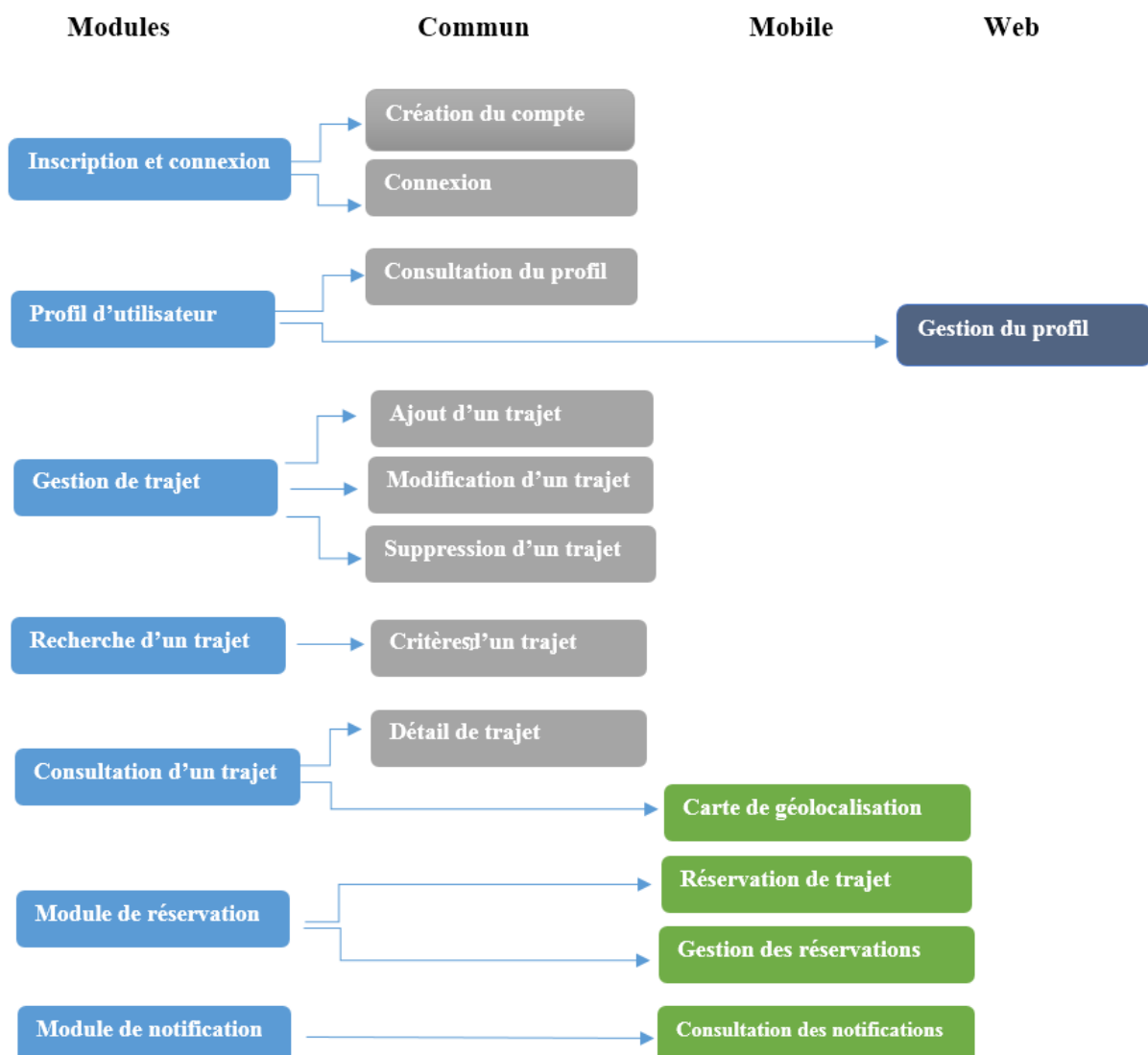


FIGURE 4.3 – Composition des modules

4.5 Réalisation de l'application mobile

4.5.1 Interface de lancement de l'application :

La figure 4.4 et 4.5 présentent l'interface principale de l'application qui contient une liste des trajets. Des informations basiques sur chaque trajet sont présentées dans cette interface, pour aider l'utilisateur à trouver rapidement le trajet désiré.

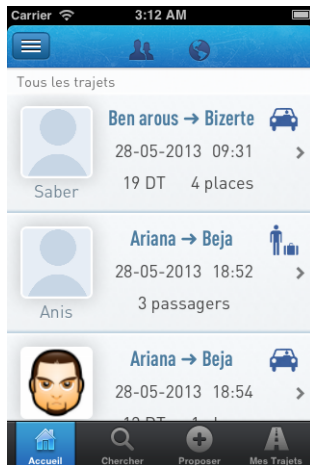


FIGURE 4.4 – Interface d'accueil sur un iphone 4

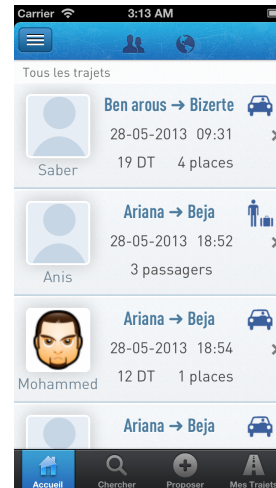


FIGURE 4.5 – Interface d'accueil sur un iphone 5

4.5.2 Interface de détail d'un trajet :

La figure 4.6 présente le détail d'un trajet sélectionné par l'utilisateur. Dans cette interface l'utilisateur peut consulter toutes les informations d'une annonce, en plus il peut contacter l'annonceur de trajet.



FIGURE 4.6 – Interface de détail d'un trajet

4.5.3 Interface de la carte de géolocalisation d'un trajet

L'utilisateur peut visualiser le trajet sur une carte dynamique, cette carte lui permet de visualiser rapidement les villes concernées par ce trajet. La ville de départ et la ville d'arrivée sont respectivement identifiées par un pin vert et un pin violet sur la carte.



FIGURE 4.7 – Interface de la carte dynamique

4.5.4 Interface de consultation de profil :

À partir de l'interface « Détail d'un trajet », en cliquant sur le bouton profil, L'utilisateur peut consulter le profil de l'annonceur, en plus il peut le contacter par téléphone ou par courrier.



FIGURE 4.8 – Interface de profil

4.5.5 Interface de recherche d'un trajet :

La figure 4.9 présente l'interface de recherche d'un trajet, l'utilisateur peut chercher une offre ou une demande suivant plusieurs critères.



FIGURE 4.9 – Interface de recherche

4.5.6 Interface de création d'un trajet :

La figure 4.10, 4.11 et 4.12 présentent le formulaire que l'utilisateur doit remplir pour publier son annonce. Les champs s'adaptent automatiquement selon la catégorie de trajet.



FIGURE 4.10 – Interface de proposition d'un trajet

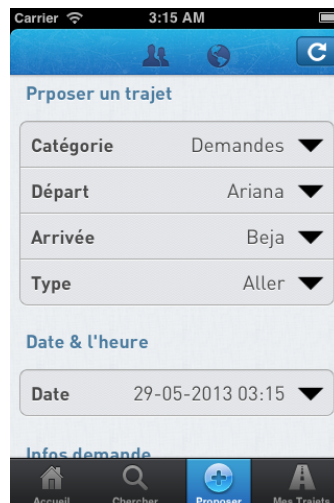


FIGURE 4.11 – Formulaire de proposition d'une demande

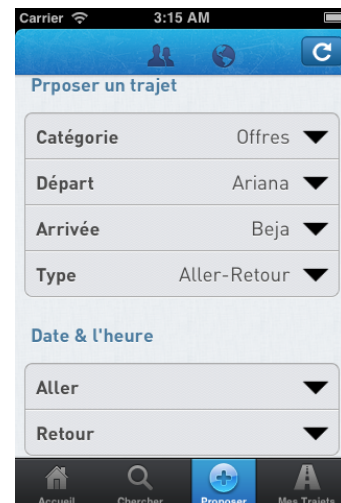


FIGURE 4.12 – Formulaire de proposition d'une offre

4.5.7 Interface de gestion d'un trajet :

À partir de l'interface de « Mes Trajets » l'utilisateur peut modifier les informations de ses propres annonces ou les supprimer de la liste des trajets.

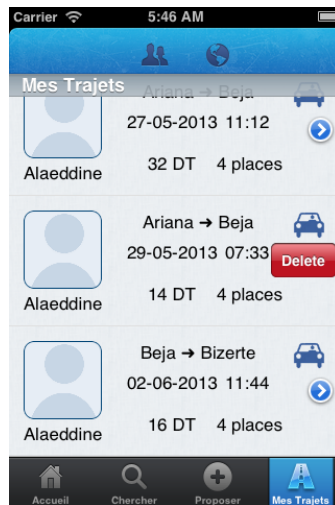


FIGURE 4.13 – Interface de gestion des trajets

4.5.8 Interface de connexion :

À travers un nom d'utilisateur et un mot de passe attribué à un profil déjà créé, l'utilisateur peut se servir de toutes les fonctionnalités offertes par l'application.



FIGURE 4.14 – Interface d'authentification

4.5.9 Interface d'inscription :

Les figures 4.15, 4.16, et 4.17 montrent qu'un nouveau utilisateur peut s'inscrire au système en remplissant les champs figurant dans ces interfaces. Il peut aussi choisir sa propre photo de profil.



FIGURE 4.15 – Formulaire d'inscription de l'étape un

FIGURE 4.16 – Formulaire d'inscription de l'étape deux

FIGURE 4.17 – Interface de choix de la photo de profil

4.5.10 Interfaces de réservation et de notification :

Les figures 4.18 et 4.19 présentent respectivement les interfaces des réservations et des notifications.



FIGURE 4.18 – Interface des réservations reçues

FIGURE 4.19 – Interface des notifications reçues

4.6 Réalisation de site web

La figure 4.20 présente l'interface d'accueil de site web, parmi ses fonctionnalités nous pouvons citer :

- Recherche d'un trajet
- Consultation d'un trajet
- Inscription
- Authentification
- Gestion d'un profil
- Publication d'un trajet
- Consultation d'un profil
- Mise en contact par courrier et messages privés

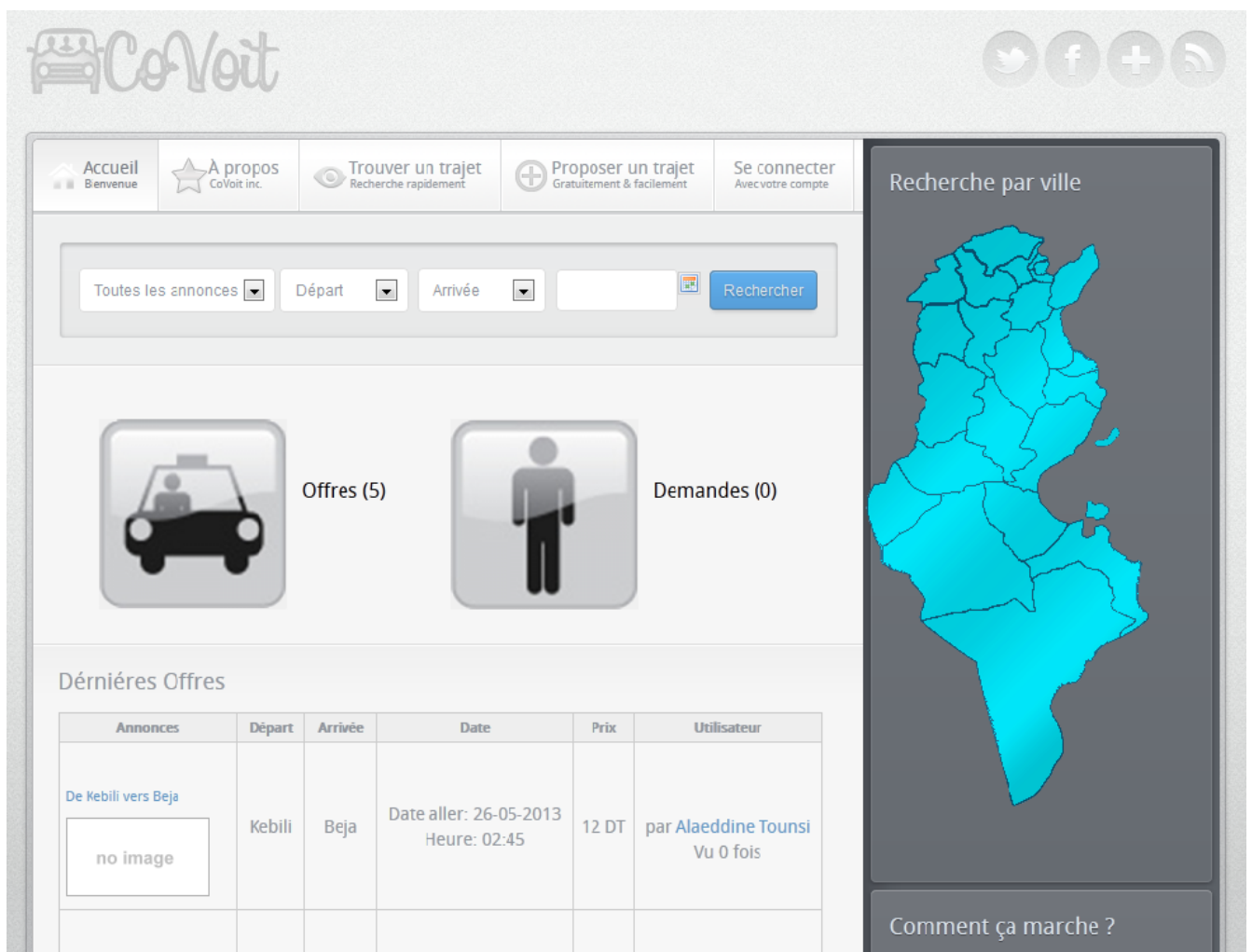


FIGURE 4.20 – Interface d'accueil de site web

La figure 4.21 présente le formulaire d'inscription qui permet aux visiteurs de fournir les informations nécessaires pour s'inscrire.

The registration form is titled "Inscription". At the top, it includes a legend: a yellow warning icon for "Champ obligatoire", a green checkmark for "Champ visible sur votre profil", a red 'X' for "Champ non visible dans le profil", and a blue 'i' for "Informations : pointez la souris sur l'icône". The form fields are as follows:

- Portrait du profil:** A dropdown menu with the option "Pas de portrait de profil".
- Nom:** A text input field.
- Prénom:** A text input field.
- Sexe:** Radio buttons for "Homme" and "Femme".
- Date de Naissance:** Three dropdown menus for day, month, and year, accompanied by a calendar icon.
- Adresse courriel:** A text input field.
- Téléphone:** A text input field.
- Ville:** A dropdown menu.
- Identifiant:** A text input field.
- Mot de passe:** A text input field.
- Vérification du mot de passe:** A text input field.

At the bottom of the form is a "S'inscrire" button. Below the button, the same legend as at the top is repeated.

FIGURE 4.21 – Formulaire d'inscription

La figure 4.22 présente le formulaire d'authentification. À travers un nom d'utilisateur et un mot de passe, l'utilisateur doit se connecter pour se servir de toutes les fonctionnalités offertes par le site web.

The login form is titled "Se connecter" with the subtitle "Avec votre compte". It contains the following elements:

- Identifiant:** A text input field.
- Mot de passe:** A text input field with masked characters (dots).
- Se souvenir de moi:** A checkbox.
- Connexion:** A blue button.
- Identifiant / mot de passe perdu ?** A link in blue text.
- S'inscrire** A link in blue text.

FIGURE 4.22 – Interface d'authentification

La figure 4.23 présente le formulaire de publication des trajets. Selon la catégorie qui peut être offre ou demande, un utilisateur doit remplir les informations concernant le trajet qui l'intéresse.

Merci de prendre connaissance des conditions d'utilisation avant de poster une annonce !

Ajouter

Rubrique

Départ ⓘ

Arrivée ⓘ

Type

Passagers

FIGURE 4.23 – Formulaire de publication d'un trajet

La figure 4.24 présente le profil d'un utilisateur, un covoitureur peut consulter son profil ou l'éditer, en plus il peut consulter et gérer ses trajets.

Page du profil de test

Clics: 1

Statut de connexion: Connecté(e)

Membre depuis: il y a 4 mois

Dernière connexion: Maintenant

Dernière mise à jour: il y a 51 jours

test

Sexe: Homme

Date de Naissance: 02/04/1962

Téléphone: 98874558

Ville: Ariana

FIGURE 4.24 – Profil d'un utilisateur

Conclusion

Dans la phase de la réalisation, nous avons développé les éléments constituant notre application tout en respectant les besoins aussi bien que la conception élaborée. Pour garantir la réussite de notre travail, l'application mobile a été testée et évaluée par des utilisateurs et des experts au sein de la société NITS. Nous pouvons constater que l'application a eu un résultat de satisfaction et de gratification par les plupart des testeurs grâce à sa simplicité d'utilisation et son ergonomie bien soignée.

Conclusion et perspectives

Bilan

Ce projet de fin d'étude, effectué au sein de la société NeatsShore IT Solution, consiste à développer un système de covoiturage. Ce projet est divisé en trois grandes parties qui sont l'analyse et la spécification des besoins, la conception et la réalisation de notre application avec ses deux parties mobile et Web. Nous avons opté pour le développement de l'application, un cycle de vie en spirale qui nous a permis de fournir régulièrement une nouvelle version de l'application en se rapprochant progressivement de l'application finale.

Perspectives

Après un travail acharné, nous avons mené l'application à un stade très satisfaisant. Afin de récompenser et concrétiser nos efforts, nous pourrions la lancer sur le marché. Ce travail a atteint ses objectifs, mais comme tout autre travail, il ne peut prétendre à la perfection. Comme perspectives de ce travail, nous comptons dans un premier temps de le perfectionner en lui rajoutant d'autres modules tels qu'un système de feedback qui permet l'évaluation du conducteur, de plus l'intégration d'un système de Push-Notification sera une priorité pour pouvoir recevoir les notification en temps réel.

Conclusion

En conclusion, ce projet de fin d'études a été une excellente expérience qui nous a permis de mettre en pratique les connaissances théoriques et les compétences acquises à la fin de notre formation à l'Institut Supérieur d' Informatique et de Mathématiques de Monastir. De même, nous avons eu une opportunité d'apprendre un nouveau environnement de développement en travaillant sur les téléphones mobiles iOS. Ceci nous a permis de maîtriser de nouveaux langages de programmation et de maîtriser le développement sous la plateforme iOS.

Nous terminons par souligner que ce projet a été une chance pour nous ouvrir sur le monde professionnel qui implique de la responsabilité, de la discipline ainsi que du travail collaboratif d'équipe. Ce projet nous a appris à surmonter les obstacles de différents types et respecter les délais malgré des multiples contraintes. Grace à ce projet, nous nous sommes initiés la vie professionnelle.

Bibliographie

- [1] Covoiturage. www.routard.com/guide_dossier/id_dp/55/ordre/2.htm.
- [2] Tunisia-today. www.tunisia-today.com/archives/56093.
- [3] Tunisiecovoiturage. www.tunisiecovoiturage.com.
- [4] Wikipedia. <https://fr.wikipedia.org/wiki/BlaBlaCar>.
- [5] Carpooling. www.carpooling.fr.
- [6] Covoiturage grand lyon. www.covoiturage-grandlyon.com.
- [7] Wikipedia. http://fr.wikipedia.org/wiki/IOS_Apple.
- [8] Wikipedia. <http://fr.wikipedia.org/wiki/Android>.
- [9] Wikipedia. <http://fr.wikipedia.org/wiki/WindowsPhone>.
- [10] Wikipedia. http://fr.wikipedia.org/wiki/Systeme_d'exploitation_mobile.
- [11] statcounter. http://gs.statcounter.com/#mobile_os-TN-monthly-201204-201304-bar.
- [12] Wikipedia. http://fr.wikipedia.org/wiki/IOS_Apple.
- [13] Wikipedia. http://fr.wikipedia.org/wiki/Unified_modeling_language.
- [14] David Gabay Joseph Gabay. *UML 2 analyse et conception*. Dunod, Paris, 2008.

Annexe A

Outils de développement

A.1 Rational Rose



FIGURE A.1 – Rational Rose

Rational Rose est un logiciel édité par l'entreprise Rational Machines pour créer et éditer les différents diagrammes du modèle UML (Unified Modeling Language) d'un logiciel. Rational Software a été vendu pour 21 milliards de dollars à IBM le 20 février 2003. Rational Rose permet également de sauvegarder et d'imprimer ces diagrammes, ainsi que de générer le code source Java ou C++ qui leur correspondent.

A.2 ArgoUML



FIGURE A.2 – ArgoUML

ArgoUML est le principal outil de modélisation UML libre et inclut le support pour tous les standards diagrammes UML 1.4. Il fonctionne sur n'importe quelle plateforme Java et disponible en dix langues. ArgoUML supporte plusieurs types de diagrammes comme le diagramme de cas d'utilisation, de classes, de séquence, d'état, de collaboration, d'activité et de déploiement.

A.3 MySQL



FIGURE A.3 – MySQL

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multiutilisateurs. C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire.

A.4 MAMP



FIGURE A.4 – MAMP

MAMP est un acronyme informatique signifiant :

- « Macintosh » (Système d'exploitation)
- « Apache » (Serveur Web frontal, il répond directement à l'internaute via le navigateur)
- « MySQL » (Ce Serveur gère le Système de Gestion de Base Données, il stocke toutes les données de l'application)
- « PHP » à l'origine « Perl », et « Python » (Langages de Programmation est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur)

C'est une application reposant sur une plateforme de développement Web : Apache + PHP + MySQL. MAMP vous permet d'installer un serveur local sur votre ordinateur, ce qui a pour fonction de pouvoir travailler directement dessus sans avoir besoin de se connecter à un serveur distant pour la mise en ligne de votre site. Vous devez l'installer sur votre Mac pour faire fonctionner sur votre Système Intégré de Gestion de Bibliothèque ou Sigh, un PMB équivalent d'EasyPhp pour PC.

A.5 Joomla



FIGURE A.5 – CMS : Joomla

Joomla est un système de gestion de contenu (en anglais, CMS, Content Management System). Et un CMS, c'est quoi ? Pour faire simple, un CMS est un logiciel web qui permet de créer des sites web dynamiques en toute simplicité. Joomla est un CMS Open Source distribué sous licence GNU/GPL (gratuit) avec lequel on peut mettre en ligne du contenu et mettre à la disposition des visiteurs des services (forum, boutique en ligne, galerie photos, etc.), l'avantage de Joomla c'est qu'il ne demande pas des connaissances techniques particulières. Joomla est reconnu comme l'un des plus puissants CMS Open Source, il est aussi le système de gestion de contenu le plus téléchargé au monde.



FIGURE A.6 – Qui utilise Joomla ?

A.6 AdsManager



FIGURE A.7 – AdsManager

AdsManager est un composant de référence pour gérer un site de petites annonces, offrant à vos utilisateurs un moyen de diffusion.

- Pour les utilisateurs :
 - Offrir à vos utilisateurs un moyen de diffuser leurs annonces sur votre site
- Pour l'administrateur :
 - Ajouter/Supprimer des champs
 - Configuration par catégorie
 - Modération, Expiration, Contact

Ses points Forts :

Sa simplicité d'utilisation :

Contrairement à un CCK, Adsmanager est un composant spécifique répondant à un problème. Par ce fait, vous éviterez de longues heures de configuration.

Ses options :

Permet de s'adapter à tout type de petites annonces : immobilier, voiture, puériculture, annuaire,etc.

A.7 OmniGraffle



FIGURE A.8 – OmniGraffle

OmniGraffle est un logiciel de création de diagramme pour Mac OS X développé par Omni Group . Dans beaucoup d'aspects, OmniGraffle est similaire à Microsoft Office Visio, de plus la version professionnelle d'OmniGraffle peut aussi bien importer qu'exporter des documents Visio. Il peut être utilisé pour réaliser des diagrammes simples, des organigrammes ou des illustrations. Il fonctionne grâce à une interface WYSIWYG tout en glissant-déposant. Des palettes regroupant des motifs sont disponibles sous la forme d'extensions et les utilisateurs peuvent également créer leurs propres palettes. OmniGraffle tire entièrement profit de la couche graphique de Mac OS X plus connue sous le nom d'Aqua, et bénéficie de l'anticrênelage, déformations lissées, ombres portées transparence. La dernière version d'OmniGraffle ajoute le support des courbes de Bézier et la construction de graphiques hiérarchiques à partir d'éléments de texte. OmniGraffle permet d'organiser l'information avec des techniques intuitives, aussi bien qu'avec un gestionnaire de style avancé, permettant d'organiser très facilement des prototypes de pages web ou des documents PDF, aussi bien et aussi rapidement que des schémas ou des croquis.

A.8 Xcode



FIGURE A.9 – Xcode

Xcode est un environnement de développement performant pour Mac OS X qui offre aux développeurs tous les outils dont ils ont besoin pour créer des applications pour Mac, iPhone et iPad. Carbon permet de programmer avec les langages suivants :

- C
- CC++

Cocoa permet de programmer avec les langages suivants :

- Objective-C (version 2.0 à partir de Xcode 3.0)
- AppleScript (à travers l'interface AppleScript Studio)
- Java

Fourni avec toute une suite logicielle (graphiques, audio, etc.) pour développeurs et programmeurs, il permet de créer des logiciels utilisant toutes les fonctionnalités, la puissance et la stabilité de Mac OS X et d'UNIX.

Cet environnement peut être obtenu gratuitement sur le Mac App Store. Il était fourni en standard avec chaque Mac, sur les disques d'installation de Mac OS X 10.6, Mac OS X 10.5, 10.4 et 10.3. Il n'est pas préinstallé sur l'ordinateur et doit être installé séparément.

La dernière version Xcode bénéficie d'une refonte rendant le logiciel plus simple d'utilisation et facilitant l'édition d'applications. L'interface a subi également quelques modifications avec le regroupement de toutes les rubriques sur une unique fenêtre : le codage, les tests et le débogage. Les erreurs de syntaxe et de logique seront immédiatement identifiées et corrigées par le logiciel. Xcode est accompagné du Xcode IDE, d'Instruments et d'iOS Simulator.

A.9 Photoshop



FIGURE A.10 – Photoshop

Photoshop est un logiciel de traitement d'images et de retouche édité par adobe. Il est principalement utilisé pour le traitement de photographies numériques. Reconnu aussi par les infographistes professionnels à travers sa puissante galerie de filtres et d'outils graphiques performants, son utilisation est maintenant enseignée dans les plus grandes écoles, instituts des beaux-arts et il est utilisé par une grande majorité des studios et agences de création.

A.10 TeXnicCenter



FIGURE A.11 – TeXnicCenter

TeXnicCenter est un logiciel libre sous la licence open source GNU GPL, qui se présente sous la forme d'un environnement de développement intégré pour le langage LaTeX sous Windows. Il est basé sur les distributions MiKTeX ou TeXLive. Il permet à l'utilisateur d'éditer un document LaTeX et de le compiler dans les formats PDF, DVI ou PS. TeXnicCenter possède un menu qui permet un accès simple et rapide aux éléments usuels (formules, symboles, sections). Il permet également la création et l'organisation des projets, l'accéder aux sections et aux environnements des documents, et l'insertion d'une bibliographie (avec BibTeX) ainsi qu'un index (avec MakeIndex).