# Introduction

# Contexte

## Introduction



## Le client



## La demande initiale



# Etude de marché



## Les produits existants



# Analyse des besoins



## Fonctionnalités



## Utilisateurs



## Interface utilisateur





## Sécurité



## Plateformes

### Mobile

### Serveur

# Conception

## Cahier des charges



## Partie visuelle

## Diagrammes

# Développement

## Dépôt du projet et versioning



## Langages « cross-platform »



## 

## Langage Swift (Apple)

En définitive afin de mener ce projet à bien, j’ai décidé de changer mon environnement de travail complètement et de passer de Windows à Apple en travaillant sur un Mac. Ceci m’a permit d’être complétement en phase avec l’infrastructure Apple et de pouvoir avoir les outils de base proposé directement par Apple pour coder des applications iOS.

Il existe plusieurs langages de programmation pour iOS. Swift est un langage de programmation multiparadigme développé par Apple depuis 2014 pour le développement d'applications iOS, MacOs, WatchOS et TvOS. Il a été conçu pour être sûr, rapide et moderne, tout en étant facile à utiliser. Swift combine des éléments de langages de programmation impératifs, fonctionnels et orientés objet, et il est destiné à remplacer le langage Objective-C, utilisé historiquement pour le développement d'applications pour les plateformes Apple.

Swift est conçu pour être sûr en évitant les erreurs de programmation courantes telles que les pointeurs nuls ou les dépassements de tableau. Il offre également une syntaxe concise et expressive, ce qui le rend facile à apprendre et à utiliser. En outre, Swift est optimisé pour des performances élevées, ce qui signifie qu'il peut exécuter des tâches complexes rapidement, même sur des appareils avec des ressources limitées.

Swift est un langage de programmation fortement typé, ce qui signifie que chaque variable, constante et fonction a un type spécifique qui doit être déclaré explicitement. Cela permet à Swift de détecter les erreurs de type dès la compilation plutôt qu'à l'exécution, ce qui rend le code plus sûr et plus prévisible. En outre, la vérification de type aide les développeurs à comprendre comment utiliser correctement les différentes parties de leur code.

Swift offre également la possibilité de déclarer des types de données optionnels, qui peuvent avoir une valeur ou être nuls. Cela permet aux développeurs de gérer facilement les situations où une variable peut être absente où avoir une valeur inconnue.

## Communications entre iOS et Odoo



### L’identification d’un utilisateur



# Test et validation

# Conclusion

# Webographie