



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Application cross-plateforme de gestion du
parc automobile universitaire
(ParkOptimizer)

Réalisé par :

- LACHACHI Mohamed Charaf Eddine

Présenté le 26 Mai 2015 devant la commission d'examination composée de :

- Benmansour F. (Encadreur)
- Brikci A. (Examineur)
- Benmaamar B. (Examineur)

Remerciements

La réalisation de ce mémoire a été possible grâce à dieu le tout puissant, grâce à plusieurs personnes à qui je voudrais témoigner toute mes reconnaissance.

Tout d'abord, j'adresse ma gratitude à mes très chers parents pour leurs patiences et leurs soutiens tout au long de ma vie.

Je remercie mes frères Djalal et Omar, et ma sœur Sara ainsi que mon cousin Walid pour leurs aides et encouragements.

Je remercie monsieur BENGUELLA le Doyen de la faculté des sciences pour son honorable présence.

Je remercie aussi madame BENMANSOUR pour son aide, sa disponibilité, sa confiance, et ses conseils judicieux, qui ont contribué à la finalisation de ce mémoire.

J'exprime aussi toute ma reconnaissance, à tous mes amis, mes collègues qui m'ont soutenu moralement.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Table des matières

INTRODUCTION	3
CHAPITRE I : GENERALITES SUR LES SYSTEMES D'EXPLOITATION ET L'ENVIRONNEMENT D'EXECUTION	5
1. INTRODUCTION.....	6
2. SYSTEME D'EXPLOITATION.....	6
2.1. ANDROID	6
2.1.1. DEFINITION DU SYSTEME.....	6
2.1.2. INTERFACE.....	6
2.1.3. VERSION.....	8
2.2. WINDOWS PHONE.....	8
2.2.1. DEFINITION DU SYSTEME.....	8
2.2.2. INTERFACE.....	9
2.3. IOS.....	10
2.3.1. DEFINITION DU SYSTEME.....	10
2.3.2. INTERFACE.....	10
3. ENVIRONNEMENT D'EXECUTION.....	11
3.1. VUE D'ENSEMBLE DU .NET FRAMEWORK.....	12
3.1.1. FONCTIONNALITES DU COMMON LANGUAGE RUNTIME.....	14
4. CONCLUSION	16
CHAPITRE II : MODELISATION DU SYSTEME.....	17
1. INTRODUCTION.....	18
2. MERISE.....	18
2.1. MODELISATION	18
2.1.1. MODELE CONCEPTUEL DE DONNEES (MCD).....	19
2.1.2. MODEL LOGIQUE DE DONNEES(MLD).....	19
3. ARCHITECTURE DU SYSTEME.....	20
3.1. GRAPHE DE DEPENDANCE FONCTIONNEL	20
3.2. MVVM PATTERN.....	21
3.3. DATA API.....	22
4. CONCLUSION	23
CHAPITRE III: REALISATION DU SYSTEME.....	24
1. INTRODUCTION.....	25
2. ENVIRONNEMENTS DE DEVELOPPEMENT (EDI)	25
2.1. VISUAL STUDIO.....	25
2.2. XAMARIN STUDIO	25

3. LES LANGAGES DE PROGRAMMATION UTILISES	26
3.1. LE C# (C SHARP)	26
3.2. LE XAML	27
3.3. SGBD	28
3.3.1. CARACTERISTIQUE DE MYSQL	28
4. DESCRIPTION DE L'APPLICATION	28
4.1. APPLICATION DESKTOP	29
4.2. APPLICATIONS MOBILES	31
4.2.1. ANDROID ET IOS	32
4.2.2. WINDOWS PHONE	33
5. CONCLUSION	33
CONCLUSION GENERALE	34
BIBLIOGRAPHIE	36
LISTE DES FIGURES	37
LISTE DES ABREVIATIONS	38

Introduction

Avec l'évolution fulgurante et continue de la technologie l'outil informatique est devenu indispensable dans la vie quotidienne de l'individu dans pratiquement toutes les tâches qu'il entreprend, que ça soit sur le plan personnel ou professionnel au niveau des administrations et institutions, notamment avec la facilité et le gain de temps considérable qui en résulte.

L'adaptation avec ce changement et l'informatisation de tous les secteurs (administration, sécurité) est devenue une nécessité. Dans cette optique, il y'a une volonté de notre part d'être une partie active à intégrer l'université de Tlemcen dans cette évolution technologique par l'amélioration de ses services sur différents plans notamment celui de l'administration, Etant étudiant au département de l'informatique on a pris l'initiative d'améliorer le rendement de l'administration universitaire par l'introduction de cet outil informatique.

A partir de ce constat on vise à faciliter la tâche des agents de sécurité de l'université de Tlemcen dans le contrôle du flux automobile par l'informatisation de leur système de travail afin de garantir un contrôle fiable et optimal et par conséquent la sécurité du campus universitaire.

Pour cela on a pensé à développer un system cross-plateforme (Android, Windows-phone, iOS) pour but d'améliorer la gestion des dossiers d'inscriptions des véhicules circulant au sein du campus universitaire et créer une coordination entre les services administratifs et les agents de sécurité sur le terrain.

La Problématique

Ce travail consiste à contrer une réalité non commode vécue sur le terrain dont les lacunes sont nombreuses, à savoir :

- Le désordre dans la gestion du parc automobile de l'université ;
- La lenteur dans l'identification des véhicules ;
- Le caractère manuel dans le travail de l'administration ;
- L'absence de coordination entre les agents de sécurité sur le terrain et l'administration ;
- Aucune solution numérique n'a été proposée auparavant dans ce cadre.

A partir de ces lacunes, on a conçu notre travail pour y remédier et proposer une solution.

Structure du mémoire

Notre mémoire comprend trois chapitres :

- 1) Généralités sur les systèmes d'exploitation et l'environnement d'exécution : Dans ce chapitre on va définir les différents systèmes d'exploitation et le .NET ou l'environnement d'exécution.
- 2) Analyse et conception : on se focalisera dans ce chapitre sur la modélisation de notre système en expliquant les différentes démarches entreprises afin de réaliser notre projet
- 3) Réalisation : est la partie dédiée au volet pratique de notre recherche par la présentation de notre application et l'explication de ses différentes fonctionnalités.

Chapitre I : Généralités sur les systèmes d'exploitation et l'environnement d'exécution

1. Introduction

Notre travail s'est fait sur des plateformes différentes et à chacune ses caractéristiques, et cela nous a conduit vers une nécessité de mettre en lumière les différents systèmes d'exploitations par des définitions et des précisions sur les interfaces et les divers versions, et d'autre part on se focalisera aussi sur l'environnement d'exécution en donnant plus de détails sur plusieurs point tel que le .NET Framework et le Common Language Runtime.

2. Système d'exploitation

Le projet consiste à créer une application multi-plateforme ce qui veut dire une application fonctionnant sous plusieurs systèmes d'exploitation. D'où l'intérêt à les citer.

2.1. ANDROID

Android est l'un des systèmes d'exploitation les plus connus et les plus utilisés actuellement et dans cette partie on mettra la lumière sur plusieurs points liés à ce système.

2.1.1. Définition du système

Android est un système d'exploitation ouvert (Open Source) pour tablettes tactiles, terminaux mobiles et TV connectées. Il a été conçu en 2007, par la société Android, une start-up puis rachetée par Google.

Android est fondé sur un noyau Linux disponible grâce à une licence Apache, Android inclut tous les utilitaires requis par un constructeur pour le mettre en œuvre dans un téléphone portable. Il est proposé à tous les fabricants de téléphones mobiles, pour faciliter son adoption.

Cependant, Android est modifiable par les constructeurs. Donc, les versions varient largement d'un constructeur à l'autre. Il est possible de télécharger des applications gratuites ou payantes sur Play Store. [1]

2.1.2. Interface

Les appareils équipés d'Android affichent un écran d'accueil qui sert de point de départ à partir duquel l'utilisateur peut accéder aux applications. Le contenu de cet écran peut être librement personnalisé par l'utilisateur et est souvent personnalisé par les fabricants; il se comporte comme une pile de feuilles qui peuvent être glissées l'une sur l'autre avec le doigt. En

haut de l'écran d'accueil se trouvent différentes icônes d'avertissement et d'état qui permettent par exemple de contrôler le niveau de la batterie et la disponibilité des réseaux.

Depuis cet écran d'accueil, une icône permet à l'utilisateur d'accéder à l'écran de *launcher* : sur cet écran est affichée la liste de toutes les applications installées dans l'appareil, que l'utilisateur pourra éventuellement ajouter à l'écran d'accueil. Lorsqu'une application est exécutée, l'écran de l'application sera affiché en lieu et place de l'écran d'accueil et un bouton permet à l'utilisateur d'y revenir.

Les gestes reconnus par l'écran tactile des appareils Android sont : toucher l'écran, enfoncer (toucher et maintenir le doigt en contact avec l'écran), déplacer (enfoncer, puis déplacer le doigt en restant en contact avec l'écran), glisser (déplacer le doigt en contact avec l'écran sans s'arrêter), double frappe (toucher deux fois de suite un dessin à l'écran dans un délai très court), pincer (mettre deux doigts en contact avec l'écran, puis les rapprocher) et tourner l'écran (changer la position du téléphone, le poser sur le dessus ou sur le côté)



Figure 1.1: Interface Android

2.1.3. Version

Il existe plusieurs versions d'Android. Dans notre projet, on a opté pour la version 4.4 (KITKAT).

Le système d'exploitation Android 4.4 kitKAt utilise des technologies très avancées pour l'optimisation de la mémoire d'où sa disponibilité sur les appareils. Il peut fonctionner avec au minimum 512 Mo de RAM. Cette caractéristique est importante car les versions précédentes demandaient plus de mémoire interne ce qui les rendaient incompatibles avec plusieurs anciens modèles d'appareils. Ce problème est connu sous le nom d'Android fragmentation.

Android 4.4 KitKAt présente plusieurs caractéristiques : une interface plus propre et dégagée, un support NFC pour communication en champ proche (Near field communication) et toujours un écran tactile complet qui remplace le besoin d'avoir des boutons physiques existants sur des appareils Android.

Android 4.4 KitKat vante la sécurité renforcée basée sur le module linux qui aide à prévenir l'accès non autorisé des applications.

Google a annoncé le lancement d'Android 4.4 KitKat en septembre 2013 et son début était sur le smartphone Nexus 5. Les observateurs s'attendaient à ce que l'OS soit appelé Key Lime Pie, alors que Google a choisi le nom KitKat dans le cadre d'une promotion marketing avec la Hershey Company qui produit des barres de chocolat appelée Kit Kat.

2.2. WINDOWS PHONE

Tout comme le système d'exploitation cité précédemment, Windows phone est largement utilisé et connu dans le milieu des OS.

2.2.1. Définition du système

Windows Phone est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, sa précédente plateforme logicielle qui a été renommée pour l'occasion en Windows Phone Classic.

Contrairement au système qu'il remplace, Windows Phone est d'abord 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.

Selon Microsoft, le développement du système d'exploitation est parti d'une feuille blanche, et s'est terminé entre fin août et début septembre 2010 avec notamment la disponibilité du kit de développement final et la diffusion d'un kit de présentation aux constructeurs. Contrairement à Windows Mobile, Windows Phone devient en mai 2013, le 3ème système d'exploitation mobile.

Windows Phone se caractérise par une capacité à se connecter à de nombreuses plateformes externes comme SkyDrive, les messageries (Outlook, Yahoo, etc.), Facebook et d'autres. De plus le hub Microsoft Office regroupe des versions adaptées de : Word, Excel, PowerPoint, OneNote et de SharePoint. Cette suite Microsoft Office Mobile est compatible avec la version pour ordinateurs de bureau, Microsoft Office.

2.2.2. Interface

Contrairement à Windows Mobile, l'interface homme-machine de Windows Phone repose nativement sur l'utilisation d'un écran tactile multipoints. Avec Windows Phone, Microsoft propose une interface utilisateur dénommée Modern UI avec un système de tuiles dynamiques, très différente de ce que l'on peut avoir l'habitude avec IOS ou Android.

Windows Phone possède une interface précédemment nommée « Metro », qui reprend celle du Zune HD. Metro se caractérise par une interface minimaliste qui met en avant la typographie et l'iconographie.



Figure 1.2 Interface Windows Phone 8.1

2.3. IOS

2.3.1. Définition du système

IOS à l'origine iPhone Os est un système d'exploitation mobile créé et développé par Apple et distribué exclusivement pour le matériel Apple. Il est utilisé actuellement sur l'iPhone, iPod et iPad.

IOS est dérivé de l'OS X dont il partage les fondations (le noyau hybride XNU basé sur le micro noyau Mach). IOS comporte quatre couches d'abstraction similaire à celle de Mac OS une couche « Core OS », une couche « Core Services », une couche « Media » et une couche «Cocoa». Le système d'exploitation occupe au maximum 3 Go de la capacité mémoire totale de l'appareil, selon l'appareil. [2]

2.3.2. Interface

L'interface du système d'exploitation iOS est fondée sur le concept de la manipulation par contact tactile de l'écran, incluant notamment la technologie Multi-touch, permettant de reconnaître des gestes à plusieurs doigts simultanés, et un accéléromètre détectant les mouvements de l'appareil.

La page d'accueil nommée springboard est composée d'une mosaïque d'icônes (4x4 icônes pour les appareils jusqu'à l'iPhone 4S, 5x4 pour l'iPad et l'iPhone 5, 6x4 pour l'iPhone 6 et 6 Plus) représentant l'ensemble des applications disponibles en mémoire ainsi qu'un dock de quatre icônes (6 pour les iPad) dans la partie inférieure de l'écran. Ce dernier permet à l'utilisateur d'accéder à ses applications favorites. En effet, il est possible de créer jusqu'à onze pages d'applications dans la partie supérieure, tandis que le dock reste visible quelle que soit la page actuelle. Pour passer de l'une à l'autre, il faut faire glisser son doigt vers la droite ou la gauche de l'écran. Le retour à la page d'applications la plus à gauche s'effectue par une pression sur le bouton « Home ». La page d'accueil courante est accessible au démarrage ou en appuyant sur le bouton « Home » de l'iPhone ou iPad, depuis une application



Figure 1.3: Interface IOS 8

3. Environnement d'exécution

Dans notre projet, on s'est basé sur le principe de cross plateforme. Dans ce cas, deux méthodes sont possibles :

- la première est de coder un programme pour chaque plateforme ce qui revient à dire que le nombre de projet sera égale au nombre de plateformes, ce qui est très redondant.
- La deuxième est de songer à une technologie qui s'occupe de virtualiser le code et le compiler suivant un Framework bien définie d'où vient le besoin de la technologie dot NET. Qu'on expliquera ci-dessous.

3.1. Vue d'ensemble du .NET Framework

Le .NET Framework est une technologie qui prend en charge la création et l'exécution de la nouvelle génération d'applications et de services Web XML. Le .NET Framework est conçu pour remplir les objectifs suivants :

- Fournir un environnement cohérent de programmation orientée objet que le code objet soit stocké et exécuté localement, exécuté localement mais distribué sur Internet ou exécuté à distance.
- Fournir un environnement d'exécution de code qui minimise le déploiement de logiciel et de conflits de version.
- Fournir un environnement d'exécution de code qui promet l'exécution sécurisée de code y compris le code créé par un tiers d'un niveau de confiance moyen ou un tiers inconnu.
- Fournir un environnement d'exécution de code qui élimine les problèmes de performance des environnements interprétés ou écrits en scripts.
- Fournir au développeur un environnement cohérent entre une grande variété de types d'applications comme les applications Windows et les applications Web.
- Générer toutes les communications à partir des normes d'industries pour s'assurer que le code basé sur le .NET Framework peut s'intégrer à n'importe quel autre code.

Le .NET Framework se compose du Common Language Runtime et de la bibliothèque de classes .NET Framework. Le Common Language Runtime est la base du .Net Framework. Le runtime peut être considéré comme un agent qui manage le code au moment de l'exécution, fournit des services essentiels comme la gestion de la mémoire, la gestion des threads et la communication à distance. Il applique également une stricte sécurité des types et d'autres formes d'exactitude du code qui promeuvent un code sécurisé et robuste. En fait, le concept de gestion de code est un principe fondamental du runtime. Le code qui cible le runtime porte le nom de code managé, tandis que le code qui ne cible pas le runtime porte le nom de code non managé. La bibliothèque de classes est une collection complète orientée objet de types réutilisables que vous pouvez utiliser pour développer des applications allant des traditionnelles applications en ligne de

commande ou à interface utilisateur graphique jusqu'à des applications qui exploitent les dernières innovations fournies par ASP.NET, comme les services Web XML et Web Forms.

Le .NET Framework peut être hébergé par des composants non managés qui chargent le Common Language Runtime dans leurs processus et initient l'exécution du code managé, créant ainsi un environnement logiciel qui peut exploiter à la fois les fonctionnalités managées et non managées. Le .NET Framework fournit non seulement plusieurs hôtes de runtime, mais il prend également en charge le développement d'hôtes de runtime tiers.

Par exemple, ASP.NET héberge le runtime pour fournir un environnement côté serveur, évolutif pour le code managé. ASP.NET fonctionne directement avec le runtime pour activer des applications ASP.NET et des services Web XML.

L'illustration suivante montre les relations du Common Language Runtime et de la bibliothèque de classes avec vos applications et avec l'ensemble du système. L'illustration montre également comment le code managé opère au sein d'une architecture plus grande.

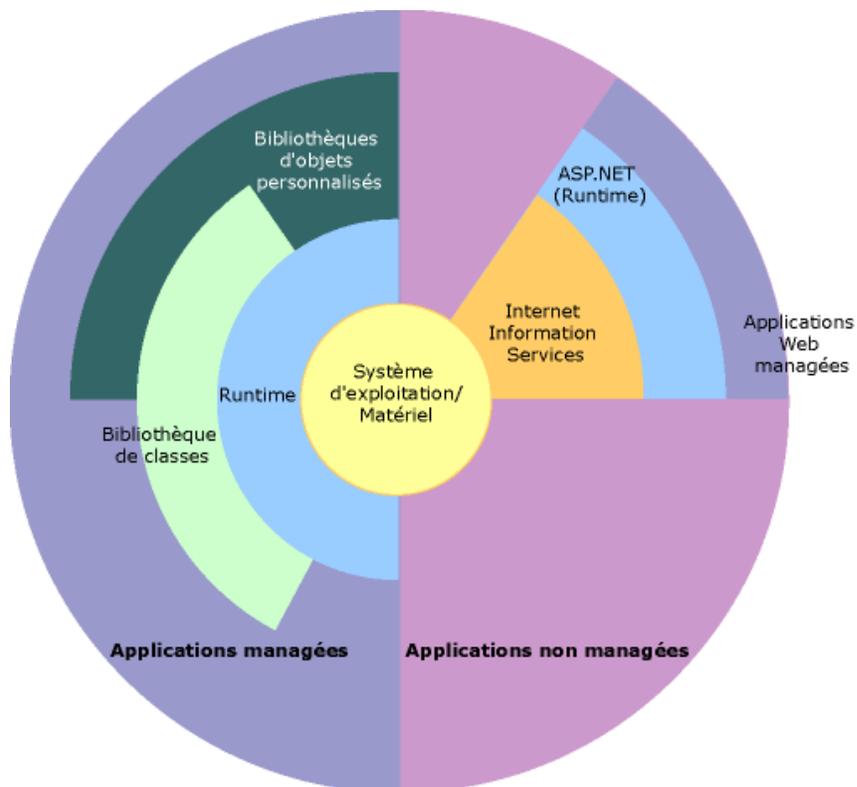


Figure 1.4: Schéma descriptif du .NET Framework

3.1.1. Fonctionnalités du common language runtime

Le Common Language Runtime gère la mémoire, l'exécution des threads, l'exécution du code, la vérification de la sécurité du code, la compilation et d'autres services du système. Ces fonctionnalités font partie intégrante du code managé qui s'exécute sous le Common Language Runtime.

En ce qui concerne la sécurité, les composants managés se voient attribués divers niveaux de confiance en fonction d'un nombre de facteurs qui comprennent leur origine (comme Internet, un réseau d'entreprise ou un ordinateur local). Cela signifie qu'un composant managé peut ou ne peut pas effectuer des opérations d'accès au fichier, des opérations d'accès au Registre ou d'autres fonctions délicates, même si ce composant est utilisé dans la même application active.

Le runtime fait appliquer la sécurité d'accès du code. Par exemple, les utilisateurs ont confiance dans le fait qu'un fichier exécutable incorporé dans une page Web peut afficher une animation sur l'écran ou chanter une chanson mais ne peut pas accéder à leurs données personnelles, leur système de fichiers ou leur réseau. Les fonctionnalités de sécurité du runtime permettent ainsi à des logiciels légitimes, déployés sur Internet de comporter un grand nombre de fonctionnalités.

Le runtime garantit également un code robuste en implémentant une infrastructure de vérification de code et de type stricte portant le nom de système de type commun (CTS, Common Type System). Le CTS garantit que le tout le code managé soit auto descriptif. Les différents compilateurs de langage Microsoft et tiers génèrent du code managé conforme au système de type commun (CTS, Common Type System). Cela signifie que le code managé peut consommer d'autres instances et types managés, tout en appliquant strictement le respect et la sécurité des types.

En outre, l'environnement managé du runtime élimine un grand nombre de problèmes logiciels courants. Par exemple, le runtime traite automatiquement la disposition des objets et gère les références aux objets, les libérant lorsqu'ils ne sont plus utilisés. Cette gestion automatique de la mémoire résout les deux erreurs d'application les plus courantes, le manque de mémoire et les références mémoires non valides.

Le runtime accélère également la productivité du développeur. Par exemple, les programmeurs peuvent écrire des applications dans le langage de développement qu'ils ont choisi, tout en tirant pleinement parti du runtime, de la bibliothèque de classes, et de composants

écrits dans d'autres langages par d'autres développeurs. Tout fournisseur de compilateur qui choisit de cibler le runtime peut en faire de même. Les compilateurs de langage qui ciblent le .NET Framework rendent les fonctionnalités du .NET Framework disponibles au code existant écrit dans ce langage, ce qui simplifie considérablement le processus de migration pour les applications existantes.

Si le runtime est conçu pour les logiciels du futur, il prend également en charge les logiciels d'aujourd'hui et d'hier. L'interopérabilité entre les codes managés et non managés permet aux développeurs de continuer à utiliser des composants COM et des DLL nécessaires.

Enfin, le runtime peut être hébergé par des applications côté serveur hautement performantes, comme Microsoft SQL Server et les services IIS (Internet Information Services). Cette infrastructure vous permet d'utiliser du code managé pour écrire une logique métier tout en profitant des performances supérieures du meilleur des serveurs d'entreprise prenant en charge l'hébergement runtime. [3]

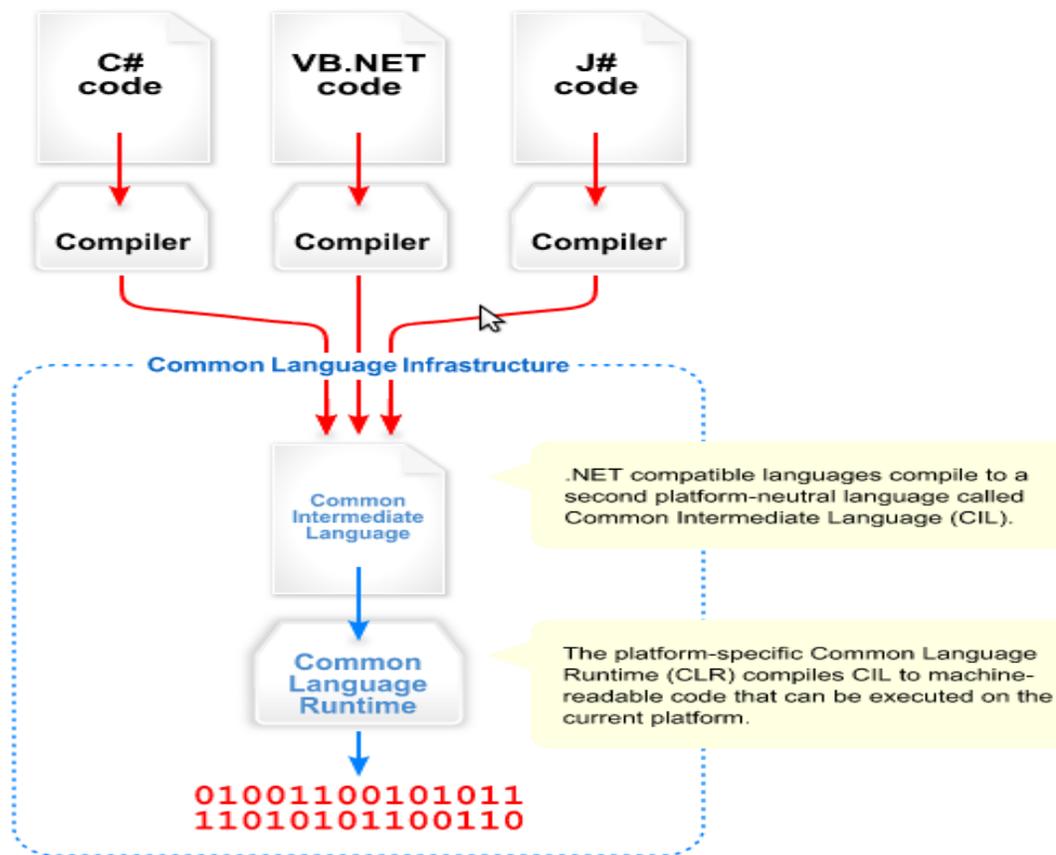


Figure 1.5: Schéma Descriptif du CLR

4. Conclusion

Dans ce chapitre on a défini les différents systèmes d'exploitation utilisés dans notre système et on a aussi pu voir l'importance du .NET Framework. et ce qui attire l'attention c'est l'évolution constante des différentes plateformes en trouvant souvent de nouvelles mises à jour qui sont exigées par les applications les plus récentes qui permettent de faire de nouvelles choses et de nouvelles manœuvres sur l'écran tactile et on a tenu à montrer cela à travers les définitions des OS .

Chapitre II : Modélisation du système

1. Introduction

Après avoir évoqué les différents systèmes d'exploitation sur lesquels l'application va s'exécuter. Le système doit être modélisé selon une méthodologie d'analyse et de conception. La méthode choisie est la méthode Merise.

Donc, l'objectif de ce chapitre consiste à fournir les étapes fondamentales pour la conception de notre système de gestion du parc automobile universitaire.

2. Merise

Merise se positionne comme une méthode de conception de SI sur le plan de son organisation générale. Cette méthode a pour principal avantage de permettre la compréhension et la formalisation des besoins du métier que vers la réalisation de logiciel. Elle favorise donc le dialogue entre concepteurs et maîtrise d'ouvrage, tout particulièrement dans les projets de développement de systèmes de gestion intégrée. [4]

2.1. Modélisation

Selon la méthode Merise, les diagrammes appropriés pour notre projet sont :

- Le Modèle Conceptuel des Données
- Le Modèle Logique des Données

2.1.1. Modèle conceptuel de données (MCD)

Cette schématisation a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information.

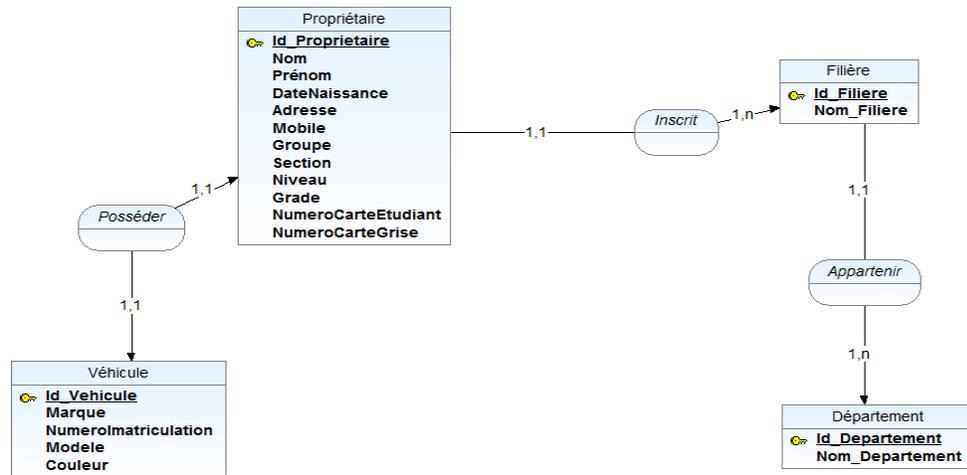


Figure 2.1 : Modèle conceptuel de données

2.1.2. Modèle logique des données (MLD)

Il consiste à décrire la structure de données utilisée et à préciser le type de données utilisées lors des traitements.

Le Modèle conceptuel des données défini plus haut est traduit en un modèle logique de données défini comme suit :

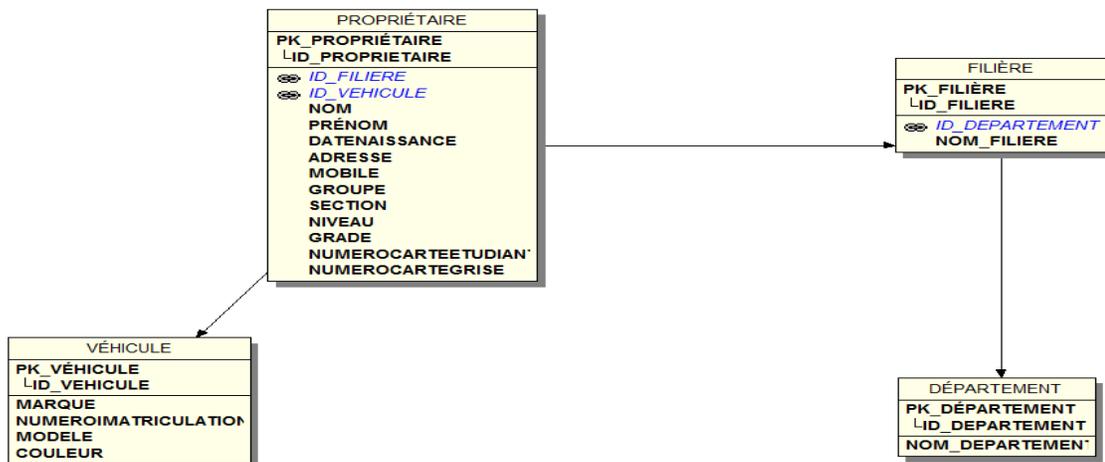


Figure 2.2 : Modèle logique de données

3. Architecture du système

3.1. Graphe de dépendance fonctionnel

Dans ce graphe nous allons présenter les différentes relations fonctionnelles entre les projets de notre système

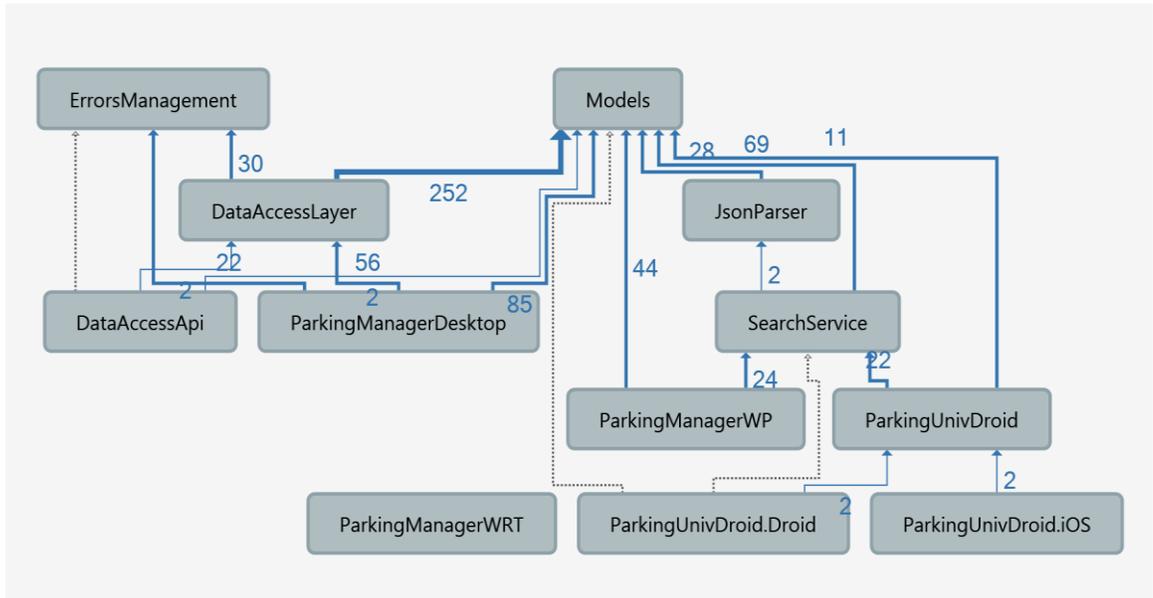


Figure 2.3: Graphe de dépendance fonctionnelle

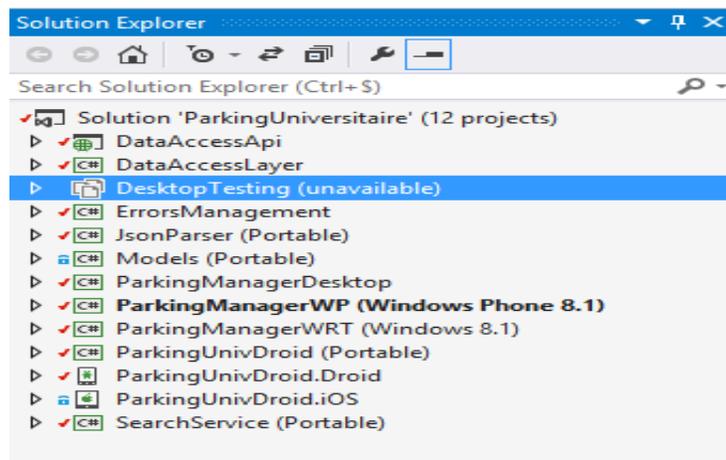


Figure 2.4 Capture d'écran de l'explorateur de solution

3.2. MVVM pattern

Le Modèle-Vue-VueModèle est une architecture et une méthode de conception utilisée dans le génie logiciel.

MVVM est un pattern intéressant à mettre en place dans les projets pour plusieurs raisons :

- Le faible couplage entre la Vue et la VueModèle permet de pouvoir modifier facilement la vue sans avoir d'impact sur la VueModèle (et vice versa).
- Il permet de tester de manière séparée les différents éléments de la solution.
- Il permet une maintenance facilitée des projets.
- Le même code dans les VueModèle et Modèle peut être facilement réutilisé dans d'autres projets tout en utilisant des vues différentes.

Pour permettre de mieux comprendre l'organisation du pattern MVVM, le schéma suivant explique de manière simple les échanges entre les différents éléments.

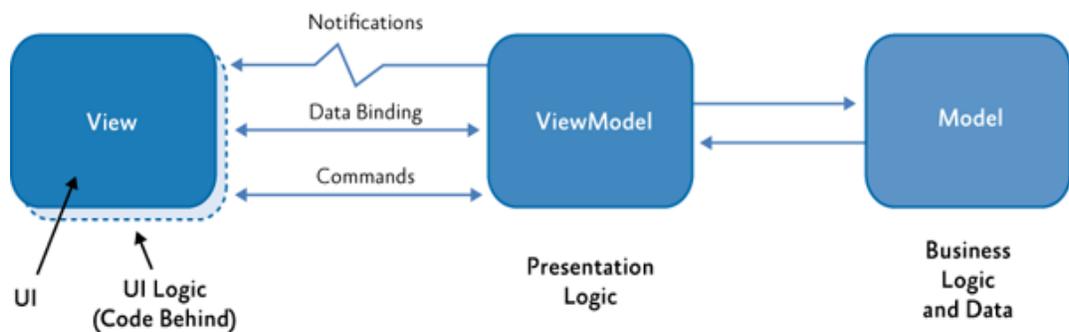


Figure 2.5: Schéma Descriptif du pattern MVVM

- **Model** : le modèle contient les données. Généralement, ces données proviennent d'une base de données ou d'un service externe comme un API.
- **View** : la vue correspond à ce qui est affiché (la page web dans notre cas). La vue contient les différents composants graphiques (boutons, liens, listes) ainsi que le texte.
- **ViewModel** : ce composant fait le lien entre le modèle et la vue. Il s'occupe de gérer les liaisons de données et les éventuelles conversions. C'est ici qu'intervient le binding. [5]

3.3. Data Api

Vu la nature de notre système cross plateforme il fallait trouver une solution pour pouvoir accéder à la base de données via les différentes plateformes mobile (smartphone, tablette). Alors on a opté pour la meilleure et la plus efficace des solutions de créer un service qui peut interroger le SGBD du serveur MySQL pour répondre aux requêtes http envoyées par les différents smartphones.

Le serveur MySQL à son tour répond aux requêtes http de deux manières différentes soit un fichier JSON soit un fichier XML tout dépend des paramètres envoyés dans la requête http, et pour suivre le nouveau mode de développement dans nos différentes applications on utilise des fichiers JSON.

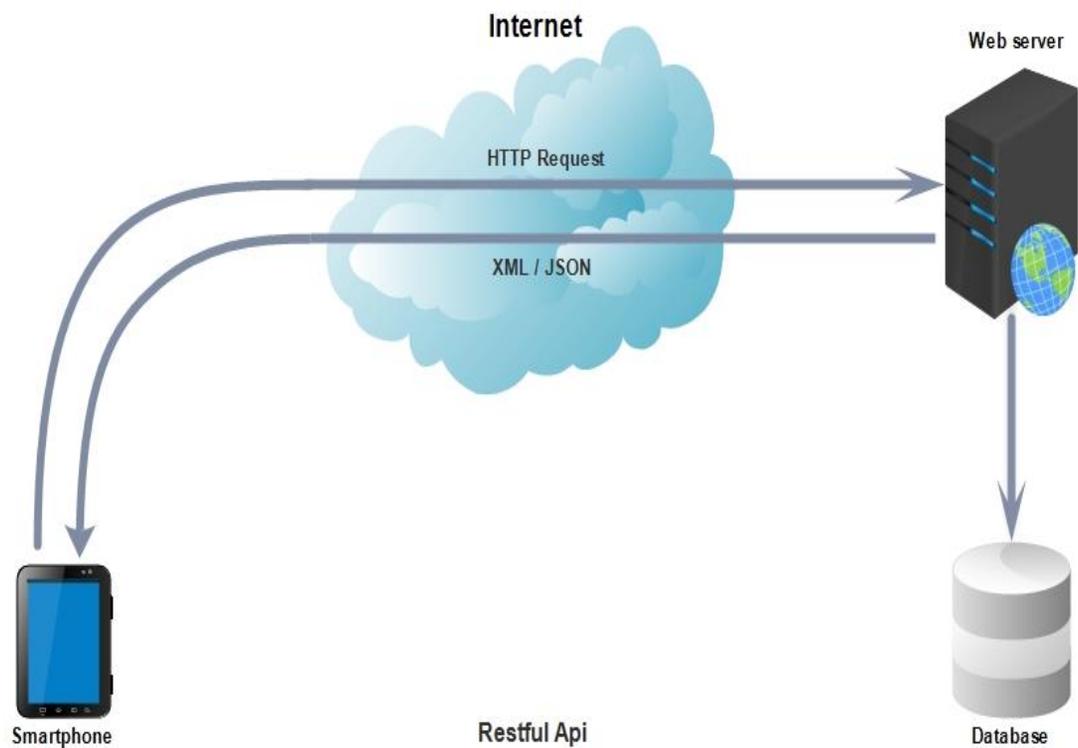


Figure 2.6: Schéma descriptif du fonctionnement du Data Api

4. Conclusion

Dans ce chapitre on a réalisé la modélisation Merise de notre base de données et l'architecture du système. Cette étape est très importante pour pouvoir créer la base de données et réaliser les applications et il faut signaler que la modélisation nous a permis de bien organiser notre travail et définir les différentes étapes afin d'arriver à l'étape finale celle de concrétiser les idées en application ce qui rend cette étape de la modélisation une étape intermédiaire entre l'idée a l'état brut et a concrétisation sur le terrain .

Chapitre III: Réalisation du système

1. Introduction

Ce chapitre permet de recenser les différents outils de programmation utilisés qui ont servi à la réalisation de l'application et de ses fonctionnalités. Aussi, l'application sera expliquée et les différentes interfaces seront présentées.

2. Environnements de développement (EDI)

Dans la réalisation de notre projet, on a utilisé deux EDI qui sont définis ci-dessous.

2.1. Visual Studio

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C# et Visual C++ utilisent tous le même environnement de développement intégré (EDI), qui permet le partage d'outils et facilite la création de solutions à plusieurs langages. Par ailleurs, ces langages utilisent les fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML.

Visual studio existe en plusieurs versions, on a utilisé la version Professional 2013 avec l'update 4. [6]

2.2. Xamarin studio

Xamarin Studio est un EDI moderne et sophistiqué avec de nombreuses fonctionnalités pour la création d'application d'iOS et Android en C#, Il inclut iOS et Android, et le contrôle de la source intégré.

Xamarin Studio est organisé en plusieurs sections pour la gestion des fichiers d'application et les paramètres, la création d'un code d'application, et de débogage.

L'utilisation de cet EDI requiert une licence qu'on a obtenu en justifiant notre statut d'étudiant et vu la nature de notre projet qui est caractérisé par le cross plateforme,

Xamarin studio représente la meilleure solution afin de garder le même langage et de pouvoir utiliser le projet Xamarin.forms qui nous permet de créer une interface native partagée entre Android et iOS en se basant sur le même code logique.

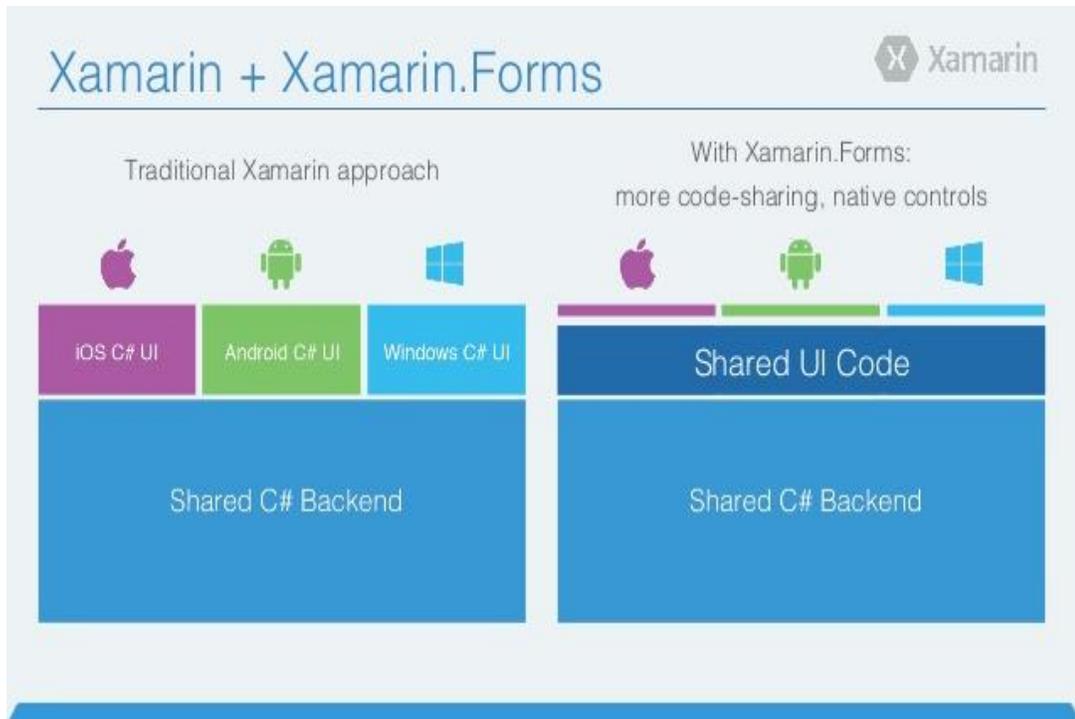


Figure 3.1: Schéma descriptif de l'utilité de Xamarin.Forms

3. Les langages de programmation utilisés

Dans un esprit de défi, et de vouloir se servir d'un langage qui n'a jamais été utilisé dans notre université, on a choisi le C# et le XAML deux langages puissants et très répondus dans le milieu professionnel.

3.1. Le C# (c Sharp)

Le C# est un langage orienté objet sécurisé qui permet aux développeurs de générer diverses applications qui s'exécutent sur l'environnement .NET comme expliqué au préalable. C# donne la possibilité de créer des applications Windows, des services Web, des composants distribués et des applications ayant des interactions avec les différents types de bases de données.

La syntaxe C# est très expressive, mais elle est également facile à reconnaître à ses accolades si vous avez déjà utilisé C, C++ ou Java. La syntaxe C# permet de répondre à de nombreuses complexités de C++ en fournissant des fonctionnalités puissantes telles que des types valeur nulle, des énumérations, des Delegates, des expressions lambda et des accès directs à la mémoire qui n'existent pas dans d'autres langages utilisant des

Chapitre III : Réalisation du système

technologies de virtualisation tel que Java. C# prend en charge des méthodes et types génériques qui améliorent la cohérence et les performances des types, ainsi que des itérables (littérateurs), qui permettent aux implémenteurs de classes de collection de définir des comportements d'itération personnalisés simples à utiliser par le code client.

N'oublions pas aussi Les expressions LINQ (Language Integrated Query) qui ont été utilisées à plusieurs reprises dans ce projet et transforment les requêtes fortement typées en construction de langage de premier ordre.

Outre ces principes orientés objet de base, C# permet de développer facilement des composants logiciel à travers plusieurs constructions de langage innovatrices, y compris les éléments suivants :

- Les signatures de méthodes encapsulées, appelées délégués, qui activent les notifications d'événement de type sécurisé.
- Les propriétés, utilisées comme accesseurs pour les variables membres privés.
- Les attributs, qui fournissent des métadonnées déclaratives à propos des types au moment de l'exécution. [7]

3.2. Le XAML

Pour la réalisation de l'application on a besoin d'une technologie permettant de concevoir des interfaces graphique et non pas console, on a songé à utiliser la technologie « Windows Presentation Foundation ». Cette technologie repose sur le XAML pour concevoir les Users interfaces. XAML est l'acronyme de « eXtensible Application Markup Language », donc c'est un langage à balises pour applications extensibles. Il est appelé extensibles car les applications ont la possibilité d'ajout des composants pendant l'exécution cette technique est appelée « RUNTIME ». Donc pour faire court le XAML sert à décrire le contenu des fenêtres que nous allons voir à l'écran. En fait, un fichier XAML correspond à une page dans nos applications. Les éléments du langage XAML correspondent à des balises de positionnement, des composants d'interface utilisateur graphiques tel que les boutons, champ de texte, menu, liste, bouton radio, ...etc.

Ces composants de même que le texte utilisent des graphismes vectoriels permettant d'ajuster leur taille sans perte de définition graphique. [8]

3.3. SGBD

Dans la réalisation de notre projet on devait impérativement choisir un SGBD qui prend en charge la structuration, le stockage, la mise à jour et la maintenance de notre base de données. Alors on a opté pour MySQL l'un des SGBD les plus utilisés

3.3.1. Caractéristique de 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 multi-thread et multi-utilisateurs.

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. [9]

4. Description de l'application

L'application a pour nom Park Optimiser. Le but est de sécuriser le campus universitaire en contrôlant le flux des véhicules circulant à l'intérieur.

L'application permet de :

- contrôler les différents véhicules entrant et sortant du campus universitaire.
- identifier le propriétaire du véhicule.
- avoir des informations sur un véhicule ou sur un propriétaire.

L'application est en fait constituée de quatre autres applications :

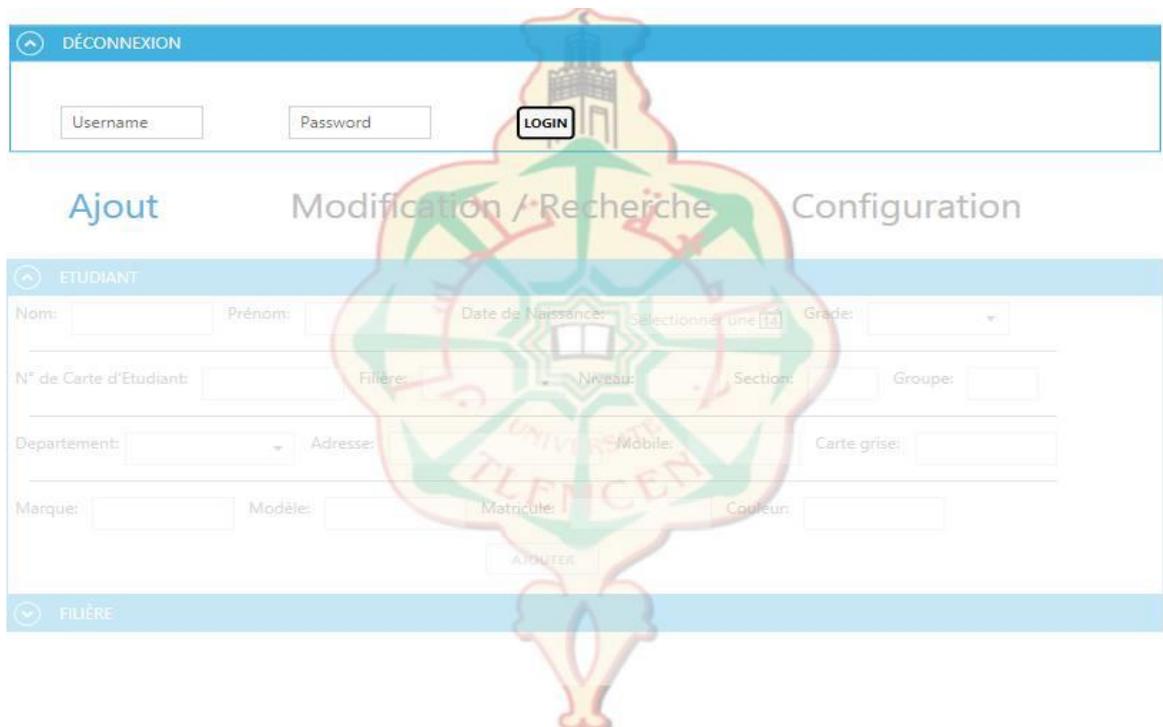
- une application pour l'administration,
- trois applications mobiles pour les agents du terrain. Ces applications phone sont multiplateformes puisqu'elles s'exécutent chacune sur un des trois systèmes d'exploitation : Android, IOS et Windows Phone. Elles permettent d'identifier les véhicules circulant pour autoriser ou interdire leur accès au campus universitaire.

Une explication détaillée est fournie pour les quatre applications et leurs principaux menus sont présentés dans ce qui suit.

3.1. Application Desktop

La version desktop de notre application est la partie la plus importante car l'utilisateur ou l'administrateur peut tout contrôler depuis son écran d'ordinateur. C'est le point de départ de l'utilisation de l'application (Park Optimizer). Elle permet de construire la base des véhicules inscrits au sein du campus universitaire et de renseigner tous les informations des propriétaires de ces véhicules.

L'interface présente un "Login" ou l'utilisateur ou l'administrateur doit se connecter par un nom d'utilisateur et le mot de passe pour accéder à l'interface d'accueil.



The image shows a screenshot of a web application interface. At the top, there is a blue header with a 'DÉCONNEXION' button. Below it is a login form with 'Username' and 'Password' input fields and a 'LOGIN' button. Underneath the login form are three main menu items: 'Ajout', 'Modification / Recherche', and 'Configuration'. Below these is a section for 'ETUDIANT' with various input fields for personal and academic information: 'Nom', 'Prénom', 'Date de Naissance', 'Grade', 'N° de Carte d'Etudiant', 'Filière', 'Niveau', 'Section', 'Groupe', 'Département', 'Adresse', 'Mobile', 'Carte grise', 'Marque', 'Modèle', 'Matricule', and 'Couleur'. At the bottom of this section is an 'AJOUTER' button. The interface is overlaid with a large, semi-transparent watermark of the University of Bafra logo.

Figure 3.2: Application Desktop/Login Page et Menu d'ajout

Une fois connecté, l'utilisateur aura trois menus : Ajout, Modification/recherche et Configuration.

- Le menu Ajout :

Son importance est indiscutable puisque l'administrateur constitue sa base de données en ajoutant à chaque fois un nouveau véhicule ainsi que toutes les informations nécessaires de son propriétaire comme le nom, le prénom, le numéro de téléphone, le grade (Etudiant,

Chapitre III : Réalisation du système

Professeur, Personnel), le numéro de carte d'étudiant, le matricule du véhicule, etc. Ainsi ses informations permettront une recherche facile et rapide des véhicules et des propriétaires en cas de problème rencontrés.

- Le menu Modification/ Recherche

La constitution d'une base de données très large rend obligatoire l'existence d'un outil de recherche de données (véhicule, propriétaire). C'est ce qui nous a amené à créer le menu « Modification/Recherche » dans lequel on peut procéder à la recherche d'un propriétaire d'un véhicule précis en filtrant les paramètres de recherche (Nom, Prénom, Matricule ou le numéro de la carte d'étudiant). Une fois le propriétaire trouvé, toutes ses informations déjà entrées dans le menu Ajout apparaissent.

Il est aussi possible de supprimer un propriétaire ou de modifier ses informations parce qu'il a changé de véhicule par exemple.

DECONNEXION

DECONNEXION

Ajout Modification / Recherche Configuration

SUPPRESSION PROPRIÉTAIRE

Selectionner le propriétaire à supprimer SUPPRIMER

RECHERCHE DE PROPRIÉTAIRE

Selectionner le parametre de rechercher RECHERCHE Nom Prénom Matricule Numero carte d'étudiant

RAFRACHIR

Nom et Prénom selectionné: Lachachi Charaf

NOM	PRENOM	DEPARTEMENT	MATRICULE
lachachi	anes	Informatique	982-111-13
Alem	salim	Mathématique	01798-112-13
lachachi	sarah	Biologie	923-113-31
Alem	samir	Physique	01877-112-13
Lachachi	Charaf	Informatique	00694-113-13
Benamar	Mohammed	Informatique	1033-115-13
Mehammedi	mohammed	Mathématique	5987-115-31
Amar	Morad	Physique	00001-115-16

DÉTAILLE DE L'ÉLÉMENT SÉLECTIONNÉ

Grade: Etudiant

Filière: Mathematique-Informatique Niveau: L3 Section: A Groupe: A2 N° carte d'étudiant: 11/170828

Mobile: 551035253

Marque: Seat Model: Ibiza Couleur: Blanche N° carte Grise: 5698/784/963121

SAUVEGARDER

Figure 3.3: Application Desktop/Menu de modification et recherche

Chapitre III : Réalisation du système

- Le menu Configuration,

L'administrateur peut ajouter d'autres utilisateurs en leurs créant un compte afin d'avoir l'accès aux commandes nécessaires.



The image shows a screenshot of a web application interface. At the top, there is a blue header bar with a back arrow icon and the text 'DÉCONNEXION'. Below this is a white box containing a 'DÉCONNEXION' button. Underneath, there are three navigation tabs: 'Ajout', 'Modification / Recherche', and 'Configuration'. The 'Configuration' tab is selected. Below the tabs is another blue header bar with a back arrow icon and the text 'AJOUT D'UTILISATEUR'. This is followed by a form with two input fields: 'Utilisateur :' and 'Mot de passe :'. To the right of the 'Mot de passe' field is an 'AJOUTER' button. A large, faint watermark of the University of Tlemcen logo is visible in the background.

Figure 3.4: Application Desktop/Menu de configuration

4.2. Applications mobiles

La version mobile de notre application est utilisée au sein du terrain par les agents de sécurité se trouvant à l'entrée du campus universitaire. Elle se caractérise par une facilité d'utilisation et une efficacité remarquable.

Un apprentissage est recommandé pour les agents de sécurité du terrain pour leur permettre d'apprendre à utiliser et à bien exploiter l'application mobile.

4.2.1. Android et IOS

Il n'y a pratiquement pas de différence entre les deux OS dans notre version mobile de l'application, les deux OS ont les mêmes fonctionnalités et la même configuration des touches.

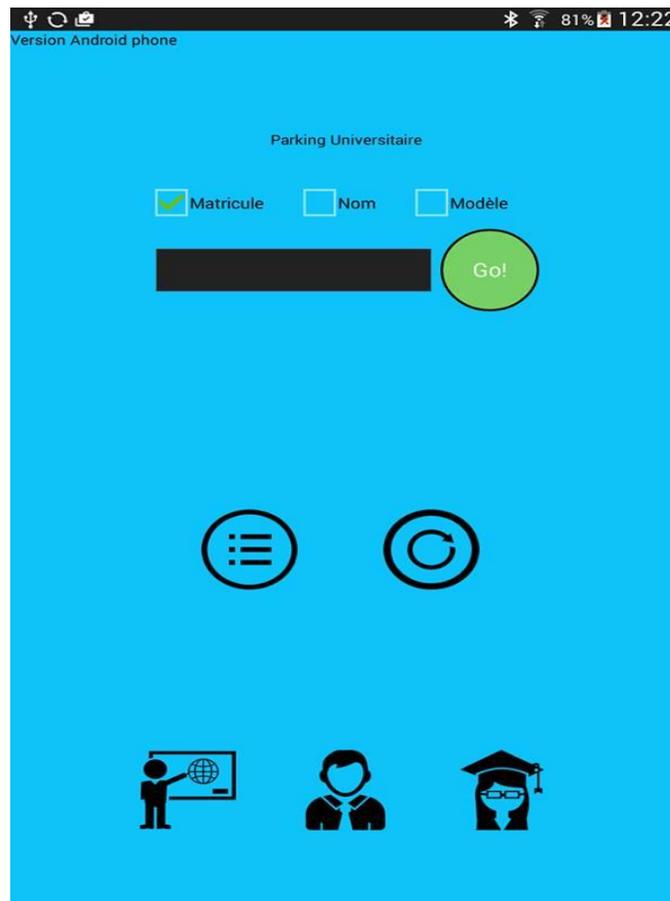


Figure 3.5 : Application IOS-ANDROID / Menu principal

L'interface est simplifiée avec une barre de recherche dans laquelle on peut filtrer la recherche par le matricule du véhicule, le nom du propriétaire ou le modèle du véhicule. On peut rétrécir encore plus notre angle de recherche en ciblant que les étudiants en appuyant sur l'icône « Etudiant », de même pour le personnel et les professeurs.

Il est possible d'actualiser les informations vu que ces dernières sont régulièrement mises à jour à partir de la version Desktop.

Enfin, l'utilisateur peut aussi accéder à la liste intégrale de tous les propriétaires de véhicules inscrits au sein du campus universitaire ainsi qu'à toutes leurs informations correspondantes.

4.2.2. Windows phone

La version mobile de notre Application sous la plateforme Windows Phone se caractérise par une interface légèrement différente de celle des autres OS mais avec quasiment la même configuration.

Tout en haut, on retrouve la barre de recherche et la possibilité de filtrer selon le choix des catégories (étudiants, personnel, et professeurs), la possibilité d'actualiser les informations et de consulter la liste intégrale, comme cela a été expliqué précédemment.



Figure 3.6 : Application Windows Phone / Menu principal

La particularité avec la version de Windows phone est la possibilité d'avoir les informations nécessaires d'un véhicule juste en prenant une photo du matricule (en appuyant sur l'icône « appareil photo »). Cette démarche facilite d'une manière

Chapitre III : Réalisation du système

considérable le travail de l'utilisateur d'un côté et le propriétaire du véhicule de l'autre en gagnant du temps. La recherche se fait automatiquement en capturant le matricule par photo et les informations sont tout de suite émises.

Cette fonctionnalité utilise le système de reconnaissance optique des caractères (OCR) comme celui utilisé dans les scanners de documents, et pour pouvoir bénéficier de ce système on a développé un code qui permet de scanner la photo prise par l'utilisateur et reconnaître les caractères qui serviront à identifier le propriétaire et ses informations dans la base de données automatiquement.

5. Conclusion

Dans cette partie de notre projet on a opté pour la conception et la réalisation d'un outil à la fois simple à utiliser et très efficace sur le terrain, qu'on a pu adapter aux différentes plateformes, le rendant disponible sur tous les Smartphones récents.

Conclusion générale

Notre projet vise à informatiser le système de contrôle du flux automobile au sein du campus universitaire et à faciliter les procédures administratives dans un but de sécuriser ce flux.

La réalisation de notre système exige plusieurs conditions comme par exemple la largeur du campus universitaire qui requiert le contrôle de tous les points d'accès. Cela nous conduit à opter pour un système déployé sur différentes plateformes afin de garantir une sécurité optimale. Ce système représente un apport considérable en matière de gain de temps et d'organisation pour parer aux lacunes telles que la lenteur dans l'identification des véhicules et l'absence de coordination entre les agents de sécurité sur le terrain et l'administration, ce qui nous conduit à atteindre nos objectifs.

Cependant, la mise en œuvre de notre système renfermait aussi des difficultés sur le plan matériel par l'absence d'un Mac OS sur lequel on pouvait travailler afin de développer l'application IOS. Ce qui nous a conduits à la nécessité de trouver une alternative avec l'utilisation d'une machine virtuelle Mac OS malgré son caractère non officiel.

La réalisation de ce système était une expérience unique pour moi, qui m'a permis de développer mes connaissances en langage C# et d'intégrer l'univers de la programmation cross-plateforme.

Tout projet scientifique doit avoir une continuité et des perspectives. Avec la concrétisation de nos idées dans un système complet, on vise à continuer le travail dans l'avenir en améliorant les différentes fonctionnalités pour donner au système plus de fiabilité et une meilleure garantie de sécurité.

Bibliographie

- [1] W.-M. Lee, Android Application Development Cookbook, John Wiley & Sons, Inc. , 2013.
- [2] A.-I.-Q. ce-qu'IOS, «<https://www.apple.com/fr/ios/what-is/>,» [En ligne]. Available: <https://www.apple.com/fr/ios/what-is/>. [Accès le 12 mars 2015].
- [3] «Overview of .net Framework,» [En ligne]. Available: [https://msdn.microsoft.com/en-us/library/zw4w595w\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx). [Accès le 05 février 2015].
- [4] D. Gros, Cours Conception Informatique / MCD – 1ère partie, lavoisier, 2012.
- [5] «Introduction à MVVM,» [En ligne]. Available: <https://havefuncoding.wordpress.com/2012/11/26/introduction-a-mvvm/>. [Accès le 19 avril 2015].
- [6] «Introduction à Visual Studio,» [En ligne]. Available: <https://msdn.microsoft.com/fr-fr/library/aa291755%28v=vs.71%29.aspx>. [Accès le 13 février 2015].
- [7] O. Dahan, Tome1 c#, e-naxos, 2014.
- [8] O. Dahan, Tome1 XAML, e-naxos, 2014.
- [9] J.-C. CABIANCA, SGBD MySql, 2012: STS IRIS .

Liste des figures

Figure 1.1: Interface Android	7
Figure 1.2 Interface Windows Phone 8.1	9
Figure 1.3: Interface IOS 8	11
Figure 1.4: Schéma descriptif du .NET Framework	13
Figure 1.5: Schéma Descriptif du CLR	15
Figure 2.1 : Modèle conceptuel de données	19
Figure 2.2 : Modèle logique de données.....	19
Figure 2.3: Graphe de dépendance fonctionnelle	20
Figure 2.4 Capture d'écran de l'explorateur de solution	20
Figure 2.5: Schéma Descriptif du pattern MVVM	21
Figure 2.6: Schéma descriptif du fonctionnement du Data Api	22
Figure 3.1: Schéma descriptif de l'utilité de Xamarin.Forms	26
Figure 3.2: Application Desktop/Login Page et Menu d'ajout.....	29
Figure 3.3: Application Desktop/Menu de modification et recherche.....	30
Figure 3.4: Application Desktop/Menu de configuration	31
Figure 3.5 : Application IOS-ANDROID / Menu principal	32
Figure 3.6 : Application Windows Phone / Menu principal	33

Liste des Abréviations

- UI : User interface
- OS : Operating System
- EDI : Environnement de Développement Intégré
- SQL : Structured Query Language
- HTTP : Hypertext Transfer Protocol
- HTML : Hyper Text Markup Language
- XAML : eXtensible Application Markup Language
- XML : Extensible Markup Language
- ASP : Active Server Pages
- JSON : JavaScript Object Notation
- CTS : Common Type System
- IIS : Internet Information Services
- COM : Component Object Model
- DLL : Dynamically Linked Library
- SGBD : Système de Gestion de Base de Données
- API : Application Programming Interface
- MVVM : Model View ViewModel
- MLD : Modèle Conceptuel de Données
- MCD : Modèle Logique de Données
- XNU : X is Not Unix
- MIME : Multipurpose Internet Mail Extensions

Résumé

A travers ce projet , on a voulu améliorer la gestion du parc automobile universitaire en mettant en place un système informatique qui vise à faciliter la tâche des agents de sécurité dans leurs mission de contrôle des véhicules tout en ayant une coordination avec l'administration, et cela a été rendu possible grâce à un système cross plateforme (Windows, IOS, Android) qu'on a développé dans le but de garantir plus de sécurité au sein du campus universitaire.

Mots clés : Multiplateforme, Android, Windows phone, iOS, C#

Abstract

Through this project, we wanted to improve the management of university's fleet vehicles by setting a data processing system in order to ease the task of the security agents in the field to check the vehicles by having a coordination with the administration, all of that became possible thanks to a cross-platform system (Android, iOS, Windows phone) that we developed in order to guarantee more security in the university campus.

Keys words: Cross-platform, Android, Windows phone, iOS, C#

ملخص

عملنا من خلال هذا المشروع على تحسين أداء الإدارة الجامعية فيما يتعلق بتسييرها لحظيرة السيارات عن طريق وضع نظام آلي خاص لتسهيل عمل أعوان الأمن على الميدان أثناء مراقبتهم لتدفق السيارات و ذلك بوجود تنسيق مع الإدارة الوصية، و هذا أصبح ممكناً بفضل نظام كروس بلاتفورم أو عبر برمجي (أندرويد ، iOS ، ويندوس فون) الذي طورناه بهدف ضمان أمن أكبر داخل الحرم الجامعي .

كلمات مفتاحية: كروس بلاتفورم؛ أندرويد؛ ويندوس فون