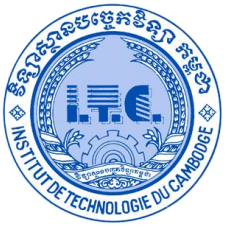
ក្រសួងអប់រំយុវជនឯកីឡា

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា



ដេប៉ាតឺម៉ង់ទេពកោសល្យព័ត៌មានវិទ្យា និងទំនាក់ទំនង

គម្រោងសញ្ញាប័ត្រវិស្វករ

ប្រធានបទ	:	កម្មវិធីផ្ញើប័ណ្ណប្រៃសណីយ៍លើទូរស័ព្ទ អាយហ្វូនជំនាន់ទី ២
និស្សិត	:	លោក សូ សុភា
ឯកទេស	:	ព័ត៌មានវិទ្យា និង ទំនាក់ទំនង
សាស្ត្រាចារ្យទទួលបន្ទុក	:	បណ្ឌិត សេង សុភាព
ឆ្នាំសិក្សា	:	២០១១- ២០១២

MINISTERE DE L'EDUCATION,
DELA JEUNESSE ET DES SPORTS

INSTITUT DE TECHNOLOGIE DU CAMBODGE

DEPARTEMENT GENIE INFORMATIQUE ET COMMUNICATION

MEMOIRE DE FIN D'ETUDESINGENIEUR

Titre	**:**	**Application Oopost pour iPhone version 2**
Etudiant	**:**	**M. SAU Sophea**
Spécialité	**:**	**Informatique et Communication**
Maître de stage	**:**	**Dr. SENG Sopheap**
Année scolaire	**:**	**2011 – 2012**

ក្រសួងអប់រំ យុវជន និង កីឡា

វិទ្យាស្ថានបច្ចេកវិទ្យាកម្ពុជា

ដេប៉ាតឺម៉ង់ ទេពកោសល្យ ព័ត៌មានវិទ្យា និង ទំនាក់ទំនង

គម្រោងសញ្ញាប័ត្រវិស្វករ

របស់និស្សិតសូ សុភា

កាលបរិច្ឆេទការពារសារណា : ០៩កក្កដា២០១២

អនុញ្ញាតអោយការពារគម្រោង

នាយកវិទ្យាស្ថាន_____

រាជធានីភ្នំពេញ ថ្ងៃទី ខែ កក្កដាឆ្នាំ ២០១២

ប្រធានបទ : កម្មវិធីផ្ញើប័ណ្ណប្រៃសណីយ៍លើទូរស័ព្ទអាយហ្វូន ជំនាន់ទី ២

សហគ្រាស : ខេមម៉ុប

ប្រធានដេប៉ាតឺម៉ង់ : បណ្ឌិត សេង សុភាព

សាស្ត្រាចារ្យដឹកនាំគម្រោង : បណ្ឌិត សេង សុភាព

អ្នកទទួលខុសត្រូវក្នុងសហគ្រាស : លោក យី សុផល្លី

រាជធានីភ្នំពេញ ឆ្នាំ ២០១២

MINISTERE DE L'EDUCATION,

DE LA JEUNESSE ET DES SPORTS

INSTITUT DE TECHNOLOGIE DU CAMBODGE

DEPARTEMENT GENIE INFORMATIQUE ET COMMUNICATION

MEMOIRE DE FIN D'ETUDES INGENIEUR

DE M. SAU Sophea

Date de soutenance : le 09 juillet 2012

« Autorise la soutenance du mémoire »

Directeur de l'Institut : Dr. OM Romny

Phnom Penh, le juillet2012

Titre : Application Oopost pour iPhone version 2

Etablissement du Stage : CamMob

Chef du département	**:**	**Dr. SENG Sopheap**
Le professeur encadrant du projet	**:**	**Dr. SENG Sopheap**
Responsable dans l'établissement	**:**	**M. YI Sophally**

PHNOM PENH, 2012

REMERCIEMENTS

Premièrement, je tiendrais à exprimer tout particulièrement mes reconnaissances et ma profonde gratitude aux personnes suivantes qui m'ont aidées á réaliser avec succès le projet proposé pendant mon stage de fin d'études:

Son Excellence **Mme. PHOEURNG Sackona**, Secrétaire d'Etat du Ministère de l'Education, de la Jeunesse et des Sports et Présidente du Conseil d'Administration de l'ITC, pour ses efforts considérables qui contribuent d'avantage au développement durable de l'ITC et qui font de l'ITC un pôle technologique reconnu pour sa qualité de formation à l'échelle nationale, régionale et internationale.

Son Excellence **Dr. OM Romny**, directeur de l'ITC pour sa bonne gestion de l'institut et ses bonnes coopérations avec les universités partenaires au niveau local, régional et international qui renforce la qualité de la formation des ingénieurs et des techniciens supérieurs.

Dr. SENG Sopheap, chef du département Génie Informatique et Communication, pour sa bonne gestion, ses bons conseils, et ses bonnes relations avec les entreprises qui accueillent chaque année des étudiants de département GIC pour faire du stage.

Dr. SENG Sopheap, mon maître de stage, de m'avoir expliqué des techniques permettant d'effectuer mon stage. Je le salue également pour ses remarques et corrections qui ont permis l'élaboration de ce mémoire.

M. YI Sophally, mon tuteur de stage, de m'avoir donné toujours des conseils très précieux pour résoudre des problèmes pendant le stage.

Merci à tous les enseignants de département GIC de me transmettre des connaissances très précieuses pendant mes études à l'ITC.

Finalement, je voudrais exprimer spécialement ma reconnaissance à mes parents pour le support financière, l'encouragement et la motivation tout au long de ma vie.

RESUME

L’initiative de la création l’application Oopost vient d’un concept « de la virtuosité à la réalité ». C’est-à-dire à partir de notre iPhone, il s’agit d’envoyer des cartes postales personnalisées à tout le monde. En faisant la synthèse entre2 idées principales qui sont la nouvelle technologie et l’envoie de la carte postale par la poste, Oopost a un grand succès avec beaucoup d’utilisateurs. En voyant tel succès, l’entreprise de CamMob, où mon stage est effectué, a proposé un projet appelé "Application Oopost pour iPhone version 2", qui garde le même conceptet ajoute desfonctionnalités supplémentaires pour aider l'utilisateur de créer lacarte postale plus personnalisée.

Le travail à faire au cours de ce stage est d'apprendre les nouvelles technologies et des caractéristiques principales d’iPhone, étudierles applications d’Oopost existants sur la plate-formeiPhone et iPad,et ajouter plus de fonctionnalitésdemandés par l’entreprise. J'ai commencé par identifier les besoins fonctionnels et le cas d'utilisation. Puis j'ai fait les tests unitaires pour valider chaque fonctionnalité et corriger lesbugs pour les éliminer dans cette nouvelle version.

ABSTRACT

The initiative of creating the application Oopost comes from a concept, "from the virtuosity to the reality." With our iPhone, we can send personalpostcards to everyone. By synthesizing 2 main ideas that are new technology and sending postcard by post, Ooposthave a great success with many users. By seeing this success, the CamMobCompany where I did the internship, has proposed a project called "ApplicationOopost for iPhone version 2", which keeps the sameconcept and add additional features to help users to create postcardmore customizable.

The work that I need to do during this internshipis to learn new technologies and special features of iPhone, study the existingOopost applications on iPhone and iPad platform, and add morefeaturesinquiredby the company. I began by identifying requirements and use cases. Then I made the unit tests to validate each feature and correct bugs in order to eliminate it in this new version.

TABLE DES MATIERES

LISTE DESILLUSTRATIONS

Figures Page

LISTE DES TABLEAUX

Tableaux Page

LISTE DES ABREVIATIONS

API	:	Application Programming Interface
APNS	:	Apple Push Notification Service
CSS	:	Cascading Style Sheet
GIC	:	GénieInformatiqueet Communication
HTML	:	Hypertext Markup Language
HTTP	:	HyperText Transfer Protocol
IDE	:	Integrated Development Environment
IHM	:	Interface Homme Machine
iOS	:	iPhone Operating System
ITC	:	Institut de Technologie du Cambodge
QA	:	Quality Assurance
SDK	:	Software Development Kit
SGBD	:	Système de Gestion de Base de Données
SQL	:	Structure Query Language
UML	:	Unified Modeling Language
URL	:	Uniform Resource Locator
UX	:	User Experience

INTRODUCTION

Dans le cadre de la formation des ingénieurs à l'ITC, un stage au moins de 12 semaines est proposé aux étudiants en 5éme année du département Génie Informatique et Communication (GIC) afin qu'ils puissent mettre en pratique leurs connaissances théoriques et acquérir des nouveaux connaissances ainsi que des nouvelles technologies dans une entreprise publique ou privée.

Jour après jour, la technologie nous offre beaucoup de nouvelles innovations, en particulier le Smartphone. Jusqu'à maintenant, il y a plusieurs plates-formes disponibles dans le marché comme BlackBerry, Android, Windows Phone et en particulier iOS qui nous offert le système opératoire pour développer ses applications. En voyant ce phénomène, l'entreprise de CamMob où mon stage est effectué est créée. L'entreprise CamMobest une start-up cambodgiennespécialisée dans le développement d'applications mobiles et les sites web mobiles.

D'abord, Oopost est un service édité parOokapi Sarl, une sociétéen France, pour envoyer la carte postale personnalisée sur le site web avec la livraison en 48h.Avec l'augmentation des utilisateurs de Smartphone, on veut l'intégreravectoutes les plateformes du Smartphone:iOS, Android, BlackBerry, Nokia et Windows Phone.

Comme l'application Oopost sur iOS devient de plus en plus populaire mais les fonctionnalités existants ne sont pas très suffisants pour les utilisateurs d'envoyer la carte postale vraiment personnalisée et aussi elle contient quelques bugs, donc, l'entreprise de CamMobme demande de créer une nouvelle version d'application d'Oopost sur la plateforme iOS en ajoutant les fonctionnalités supplémentaires et éliminer les bugs existants.

Ce mémoire de fin d'études est divisé en 5 chapitres. Le premier est la présentation générale du projet de stage. La seconde porte surl'analyse et la spécification des besoins. Le troisième met l'accent sur le choix de technologie et la conception. Le quatrième est l'implémentation. Et le dernier touche la conclusion.

I. PRESENTATION GENERALE

Cette partie présente l'entreprise où mon stage a été effectué, ses activités et services, son structure globale et le contact. Elle contient également l'introduction au projet de stage. On présente aux lecteurs avec les problématiques avant le développement de l'application, l'objectif principal du projet et certains contraints reliés. Finalement, on propose la méthodologie de travail et le planning que l'on définit en avance afin de pouvoir finir le travail pendant 4 mois de stage.

1. Présentation de l'entreprise

1.1. Introduction à l'entreprise

CamMob est une start-up cambodgiennespécialisée dans le développement d'applications mobiles et les sites web mobiles. L'équipe á CamMob estspécialisée sur la conception, le développement et l'intégration des sites et les applications pour iOS, Android,Windows Phone et BlackBerry. CamMob met à la disposition dans chaque projet une équipe d'experts fonctionnels, ergonomes mobiles, architectes et développeurs avec soucis de la qualité du service rendu et de l'intégration la plus légère possible dans les plates-formes existants de ses clients.

Fondée en novembre 2009, CamMob a développé ses activitésdans le marchéd'Outsourcing, et plus particulièrement avec les partenaires européens. CamMobforme et met à la disposition de ses clients une équipe de jeunes ingénieurs cambodgiens motivés et passionnés par les technologies mobiles.

Figure 1: Logo de CamMob

1.2. Activités et services de l'entreprise

CamMob est une agence mobile qui accompagne et conseille ses clients dans la conception, le design, le développement d'applications sur les plates-formes comme ci-dessous :

- iOS (iPhone and iPad)
- Android
- Windows Phone
- BlackBerry
- Site web mobile compatible avec tout typedes téléphones dans le marché, des basesterminales aux Smartphones récents comme iOS et Android.

1.3. Situation géographique et contact

On peut contacter l'entreprise de CamMob avec les informations de contact suivant :

Adresse : #18C, rue 584, Toul Kork, Phnom Penh, Cambodge.
Téléphone : +(855)15741234
Courier électronique : info@cam-mob.com
Site d'Internet : http://www.cam-mob.com

2. Présentation de projet du stage

Obligatoirement, les étudiants en cinquième année dans tous les départements, y compris le département Génie Informatique et Communication (GIC), doivent faire un stage de fin d'études pendant trois mois au minimum au deuxième semestre. Le stage peut être fait dans un établissement public, une organisation, ou une entreprise privée.L'objective du stage est non seulement de faire l'intégration entre les études théoriques à l'école à la vie professionnelle à l'entreprise, mais encore de donner des grandes opportunités aux étudiants d'acquérir des nouveaux connaissances ainsi que des nouvelles technologies. A la fin du stage fin d'études, les étudiants doivent réaliser un mémoire pour faire la présentation de ce qu'ils ont fait.

D'après la proposition et l'accord de la convention de stage, validés par le tuteur et le maître de stage, le sujet apporte sur le thème « Application Oopost pour iPhone version 2 ». Oopost est une application utilisée pour créer la carte postale personnalisée et l'envoyer partout dans le monde entier. Puisqu'il est développé orienté sur la plate-formeiPhone spécialement, il est donc nécessairement d'implémenter le maximum possible les caractéristiques principales d'iPhone. L'utilisation de ces caractéristiques est non-seulement donner une interface qui suivre le model UX (User Experience), mais en plus garantir l'objective de notre application : « Créer une carte personnalisée à partir de votre iPhone».

Mon stage est réalisé pendant environ 4 mois du 13 Février 2012 au 22 Juin 2012 à CamMob, une entreprise privée.

J'ai été encadré par :

- Maître de stage : **Dr. SENGSopheap**

Chef du département Génie Informatique et Communication

- Tuteur de stage : **M. YISophally**

Chef du projet auCamMob

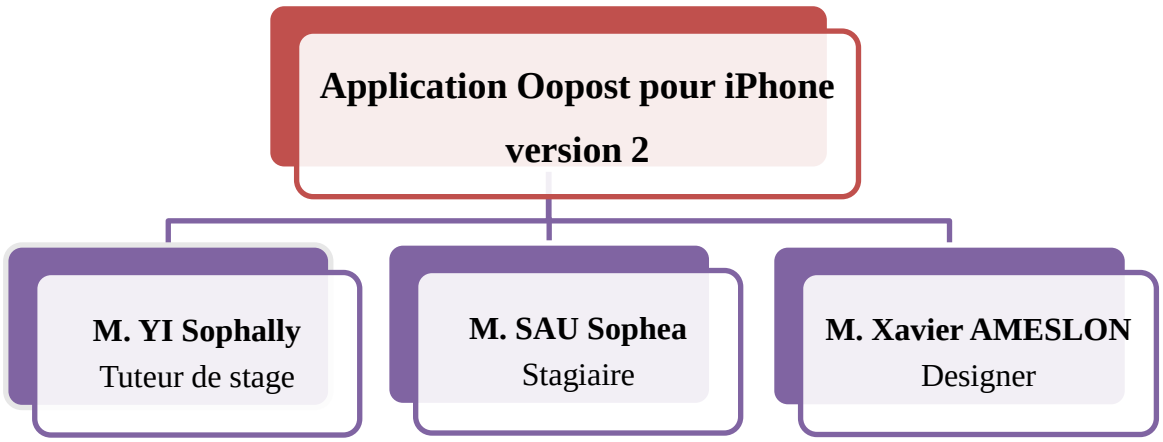


Figure 2: Organisation de l'équipe

2.1. Problématique

Comme on a mentionné dans la partie d'activités et services, l'entreprise CamMob fournit le service de développer l'application sur iOS. Beaucoup d'applications d'iOS de l'entreprise CamMob sont déjà disponible sur l'App Store, parmi lesquels y compris l'application Oopost qui sort depuis 2009.

En fait, cette application est très populaire dans le marché d'Apple et aussi elle obtient un grand succès parce qu'elle est classé par l'utilisateur 4 étoiles plus sur 5 étoiles avec beaucoup des utilisateurs courant. Par contre, cette application manque beaucoup de fonctionnalités clefs pour personnaliser la carte postale. Elle est très simple parce qu'elle contient seulement quelques opérations de base pour envoyer la carte postale.

De plus, comme dans les autres canaux de création en particulier le site web, il existe déjà beaucoup de façons pour rendre la carte plus attractive et agréable comme l'ajoute le cadre avec les styles variés et la personnalisation du message. Par conséquence, le chef du

produit nous demande de les intégrer à notre application en adaptant avec les caractéristiques de Smartphone iPhone particulièrement l'écran tactile et la taille d'écran.

Autrement, comme la technologie est développé très vite en particulier le système opératoire d'iPhone, donc il faut qu'on réactualise notre application avec cesnouvelles fonctionnalités supplémentairespour qu'elle puisse aider l'utilisateur de créer la carte plus personnalisé dans le but degarantir l'objective de notre application: « Créer une carte personnalisée à partir de votre iPhone».

2.2. Objectif

A cause de la problématique au-dessous, l'objectif de notre projet est de mettre à jour une application Oopost pour la plateforme iPhone vers une nouvelle version. À fin de résoudre ces problèmes, on doit penser essentiellement aux 2 choses qui sont les erreurs et l'ajoute des fonctionnalités supplémentaires.

En testant Oopost pour iPhone version 1, on a vu des erreurs qui doivent être corrigé effectivement. On doit savoir ses causes pour qu'on puisse les éliminerdans cetteprochaine version.

C'est aussi important d'intégrer quelques fonctionnalités existantes dans le site web d'Oopost pour la création de la carte à notre application essentiellement la personnalisation du message et la gestion du compte. Pour la première, on doit permettre utilisateur de changer la taille du message et aussi de changer son couleur, sa police de caractère et son alignement. Pour la dernière, ce qu'il manque dans la dernière version, c'est la modification du compte, l'achat des crédits et le suivi de commande. Donc, l'utilisateur n'a pas besoin d'aller au site web d'Oopost pour faire ces opérations. Par conséquence, tous les 2 fonctionnalités doivent être existées dans l'application Oopost version 2.

Il est également nécessaire de rajouter les fonctionnalités requis par l'entreprise pour permettre l'utilisateur d'envoyer la carte plus personnalisée comme l'importation de photo depuis les réseaux sociaux, la décoration ducôtérecto et plusieurs destinataires envoyés. Les réseaux sociaux choisis pour récupérer la photo sont Facebook, Picasa, Flickr et Instagram. En plus, après avoir une photo pour mettre sur le côtérecto, on peut appliquer le cadre et l'effet aussi. Un autre point important dans cette version est qu'on peut partager une carte vers plusieurs destinataires à la fois.

De plus, en utilisant des conceptions de based'Oopost pour iOS version 1 et enajoutant le principe de la personnalisation de carte par les fonctionnalités supplémentaires au-dessus, cette nouvelle application va donner la possibilité de créer des cartes plus personnaliséesà partir d'iPhone que la dernière version.

2.3. Contraint

L'applicationOopost sur iOS version 2 doit intégrer avec les bases données existantes sur le serveur Oopost et réutiliser l'API étant déjà en service actuellement pour toutes les plates-formes (iOS, Android, BlackBerry et Nokia).

3. Méthodologie de développement

Pour développer une application plus efficace, on doit avoir une méthodologie de travail. Lorsqu'il existe beaucoup de méthodologies de développement, on doit choisir un parmi lesquelles.

Comme mon stage est effectué à l'entreprise CamMob, ce projet est utilisé la méthodologie de développementScrum parce que c'est une méthode qui a choisi par l'entreprise pour développer tous les applications.

3.1. C'est quoi le Scrum ?

Scrumest une méthode agile dédiée à la gestion de projetsqui existe beaucoup des avantages comme ci-dessous :

- Le travailitérative et incrémentale
-Une productivité plus élevé
-Amélioration engagement de l'équipe et lasatisfaction du travail
-Réduction du temps demise sur le marché
-Concentrer surla satisfaction du client

Figure 3: Le processus de Scrum

3.2. Les rôles et outils impliqués dans la méthodologie

Il y a 3 rôles principaux qui participent dans le déroulement de développement :

- Le propriétaire du produit =Le propriétaire du produit est le représentant des clients et des utilisateurs et ce qui écrit des User Stories, les priorise et les ajoute au Product Backlog. Il sait la logique de business et les souhaits de l'utilisateur.Son objectif est de maximiser la valeur du produit développé.

- L'équipe = Celui qui est chargée de distribuer le produit avec croix-fonctionnelle compétences et qui font le travail réel (analyser, concevoir, développer et tester). En plus il doit être l'auto-organisation et autodirigé.

- ScrumMaster = Il est responsable de la méthodeScrum. Il doit s'assurer que celle-ci est comprise, et bien mise en application. Ce n'est pas un chef de projet, ni un intermédiaire de communication avec les clients En tant qu'un facilitateur, il aide l'équipe à déterminer quelles interactions avec l'extérieur sont utiles, et lesquelles sont obstacles. Il aide alors à maximiser la valeur de produit par l'équipe.

ProductBacklog = c'est la liste des User Stories avec la priorité qui définit par le propriétaire du produit.

User Story = il concentre sur le souhait d'utilisateur. La forme d'User Story peut être différente de l'un à l'autre.

EX : - En tant que <type d'utilisateur>, je veux <quelques buts> pour <réalisation>.
- Afin de <réalisation >, comme <type d'utilisateur>, je veux<quelques buts>.

Sprint Backlog = il estla liste des User Stories avec la priorité qui sélectionne à partir de Product pour chaque Sprint.

3.3. Comment le Scrum marche ?

Après avoir analysé les besoins qui sont demandés par le client, le propriétaire de produit va créer le Product Backlog qui contient les User Stories avec les priorités.

D'après le Product Backlog, le ScrumMaster et l'équipe vont faire la réunionde planification de Sprint pour sélectionner les travaux ou bien les User Stories à faire selon les priorités et lesmettredans le Sprint backlog avec le détail du temps qu'il faudra pour le faire. La moyenne pour définir le période pour chaque élément (User Story) de Sprint est faiteselon le Planning Poker. Le Planning poker est une méthode que l'équipe et le ScrumMasterse discutent à fin d'estimer le temps pour finir chaque fonctionnalité. Normalement, le ScrumMaster préfère le plus court, en revanche l'équipe veut plus que ça, donc l'équipe doit exprimer les raisons pourquoi ça prend beaucoup de temps.

Donc on a le Sprint Backlog qui est la liste destravaux que l'équipedoit tenir comptelors du Sprintsuivant. Après le Sprint Backlog est fait, l'équipe doit le suivre particulièrement le temps définit pour chaque User Story.

Tous les jours, le ScrumMaster et l'équipe doit avoir une réunion qui s'appelle Scrum quotidien.Scrum quotidien est une technique très efficace de contrôler pendant le développement d'application.Pendant 15 minutesau maximum, le ScrumMaster pose quelques questions comme ces 3 questions :Qu'est-ce que avez-vous faitdepuis hier?Qu'est-ce vous allezfaire aujourd'hui?Avez-vous desproblèmesquivous empêchent del'accomplissement de votreobjectif?

Ensuite, quand le temps pour le premier Sprint est atteint, le ScrumMaster doit faire une autre réunion qu'on appelle la réunion de révision de Sprint. Ce qu'on doit faire dans cette réunion est de réviser le travailqui est terminé et n'est pas terminé et faire unedémo pour

valider les fonctionnalités qui ont fait. Après la fin de chaque Sprint on doit obligatoirement obtenir un produit qui peut être fait une démo pour montrer au client l'avancement du projet. Pour le travail qui n'ont pas encore terminé, qu'on a 2 possibilités : on peut mettre ça dans le Sprint suivant s'ils sont les fonctionnalités principales ou bien on peut refaire ça après on a terminé les autres.

Donc, on a le Sprint suivant qui peut être contient les fonctionnalités de dernier Sprint ou pas. On refaire les étapes au-dessus jusqu'à tous les fonctionnalités dans le Product backlog sont tout à fait bien.

En plus, la méthodologie Scrum donne la possibilité d'évaluer notre travail par nous-mêmes ou bien par le ScrumMaster. Ça peut être apparaît par un outil qu'on appelle Burndownchart quiest un graphiqueaffiché publiquement pourmontrerle travailrestantdans leSprintbacklog en comparant avec le temps. Ce graphique doit être mit à jourtous les jours pour qu'il donneune vision simple del'état d'avancementdu Sprint.

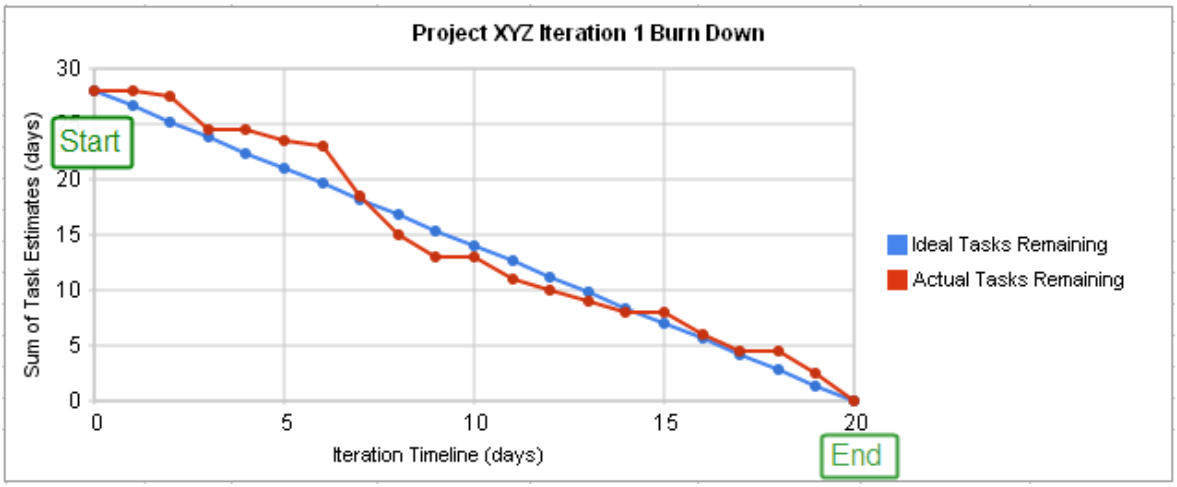


Figure 4: Burndownchart

4. Planning

Le stage est effectué pendant 4 mois (16 semaines en total). Le planning est donc proposé après que l'on a identifié l'objectif principal de ce stage et que l'on avait le Product Backlog du projet :

Tâche	Semaine												
	1 - 4	5	6	7	8	9	10	11	12	13	14	15	16
Études des nouvelles technologies													
Études des applications existantes													
Sprint 1-7			S1	S2	S3	S4	S5	S6	S7				
Teste et débogage													

Tableau 1: Planning

Sprint 1 = On étude les besoins et le scénario globale du système pour savoir quelles sont les technologies et les techniques de programmation utilisés dans notre projet. En plus, on fait le menu principal.

Sprint 2 = Il contient toutes les fonctionnalités concernant le côté recto de la carte postale comme les opérations sur l'image, et l'ajoute du cadre et l'effet.

Sprint 3 = Il concentre sur le côté verso en particulier la personnalisation du message comme le changement de taille du message, le couleur, l'alignement et la police de caractère.

Sprint 4 = La partie d'ajouter des destinataires est effectuée dans ce Sprint. On crée la forme pour remplir et on permettre à l'utilisateur de récupérer les informations à partir de l'Address Book aussi.

Sprint 5 = Ce Sprint contient toutes les fonctionnalités concernant la gestion de compte comme la création et la modification du compte. En plus, on crée la fonction pour afficher le suivi de commande et acheter crédit dans ce sprint aussi.

Sprint 6 = C'est le dernier Sprint qui contient les fonctionnalités pour envoyer la carte au serveur web d'Oopost en coupant la partie d'image qui est utilisé pour mettre surs la carte. Pour envoyer la carte, on utilise l'API existant.

Sprint 7 = Il contient les fonctionnalités restants de tous les Sprint avec quelques fonctionnalités supplémentaires.

II. ANALYSE ET SPECIFICATION DES BESOINS

1. Scénario globale

Tout d'abord, l'application commence en affichant un écran de démarrage pendant quelques secondes avec une animation de Fade-in sur le logo Oopost.À ce moment, s'il y a une carte en cours de création, l'application va alerter à l'utilisateur. Dans ce cas, s'il l'utilisateur veut continuer, on initialise l'état de l'application comme le moment qu'il a quitté à la dernière fois par exemple la modification de l'image, la personnalisation du message et les informations des destinataires, et on affiche la page de la création du carte directement. Par contre, s'il n'y a pas carte sauvegardé ou le choix de l'utilisateur est de supprimer la carte en cours de création, l'application reste au menu principale.

Au menu principal, l'utilisateur a le choix d'importer les photos comme à partirde l'appareil photo, la pellicule et les réseaux sociaux comme le Facebook, Flickr, Instagram et Picasa qui sont très populaire dans le monde. Après avoir choisi une photo, on va arriver à la page de la création de carte de côté recto avec l'image choisi. Dans ce côté, l'utilisateur a possibilité d'ajouter le cadre selon les catégories et d'ajouter les effets comme noir et blanc, sépia et vintage. Sinon il peut continuer au côté verso pour saisir le message. Dans le côtéverso, on a aussi la possibilité de personnaliser le message en changeant la taille, la police de caractère, l'alignement et le couleur. Ensuite, on va au page pour ajouter un ou plusieurs destinataires manuellement ou à partir de contact.

Après avoir fini la création, l'utilisateur peut bien sûr envoyer au serveurmaintenant. Maiscela doit être authentifié, et s'il n'a pas le compte, il peut le créer. Avant d'envoyer la carte, il faut que l'utilisateur ait des crédits suffisants. On peut lui proposer d'aller acheter des crédits pour envoyer dans le car qui n'y a pas assez crédit restant. De plus, l'utilisateur peut utiliser le codePromo valide pour envoyer la carte aussi. Le codePromo est le code qu'on obtient à la première fois qu'on crée le compte ou après avoir gagné le jeu d'Oopost.

Après avoir envoyé, la carte créée peut être réutilisée pour envoyer aux autres ou bien modifiée, sinon on peut créer la nouvelle carte en retournant au menu principal.

En plus, dans le menu principal, utilisateur a le droit de gérer son compte s'il existe. Sinon, il peut créer et faire la connexion. Après avoir connecté au serveur Oopost, pour la gestion de compte, l'utilisateur peut afficher l'histoire de commande, acheter crédits et modifier

son compte. A chaque fois que l'utilisateur faire la connexion, l'application doit enregistrer des informations pour la prochaine connexion. L'enregistrement des informations de connexion ne peut que détruit dans le cas l'utilisateur déconnecte du serveur ou supprime l'application. Donc à la prochaine fois, l'utilisateur va arriver à la page gestion de compte, ce n'est pas la page connexion.

Un diagramme d'activités nous permet de modéliser un processus interactif, global ou partiel pour un système.

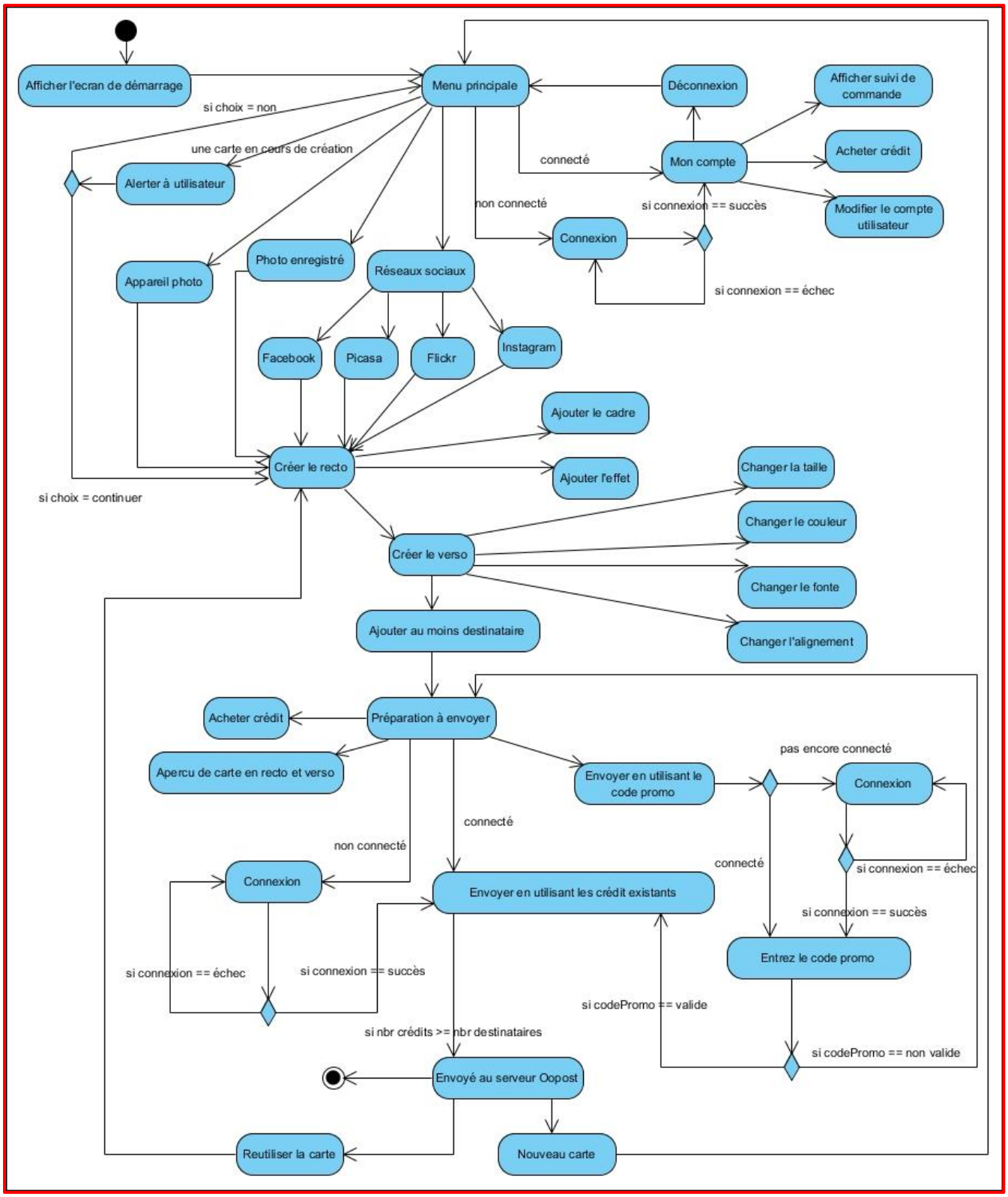


Figure 5: Diagramme d'activité présentant le scénario globale du système

2. Etude de besoins

2.1. Besoins fonctionnels

Les Besoins fonctionnels sont les besoins demandés par notre chef du produit et ces sont les fonctions qui doivent être implémentées. Il est très important de bien définir des besoins fonctionnels car ils doivent répondre et garantir l'objective du projet. En utilisant ces besoins et la référence de l'objective du projet, on peut définir des besoins comme la présente suivante:

- Afficher l'écran de démarrage avec animation
- La connexion automatique avec les informationsenregistrées
- Connexion et déconnexion
- Administrer le compte comme la création et la modification
- Afficher l'histoire de commande
- Afficher une page web pour acheter crédit
- Aperçu de la carte des 2 côtés: recto et verso
- Prendre une photo via appareil photo
- Prendre une photo de bibliothèque ou bien les photos enregistrées
- Prendre une photo à partir des réseaux sociaux comme Facebook, Flickr, Instagram et Picasa
- Faire les opérations sur la photo comme redimensionner, pivoter et déplacer
- Appliquer le cadre selon les catégories de manière PageFlow
- Appliquer l'effet comme noir et blanc, sépia et vintage
- Personnaliser le message comme la taille, l'alignement, le couleur et la police de caractère
- Importer les informations desdestinataires à partir du contact
- Envoyer la carte avec les crédits restants et le code promo
- Sauvegarder et restaurer la carte
- Faire le Push Notificationà l'utilisateur d'Oopostpour transmettre les informations
- Evaluer application sur l'App Store
- Envoyer la newsletter par e-mail à tous les clients d'Oopost
- Couper la photo pour l'envoyer au Serveur Oopost
- Parrainer ami par envoyer l'e-mail

2.2. Besoins non fonctionnels

Les besoins fonctionnels sont notés dans la partie précédent, alors il existe aussi un autre type besoin appeléles besoins non-fonctionnels. Les besoins non-fonctionnelles sont les besoins qu'on définit pour améliorer le système. Ils sont:

- Ergonomie: le système doit être uniforme et facile à utiliser avec une interface agréable.
- Maintenance et extensibilité: le système doit être bien organisé en modules indépendants pour être facilement maintenu et extensible.
- Performance: la vitesse rapide avec le déroulement confiance sans erreurs.
- Multi-langue: l'application peut choisir la langue de manière automatique en faire référence au réglage du système d'iPhone. Deux langues nécessaires sont: le français et l'anglais.

2.3. Cas d'utilisation

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en écrire les exigences fonctionnelles. Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devraient interagir avec les utilisateurs pour atteindre un but ou une fonction spécifique d'un travail.

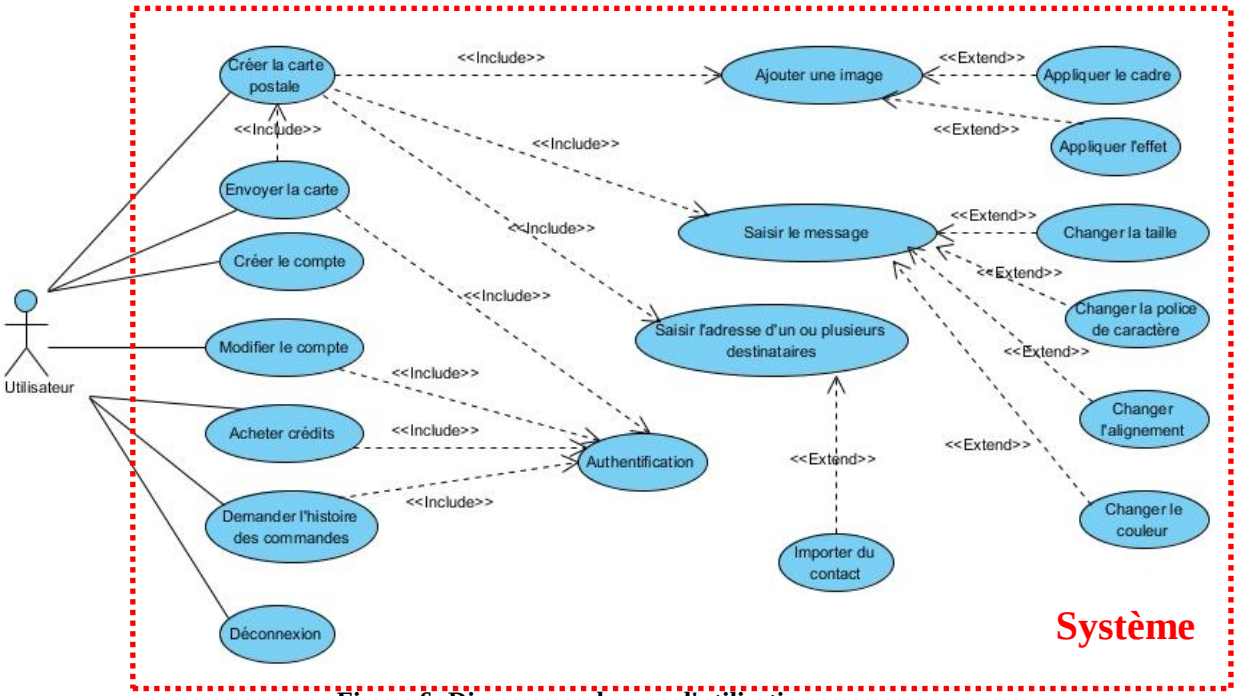


Figure 6: Diagramme de case d'utilisation

2.3.1. Acteur

Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe que l'on tente de définir. Dans notre cas, on a seulement un utilisateur qui est un utilisateur d'iPhone.

2.3.2. Spécification de chaque cas d'utilisation

Dans cette partie, on va parler en détail ce qu'on peut faire dans chaque cas d'utilisation et définir les fonctionnalités dedans.

2.3.2.1. L'envoi de la carte postale

L'envoi de la carte postale est une fonctionnalité très important qu'on ne peut pas la manquer parce que c'est l'objectif principale de notre projet. En fait, pour envoyer la carte postale, on doit avoir une carte avec les 2 côtés: recto et verso. En plus pour pouvoir l'envoyer, votre propre compte d'utilisateur doit bien créer pour que l'entreprise puisse profiter de ce service parce que c'est unoutil d'acheter des crédits, sans le compte sans le droit d'achat. Donc à chaque l'envoi sauf ils ont connecté, ils doivent faire la connexion pour être autorisé d'envoyer la carte.

2.3.2.2. La création de la carte postale

En réalité, une carte postale consiste de 3 parties principales qu'on doit remplir obligatoirement comme le recto, le verso et le destinataire. Donc c'est le même dans notre application, on a transformé ces concepts classiques vers l'application Oopost version 2. C'est-à-dire, tout d'abord on doit avoir une image pour lecôté recto, après on doit saisir des messages pour exprimer vos émotions. De plus, on doit savoir l'information des destinataires pour qu'on puisse les envoyer.

2.3.2.3. L'importation de la photo

Dans l'application Oopost version 2, on a beaucoup de ressources possibles pour récupérer l'image comme l'appareil photo, les photos enregistrées et les réseaux sociaux. Le dernier est un nouveau type de ressource qu'il n'existe pas dans le Oopost version 1. Les réseaux sociaux disponibles dans notre application sont Facebook, Flickr, Picasa et Instagram, ces sont les réseaux sociaux qui sont très célèbre et populaire dans le monde. Donc c'est une bonne intégration entre les réseaux sociaux et le processus de la création de carte postale pour rendre notre application plus célèbre et plus utile.

2.3.2.4. L’ajoute du cadre

Comme on a vu dans le diagramme de cas d’utilisation, l’ajoute du cadre est une fonctionnalité facultative, c’est-à-dire l’utilisateur a le droit de décider de mettre le cadre sur le photo ou pas. Il y a 6 thèmes qu’on peut choisir un parmi lesquelles. Dedans, il y a beaucoup des cadres qui sont adaptés à tous les cartes. La présentation des cadres est faite en mode PageFlowqui rendre le choix de carte plus facile parce qu’on peut tester tous les cadres directement quand on fait défiler.

2.3.2.5. L’ajoute de l’effet

On vous donne la possibilité d’appliquer 3 effets sur la photo choisi comme le noir et blanc, la sépia et le vintage pour rendre la photo plus attractive. En plus, on peut revient à l’état originale en cliquant sur l’effet normale.

2.3.2.6. La saisiedu message

Pour le message, l’utilisateur peut saisir le message au maximum 12 lignes qui peut être personnalisé par la taille, le couleur, la police de caractère et l’alignement. On peut varier la police de caractère dans l’intervalle de 10 à 30 selon notre besoin. Il y a 5 couleurs possibles pour nous de choisir comme noir, orange, bleu, verte et rouge. Pour la police de caractère, on peut choisir 1 parmi les 4 polices de caractère : Arial, Segoe Print, Traveling Typewriter et Bell MT. En plus, on peut aligner notre message à droit, centre et gauche.

2.3.2.7. La saisie des destinataires

Pour envoyer la carte postale, l’adresse des destinataires doit être indiquée avec le nom. Pour saisir cette information, utilisateur peut faire manuellement ou à partir du contact existant qui contient toutes ces informations-là. En plus, pour satisfaire le besoin d’utilisateur d’envoyer aux plusieurs destinataires, on est possible d’insérer plus qu’un destinataire.

2.3.2.8. La création du comte d’utilisateur

C’est obligatoire d’avoir une compte d’utilisateur pour envoyer la carte, donc si utilisateur n’a pas encore le compte, il peut le créer dans notre application en remplissant les informations nécessaires comme le nom, le courriel électronique et le mot de passe, il n’a pas besoin d’aller à site web Oopost pour créer le compte.Après avoir créé le compte,

l'utilisateur peut l'utiliser immédiatement en faisant la connexion, acheter des crédits et envoyer la carte.

2.3.2.9. La modification du compte

Pour la gestion du compte hors de la création du compte, on peut modifier aussi le compte mais ce n'est pas toutes les informations peuvent être modifié, on laisse de modifier seulement le mot de passe. Donc les autres informations comme le nom d'utilisateur et le courriel électronique ne peuvent pas changer.

2.3.2.10. L'achat des crédits

On a créé une page web pour faciliter d'utilisateur d'acheter des crédits à partir de son iPhone, il n'a pas besoin d'acheter le crédit sur l'ordinateur. Dans la page web, on laisse utilisateur de choisir les paquets qu'il veut, et après cliquer sur ces paquets, il va être lié à l'autre page web sécurisée pour insérer le numéro de carte, la date de fin de validité et cryptogramme visuel (3 dernier chiffres au dos de la carte). Cette page web est sécurisée par le système Paybox qui nous garantit que nous utilisons le standard SSL pour sécuriser vos connexions d'internet. Après avoir achetéles crédits, l'application va mis à jour automatiquement, donc quand on retour à l'application, on peut l'envoyer immédiatement si le crédit est plus grand que le nombre des destinataires.

2.3.2.11. Le suivi de commande

Suivi de commande est une fonctionnalité qui nous permet de savoir l'histoire de notre commande depuis la création du compte. On va afficher quelques informations comme le numéro de commande, les détails (la date et la destinataire) et le statut (envoyée ou en cours de traitement).

2.3.2.12. La déconnexion

Normalement, à chaque fois l'utilisateur faire la connexion, l'application a récupéré les informations nécessaires pour la prochaine fois, donc on n'a pas besoin de faire la connexion encore une fois sauf on fait la déconnexion. La déconnexion est une fonctionnalité de déconnecter de serveur Oopost en supprimant ces informations à partir de votre iPhone, donc quand on veut envoyer la carte, on doit faire la connexion encore une fois.

III. CHOIX DE TECHNOLOGIE ET CONCEPTION

1. Choix de technologie

Cette partie présente de la technologie que l'on a utilisée pour réaliser notre projet. Elle porte aussi sur les langages et les outils qui aident au développement de l'application.

1.1. Présentation de la plate-forme iPhone

iPhone est une Smartphone conçue et commercialisée par Apple Inc. depuis 2007. Les modèles, dont l'interface utilisateur a été conçue autour du multi-touch, disposent d'un appareil photo, d'un client Internet (pour naviguer sur le Web ou consulter son courrier électronique), et de fonctions basiques telles que les SMS et les MMS.

Figure 7: iPhone 3G/3GS

Figure 8: iPhone 4/4S

1.1.1. Caractéristique logicielles d'iPhone

L'iPhone compte plusieurs applications intégrées, dont Safari, Mail, Photos, Vidéo, YouTube, iPod, iTunes, App Store, iBooks, Cartes, Notes, Agenda, Contacts et recherche Spotlight. Apple a également porté sa suite iWork, et vend séparément ses applications Keynote, Pages et Numbers.

Tous les appareils tournant sous iOS peuvent être jailbreaké, permettant à des applications qui ne sont pas autorisés par Apple de fonctionner sur l'appareil. Une fois jailbreaké, les utilisateurs de l'iPhone peuvent télécharger de nombreuses applications inédites sur l'App Store via des installateurs non officiels tels que Cydia, mais aussi de télécharger des applications illégalement piratées.

1.1.2. App Store

L'App Store est une plateforme de téléchargement d'applications, similaire au Google Play, distribuée par Apple sur les appareils mobiles fonctionnant sous iOS (iPod Touch, iPhone et iPad) depuis le 11 juillet 2008.Ce logiciel s'intègre dans le service iTunes Store et permet de télécharger des applications tierces pour l'iPod Touch, l'iPhone et l'iPad.

À sa mise en ligne, le prix médian est de 0,99 USD et la moyenne de 3,33 $. La répartition est aussi intéressante, 25 % des logiciels sont gratuits et 90 % sont inférieurs à 9,99 USD. En juillet 2009, soit un an après l'ouverture de l'App Store, le nombre d'applications disponibles dépasse les 85 000. Le 6 janvier 2010, Apple annonce avoir dépassé les 3 milliards de téléchargements d'applications iPhone. Début mars 2010, plus de 150 000 applications, développées par plus de 28 000 développeurs, sont disponibles au téléchargement sur l'App store (au niveau mondial).Le 7 juin 2010, Apple annonce que 225 000 applications sont disponibles sur l'App store. Chaque semaine, 15 000 applications sont soumises et 95 % sont acceptées. Depuis le lancement de l'App store 1 milliard de dollars a été reversé aux développeurs.

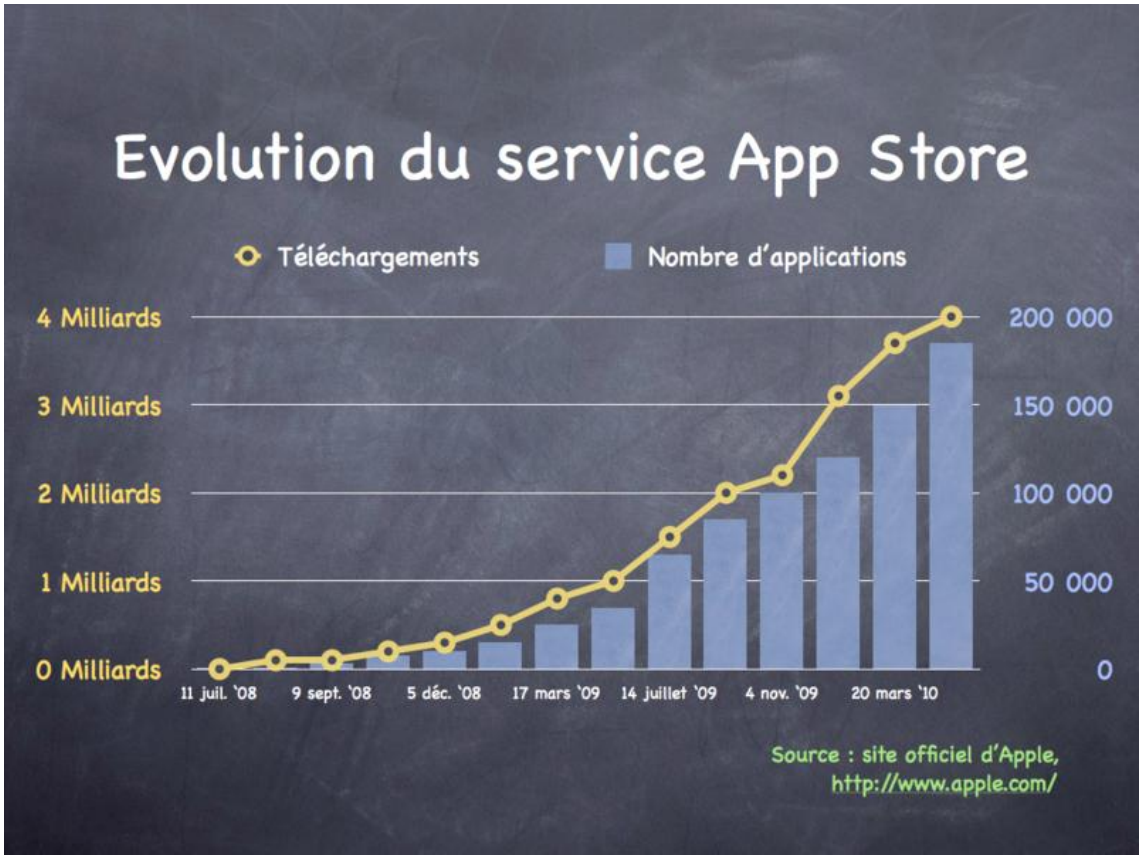


Figure 9: Evolution du service App Store

1.1.3. Les résultats commerciaux

Les ventes mondiales del'iPhonepar trimestreen millions

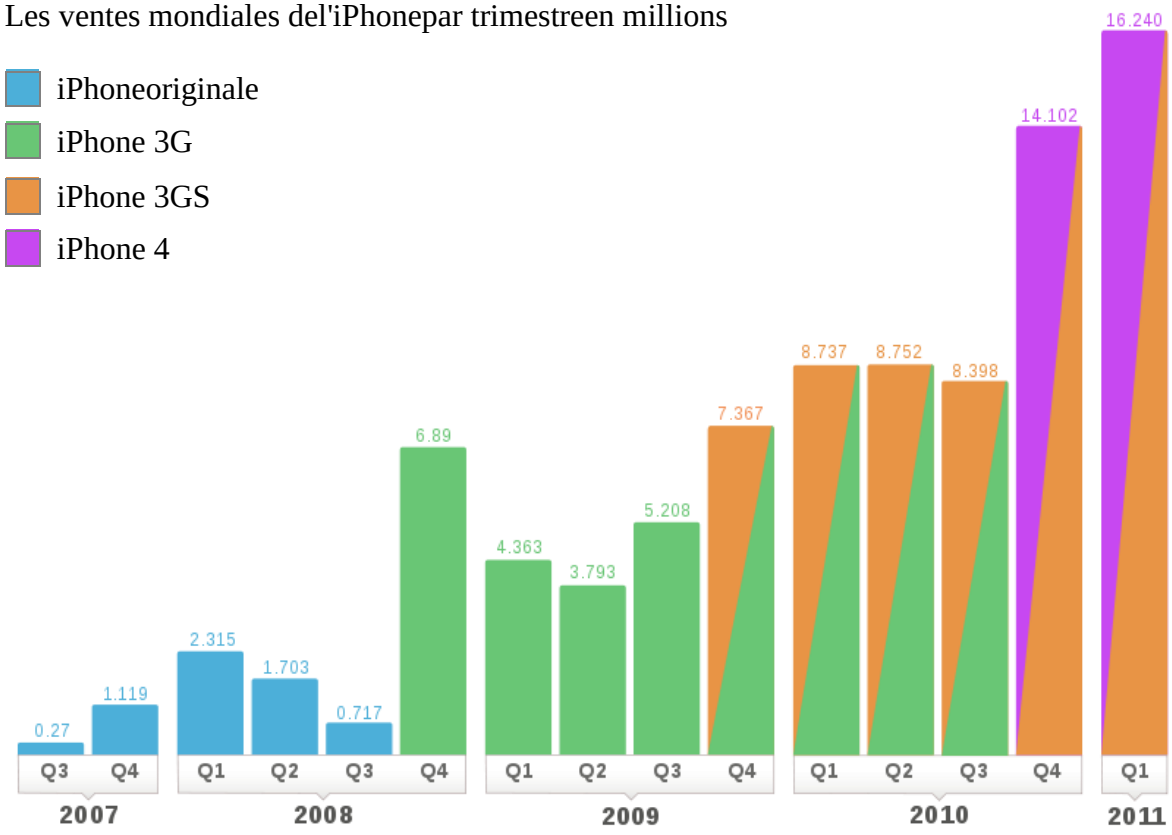


Figure 10: Les résultats commerciaux d'iPhone

Comme on a vu dans le graphique au-dessus, l'iPhone est de plus en plus célèbre et est utilisé par une énorme quantité d'utilisateur. Il augment très rapidement de 270 milles dans le troisième trimestre 2007 à 16 240 millions dans le premier trimestre 2011. Lorsqu'il y a le baisse dans quelques trimestres mais l'Apple traite bien ce problème. Il leressoude en créant un nouveau modèle de l'iPhone pour bien contrôler le marché et rendre le nombre de vent augmenté de façon significative.

1.2. Les technologies et les outils utilisés

Cette partie présente en bref des technologies que l'on utilise pour développer cette application. Elle couvre aussi les outils que l'on utilise pour réaliser le projet et préparer le rapport.

1.2.1. Les technologies utilisées

1.2.1.1. iOS

iOS, anciennement iPhone OS, 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 dont il partage les fondations (le kernel hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). 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.

Le support d'application utilisé sur l'iPhone et l'iPod touch est basé sur une architecture ARM contrairement aux processeurs utilisés sur les anciennes versions des ordinateurs Apple (PowerPC) ou au récents (Intel x86). De plus, iOS utilise l'API OpenGL ES tournant sur une carte graphique 3D double cœurs PowerVR. En somme, les applications développées sous Mac OS X ne peuvent pas fonctionner sur un iPhone ou un iPod Touch, toutes les applications natives sont redéveloppées spécifiquement pour l'architecture ARM et les composants logiciels d'iOS.

Les différentes versions d'iOS depuis la sortie:

iOS 1 : iOS initial sort depuis 29 juin 2007.
iOS 2 : Il sort depuis 11 juillet 2008 et il est disponible pour iPhone 3G
iOS 3 : Il sort depuis 17 juin 2009 et il est utilisé dans iPhone 3GS
iOS 4 : La sortie est en 21 juin 2010. iPhone 3G/3GS et iPod touch le utilise pour son système opératoire.
iOS 5 : Au 6 juin 2011 jusqu'à maintenant, cet iOS est disponible pour iPhone 3GS/4/4S et iPad, iPad2 et new iPad.
iOS 6 : il a été aperçu en 11 juin 2012, mais tous les appareils tournant par iOS ne sont pas encore utilisé cette version.

1.2.1.2. iOS SDK

Le kit de développement iOS SDK a été officiellement annoncé le 6 mars 2008 par Apple, lors d'une présentation communément appelée « Apple 6 March Event ». Il est compatible uniquement avec Mac OS X, donc c'est-à-dire seulement la machine Mac peut développer

l’application sur iPhone. La première version Beta du SDK a été disponible immédiatement avec la version 1.2b1 iOS, aussi appelée iOS 2.0 (build 5A147p), sans que les applications développées sur la plate-forme soient alors distribuables.

En plus, le SDK contient de nombreux outils facilitant le développement et le test d'applications pour iOS. Voici une liste non exhaustive des API principales contenues dans le SDK, classées par couche d'abstraction:

- Core OS, couche la plus « profonde », contient les bases du système d'exploitation comme le Kernel d'OS X, gestion de la charge du processeur en fonction de la batterie, « Lib System », le système de librairies, Le système de sécurité.
- Core Services propose des API de plus haut niveau, permettant une gestion plus poussée du système comme Core Location, qui permet la géolocalisation de l'appareil, la gestion d'un carnet d'adresse, l'accès à des fichiers,SQLite (bibliothèque permettant la gestion de bases de données), des utilitaires pour la gestion d'URL.
- La couche Media gère quant à elle les données multimédia. Son contenu est accéléré matériellement pour de meilleures performances et une meilleure durée de batterie comme Core Audio, prise en charge des formats d'image JPEG, PNG, TIFF,Prise en charge du format PDF,Core Animation, la gestion du « Video Playback »,
- CocoaTouch est une réécriture de l'interface graphique Cocoa de Mac OS X, adaptée cette fois à l'interface Multitouchd'iOS comme la gestion des événements Multitouch, la gestion des alertes.

1.2.1.3. Les langages utilisés

Maintenant, on va parler en bref tous les langages que l’on a utilisés pour développer l’application:

- **Objective-C**: est un langage de programmation orienté objet réflexif. C'est une extension du C ANSI, 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 des moyens de combiner les avantages de C++ et d'Objective-C.Aujourd'hui, il est principalement utilisé dans deux systèmes

d'exploitation. L'un est Mac OS X d'Apple (et son dérivé iOS), basé sur la bibliothèque de classes Cocoa, l'autre est GNU avec sa bibliothèque de classes libre GNUstep. Il est le langage le plus utilisé pour réaliser notre projet parce qu'il est un seul langage de programmation qui peut être utilisé pour développer des logiciels sur iOS plateforme.

- **PHP :**un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur. Il est utilisé pour modifier et créer la page web pour qu'il s'adapte à notreapplication spécialement le Backend étant l'API. Aussi on utilise ça pour créer le script pour envoyer la newsletter. De plus, il est utilisé pour créer un fichier PHP pour faire la Push Notification du serveur Oopost.
- **HTML :**est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. HTML permet aussi de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias comme des images, des formulaires de saisie, et des éléments programmables tels que des applets. Il est utilisé pour créer la page web en particulier la newsletter et modifier la page web existant pour qu'il puisse adapter à notre besoin.
- **CSS :**est un langage informatique qui sert à décrire la présentation des documents HTML. Il est utilisé au moment où on crée la newsletter.
- **SQL :** est un langage utilisé pour envoyer la requête structurée au serveur de base de données. il est utilisé pour récupérer les informations d'utilisateur d'Oopostà partir du serveur Oopost pour qu'on puisse envoyer la newsletter.

1.2.2. Les outils utilisés

1.2.2.1. Xcode

Xcode est un environnement de développement intégré (IDE) qui fournit tous les outils dont développeur avez besoin pour créer et gérer les projets iOS et les fichiers source, de construire leur code dans un fichier exécutable, et exécuter et déboguer leur code soit dans iPhone simulator ou sur un périphérique.L'iPhone SDK y ajoute les librairies de développement pour iOS.

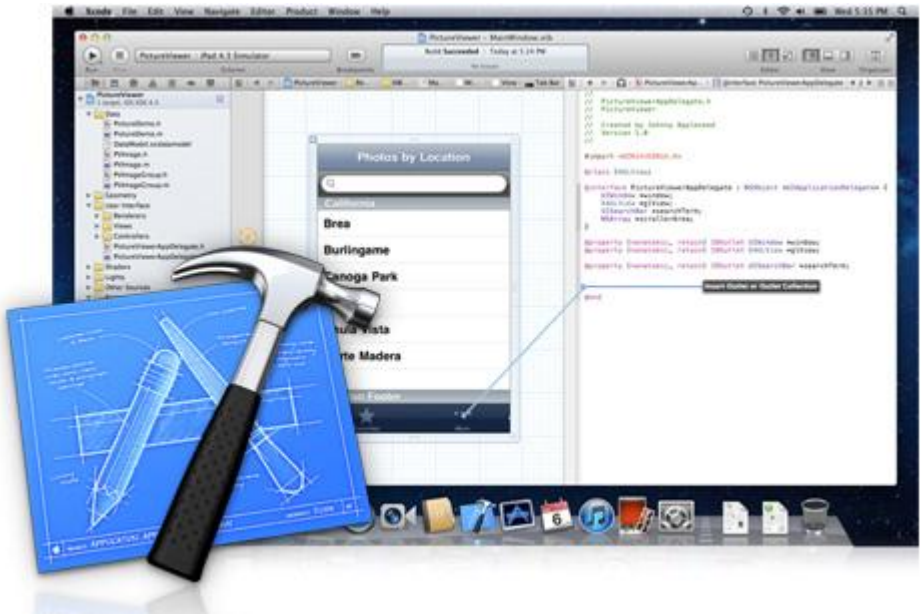


Figure 11: L'interface graphique d'Xcode

1.2.2.2. Interface Builder

Interface Builder permet de construire une interface pour CocoaTouch manuellement, àl'aide de glisser-déposer. Il permet également de traduire facilement une application dansplusieurs langues. De plus, il permet de gérer visuellement le schéma Modèle-Vue-Contrôleur, en connectant des éléments d'une interface à un code écrit pour eux aupréalable, à l'aide d'un glisser-déposer. Finalement, le fichier d'interface ainsi créé estajouté au projet Xcode.

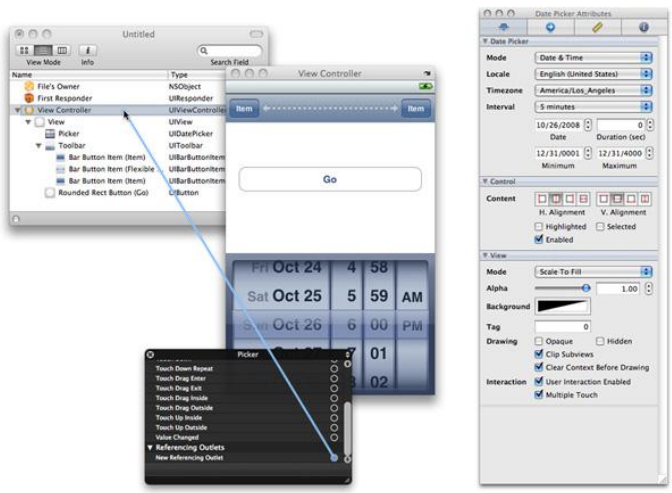


Figure 12: Interface Builder

1.2.2.3. Instruments

Instruments est un outil de monitoring informatique. Il permet, une fois l'application lancée sur un iPhone ou iPod Touch branché à l'ordinateur, d'observer en temps réel ses performances au niveau du processeur, mais également, par exemple, du moteur graphique ou de l'accéléromètre. Par ailleurs, il est également possible de surveiller les performances du système dans iPhone Simulator.

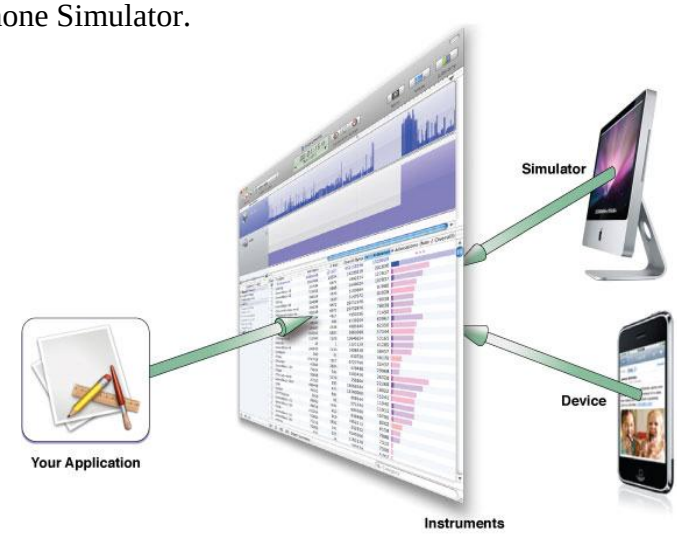


Figure 13: Instruments

1.2.2.4. iPhone Simulator

iPhone Simulator (anciennement Aspen Simulator) est le seul de ces outils à avoir été développé spécifiquement pour l'iPhone SDK. Il simule de manière logicielle un iPhone virtuel, qui peut exécuter des applications directement sur l'ordinateur. Les mouvements Multitouch sont alors reproduits manuellement à la souris par l'utilisateur, et il est possible de faire pivoter le simulateur grâce à des raccourcis clavier. Par ailleurs, l'utilisateur est en mesure de choisir quel matériel et version du Firmware il désire utiliser.

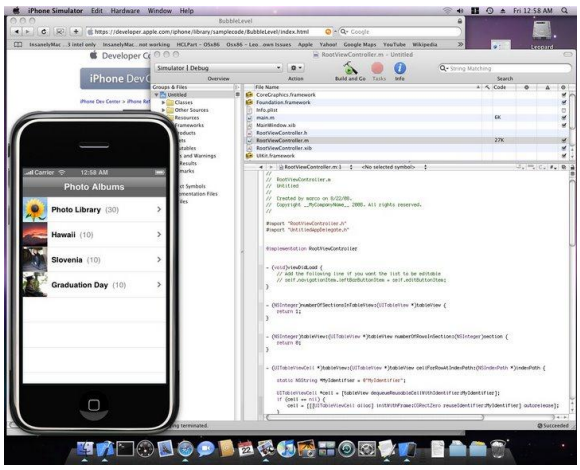


Figure 14: iPhone Simulator

1.2.2.5. Les autres outils

Tout au long de notre projet, il y a aussi autre outils utilisés pourréaliser le travail au-dessous :

- **Adobe Dreamweaver CS5.5 :** éditeur de texte pour aider à écrire le code source
- **Adobe Photoshop CS5 :**outil très puissant pour dessiner et éditer l'image
- **FileZilla :** outil pour télécharger et télétransmettre le code source au serveur.
- **Microsoft Word 2010 :** est utilisé pour la réduction de document concernant le projet tel que le cahier de charge, le manuel, le rapport ...etc.
- **Navigateur web :**Google Chrome, utilisé pour faire la recherche et également pour faire le test de l'application.
- **Visual Paradigm for UML 8.0 :** est un outil permet de faire la conception d'UML dans le développement du projet.
- **Google docs :** il est utilisé pour le travail en collaboration.

2. Conception du système

2.1. Architecture physique du système

L'architecture physique donne l'idée globale comment le système fonctionne etelle présente les composants physiques participants.

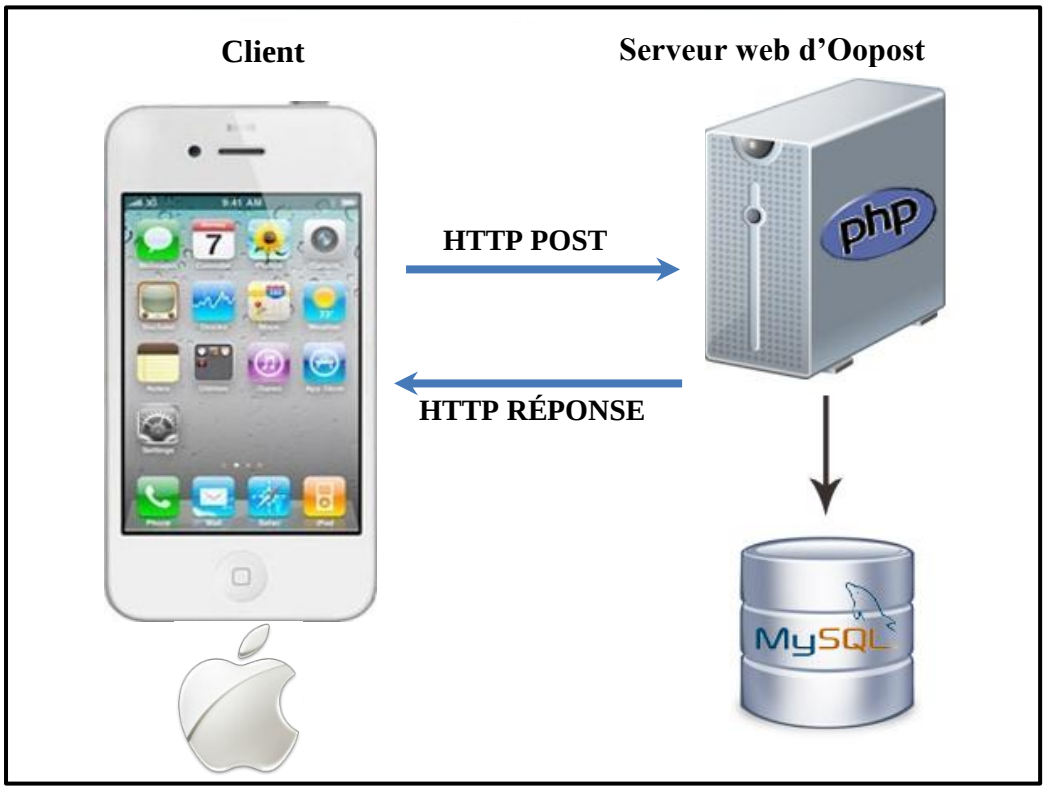


Figure 15: Architecture physique d'application

Trois composants principaux sont illustrés dans l'image ci-dessus :

- iPhone: le machine contient l'application et il communique avec le serveur web d'Oopost (réception et transmission de données sous forme de requête HTTP).
- Serveur Web Apache : le serveur web reçoit la requête et exécute le processus de PHP correspondant et il consulte la base de données s'il est nécessaire.
- Base de Données MySQL : la base de données stocke toutes les informations et elle va les chercher et les envoyer au serveur web quand il lui demande.

2.2. Architecture logique du système

Le graphe ci-dessous vous présente l'architecture logique de notre application.

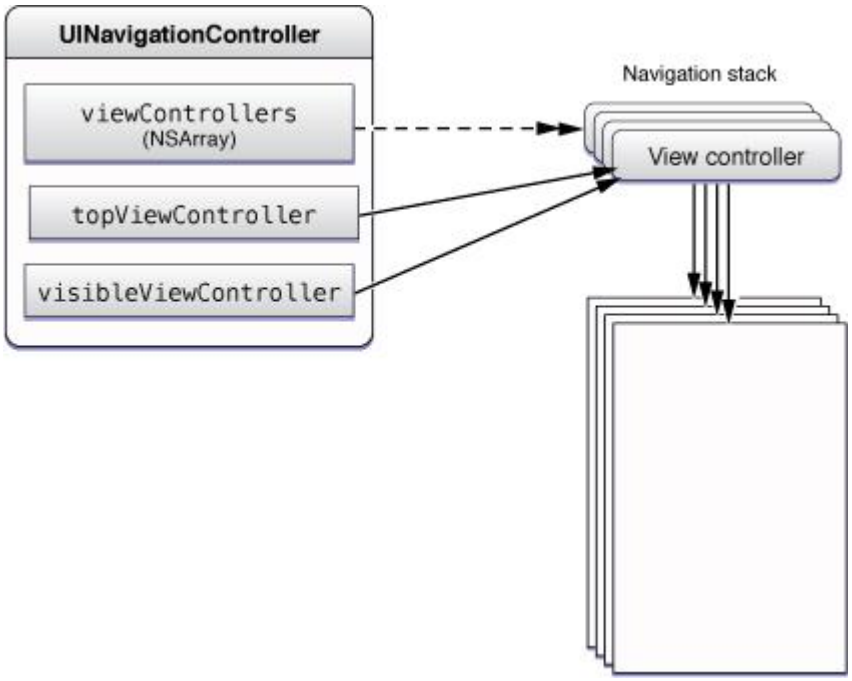


Figure 16: Architecture logique d'application

Comme l'utilisateur lance l'application, l'AppDelegate, c'est où l'application commence à exécuter, va initialiser un ViewController de base pour notre application. Si on change vers l'autre page, il va créer un nouveau ViewController et la mise sur la vue précédente, c'est comme le pile dans la figure au-dessus. Et si on navigue vers la page précédente en utilisant le bouton retour, il va dépiler le plus haut ViewController. Tous les ViewControllers peuvent utiliser la classe AppDelegate comme la source des données et ils peuvent être accédés partout dans notre application.

2.3. Organisation de l'IHM

On présente dans cette partie le diagramme de l'enchaînement de fenêtre de l'application. Ce diagramme indique la relation entre les contrôleurs des vues de notre système et comment utilisateur peut aller d'une page vers autre page. Comme on a seulement un type d'utilisateur qui utilise l'équipement tournant sur iOS, donc on a seulement un type diagramme.

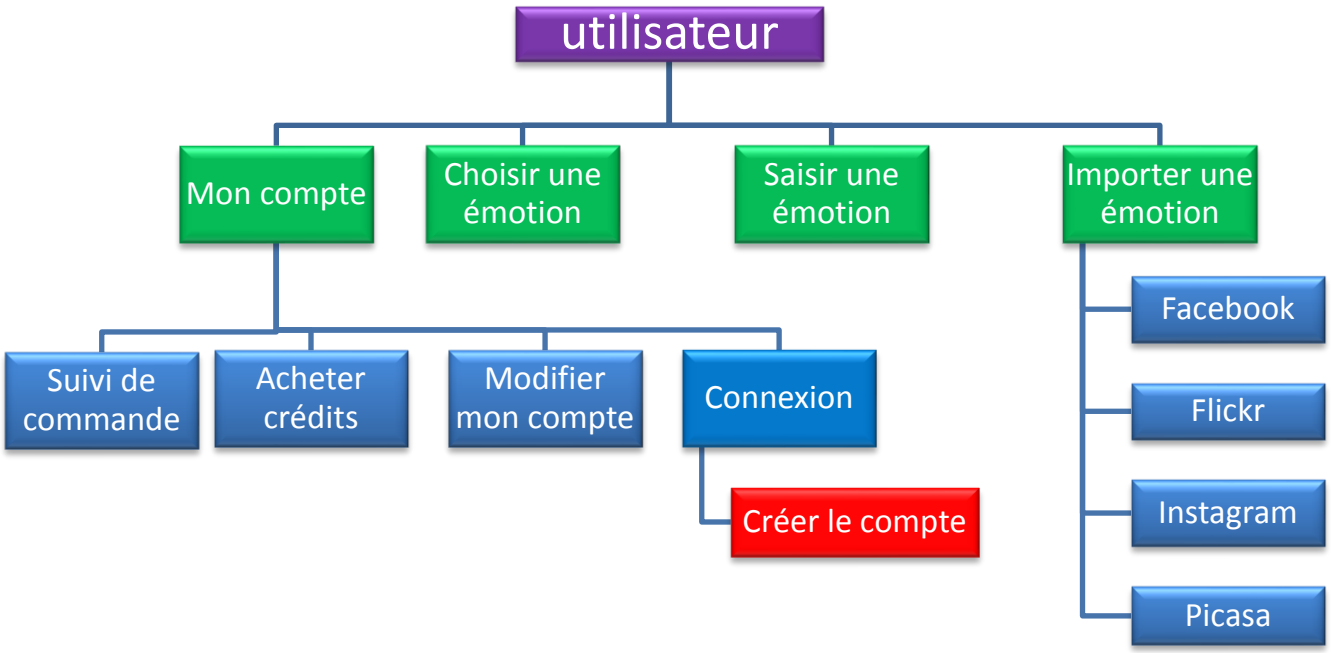


Figure 17: Diagramme de l'enchaînement de fenêtre du menu principale

Après on a une émotion qui obtient par Choisir une émotion, Saisir une émotion ou Importer une émotion, on arrive au même page de recto avec une image choisie.

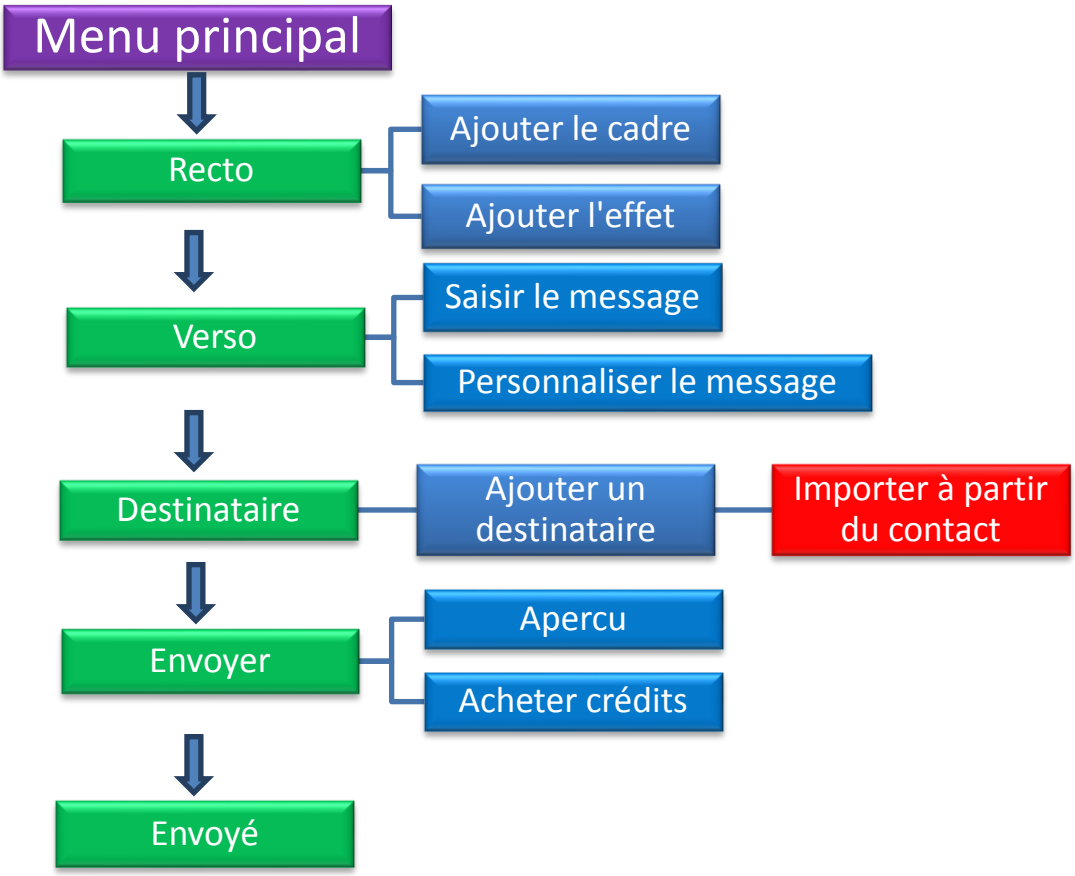


Figure 18: Diagramme de l'enchaînement de fenêtre après avoir une émotion

IV. IMPLEMENTATION

Cette partie consacre de l'implémentation du projet. Par conséquence, on vadiscuter en détaille comment on réalise ce projet. On parle également de quelques solutions que l'on a proposées pourrésoudre les problèmes qui sont étudiés au cours de développement.

1. L'importation photo à partir des réseaux sociaux

Dans notre application, cette fonctionnalité est la plus intéressent et la plus importante que le propriétaire de produit me demande de faire pour différencier et améliorer la dernière version. Il y a 4 réseaux sociaux disponibles pour permettre l'utilisateur de récupérerla photo: Facebook, Flickr, Instagram et Picasa.

L'API de Chute est un API que j'ai utilisé pour récupérer les photos à partir de ces réseaux sociaux, c'est-à-dire, on n'utilise que ce API pour connecter aux réseaux sociaux.Pour utiliser cet API, on doit suivre les étapes suivantes :

[1]Télécharger le composantPhotoPickerPlus et le SDK Chute dehttps://github.com/chute/photo-picker-plus/tree/master/iOS/PhotoPickerPlus
[2] Créer un compte développeur de Chute et créer une nouvelle application dans Chute à http://apps.getchute.com/

Dans la création du nouveaucompte dans Chute:

◦ Pour l'URL, vous pouvez entrer http://getchute.com/ si vous n'avez pas un site pour votre application.

◦ Pour l'URL de Callback, vous pouvez utiliser http://getchute.com/oauth/callbacksi vous n'avez pas besoinde Callbackpour autres raisons.

Figure 19: La création de compte pour utiliser API de chute

[3]Ajouterle SDK et le composant PhotoPickerPlus et lier lesbibliothèques requises comme suivant :

- AssetsLibrary
- CFNetwork
- EventKit
- libz.dylib
- MessageUI
- MobileCoreServices
- Security
- SystemConfiguration

[4]Vous aurez besoin de modifier votre App ID et AppSecretdans le fichier GCConstants.h qui peut être trouvé à la racine du répertoire Chute SDK. L'App ID et l'AppSecret situent dans l'onglet de Summary de votre panneau d'administration. Vous aurez aussi besoin d'ajuster l'URL de redirection pour correspondre au Callback URL à partir du panneau d'administration.

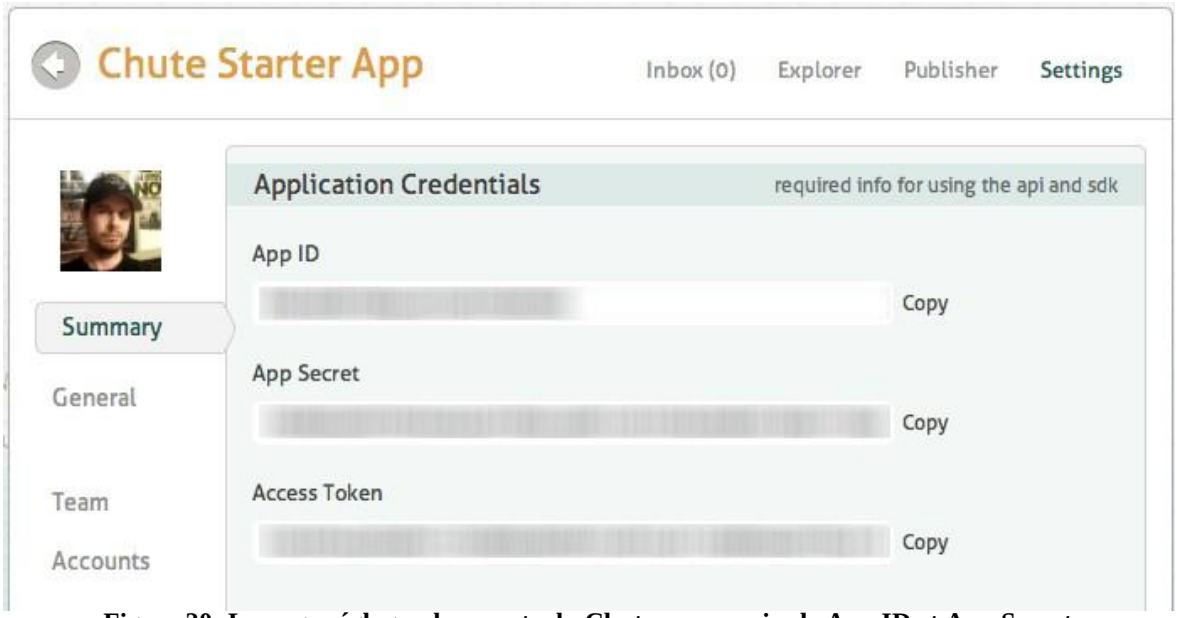


Figure 20: La page réglages de compte du Chute pour copier le App ID et App Secret

Vous devez copiez les informations au dessus comme l'App ID, App secret et mettre dans le fichier GCConstants.h suivant :

```
//replace the following setting with your own client info
#define kOAuthCallbackURL               @"http://getchute.com/oauth/callback"
#define kOAuthCallbackRelativeURL       @"/oauth/callback"
#define kOAuthAppID                     @"XXXXXXXXXXXXXXXXXXXXXXXX"
#define kOAuthAppSecret                 @"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

Àce stade, vous pouvez peut-êtreessayer de lancervotre projetpour s'assurerque toutestajouté bien.Vous obtiendrezquelques avertissements, mais s'il n'y a pasd'erreurs,tout estcorrectementajoutée etliée.

[5]Dans votre fichierviewController.h,vous devez importer PhotoPickerPlus.h et mettre en placePhotoPickerPlusDelegate comme le Delegate de la classe. Ensuite, vous devez créer un pointeur au ImageView et une méthode pour choisir les ressources. Cela devrait ressembler à ceci.

viewController.h

```
# import <UIKit/UIKit.h>
# import "PhotoPickerPlus.h"
@ interface ViewController: UIViewController<PhotoPickerPlusDelegate>
@ property (nonatomic, en lecture seule) IBOutletUIImageView * imageView;
- (IBAction) pickPhotoSelected: (id) sender;
@ end
```

Dans le fichierviewController.m,vous devez maintenantfaire le Synthesizede votre objetd'imageViewet écrirela méthodepour afficherle sélecteur de laphoto.La méthodepermet d'initialiserle contrôleur etdéfinirlui-mêmecomme leDelegate pour le présenteret le relâcher. Nous voulons aussile vueactuelvisible au derrièrele premier écrandu sélecteur, donc nous allons ajouter une lignepour cela aussi. Le code decela est :

viewController.m

```
@synthesizeimageView;
-(IBAction)pickPhotoSelected:(id)sender{
PhotoPickerPlus *temp = [[PhotoPickerPlusalloc] init];
[tempsetDelegate:self];
[selfsetModalPresentationStyle:UIModalPresentationCurrentContext];
[selfpresentViewController:tempanimated:YES completion:^(void){
[temp release];
}];
}
```

[6]Les méthodes de PhotoPickerPlusDelegatequ'on doit implémenter pour pouvoir récupérer l'image sontPhotoPickerPlusControllerDidCancel : et un autre méthode qui est aussi important estPhotoPickerPlusController:didFinishPickingMediaWithInfo:.Ils marchent exactement le même que les méthodes déléguées d'Apple UIImagePickerController. Vous pouvez vous référer à la documentation d'Apple sur UIImagePickerControllerDelegatepour voir les clés du dictionnaire. Celui que nous allons être concernés estUIImagePickerControllerOriginalImage. Donc, notre méthode d'annuler est juste de disparaître le sélecteur et notre méthode de réussite va afficher l'image dans le imageViewet disparaîtrele sélecteur. Le code de ces méthodes est :

viewController.m

```
-(void) PhotoPickerPlusControllerDidCancel:(PhotoPickerPlus *)picker{
        [selfdismissViewControllerAnimated:YES completion:^(void){
}];}
-(void) PhotoPickerPlusController:(PhotoPickerPlus *)picker
didFinishPickingMediaWithInfo:(NSDictionary *)info{
        [selfdismissViewControllerAnimated:YES completion:^(void){
```

```
        [[self imageView] setImage:[info
        objectForKey:UIImagePickerControllerOriginalImage]];
}];
}
```

[7] Finalement, vous devez avoir une application entièrement fonctionnelle maintenant qui vous permet de choisir une image à partir d'une variété des sources en ligne. En raison de l'accès à ALAssets, il présentera à l'utilisateur une boîte de dialogue demandant s'ils veulent permettre aux services de localisationparcequ'ALAssetsdispose de données de localisation associées aux images. Si l'utilisateur refuse, puis le sélecteur ne lui permet pas de choisir des images qui se trouvent sur l'appareil mais toutes les autres sources devraient toujours fonctionner.

2. La manipulation d'image en utilisant le geste

Pour permettre utilisateur de personnaliser la photo choisie, sur laquelle on peut faire quelques opérations comme redimensionner, pivoter et déplacer. Ces opérations sont effectuées par le geste parce que l'iPhone est un Smartphone ayantl'écran tactile.

UIGestureRecognizer est une classe (ou, tout simplement, un système de reconnaissance gestuelle) qui découple la logique de reconnaissance d'un geste et d'agir sur cette reconnaissance. Lorsque l'un de ces objets reconnaît un geste commun ou, dans certains cas, un changement dans le geste, il envoie un message d'action à chaque objet cible désignée.

Les sous-classes concrètes d'UIGestureRecognizer sont les suivantes:

- UITapGestureRecognizer
- UIPinchGestureRecognizer
- UIRotationGestureRecognizer
- UISwipeGestureRecognizer
- UIPanGestureRecognizer
- UILongPressGestureRecognizer

Dans notre projet, la manipulation d'image nous permet de faire 3 opérations : redimensionner, déplacer et pivoter. En faisant ces fonctionnalités, on doit les attacher avec le geste en détectant les touches. Pour redimensionner on utilise la classe

UIPanGestureRecognizer, pour déplacer on utilise la classe UIPinchGestureRecognizer et pour pivoter on utilise la classe UIRotationGestureRecognizer.

L'utilisation UIGestureRecognizers est extrêmement simple. Vous venez de réaliser les étapes suivantes:

- Créer unereconnaissance de geste. Lorsque vous la créez, vous spécifiez une méthode de Callbackpour que la reconnaissance de geste puisse vous envoyer des réactualisations lorsque le geste démarre, modifie ou finit.

- Ajouter la reconnaissance de geste à un vue. Chaque geste de reconnaissance est associée à un (et un seul) vue. Lorsque la touchesetrouvedans les limites de ce vue, la reconnaissance de geste va voir si elle correspond au type de la touche qu'elleest en train de chercher, et si une correspondance est trouvée, elle en informera à la méthode de Callback.

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    self.imageView.userInteractionEnabled = YES;

    UIPanGestureRecognizer *panRecognizer = [[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(panDetected:)];
    [self.imageView addGestureRecognizer:panRecognizer];

    UIPinchGestureRecognizer *pinchRecognizer = [[UIPinchGestureRecognizer
alloc] initWithTarget:self action:@selector(pinchDetected:)];
    [self.imageView addGestureRecognizer:pinchRecognizer];

    UIRotationGestureRecognizer *rotationRecognizer =
[[UIRotationGestureRecognizer alloc] initWithTarget:self
action:@selector(rotationDetected:)];
    [self.imageView addGestureRecognizer:rotationRecognizer];
```

- Créer la méthode de Callback pour traiter chaque évènement

```
- (void)panDetected:(UIPanGestureRecognizer *)panRecognizer
{
    CGPoint translation = [panRecognizer translationInView:self.view];
    CGPoint imageViewPosition = self.imageView.center;
    imageViewPosition.x += translation.x;
    imageViewPosition.y += translation.y;

    self.imageView.center = imageViewPosition;
    [panRecognizer setTranslation:CGPointZero inView:self.view];
}

- (void)pinchDetected:(UIPinchGestureRecognizer *)pinchRecognizer
{
    CGFloat scale = pinchRecognizer.scale;
    self.imageView.transform =
CGAffineTransformScale(self.imageView.transform, scale, scale);
    pinchRecognizer.scale = 1.0;
}
```

```
- (void)rotationDetected:(UIRotationGestureRecognizer *)rotationRecognizer
{
    CGFloat angle = rotationRecognizer.rotation;
    self.imageView.transform =
CGAffineTransformRotate(self.imageView.transform, angle);
    rotationRecognizer.rotation = 0.0;
}
```

En plus, si vous voulez manipuler les 2 évènements (redimensionner et pivoter)simultanément, vous devez ajouter une autre méthode de Callback comme le suivant :

```
- (BOOL)gestureRecognizer:(UIGestureRecognizer *)gestureRecognizer
shouldRecognizeSimultaneouslyWithGestureRecognizer:(UIGestureRecognizer
*)otherGestureRecognizer
{
    return YES;
}
```

3. L'importation d'Address Book

Dans la phase d'ajouter les destinataires, on permet utilisateur de remplir la forme qui contient le nom, l'adresse, le code postale, la ville et le pays manuellement ou bien récupérer à partir du Address Book existantdans le contact. Pour réaliser le dernier, on doit suivre les étapes suivantes :

[1] Ajouter le Framework deAddressBook au notre projet.
[2] Importer 2 fichiers qui appartiennent au Framework de AddressBook à notre ViewController qui doit implémenter les codes pour permet l'utilisateur d'importer du Address Book. En plus, on doit aussi ajouter le protocole ABPeoplePickerNavigationControllerDelegate à notre ViewController pour implémenter quelques prototypes existants.

#import <AddressBook/AddressBook.h>
#import <AddressBookUI/AddressBookUI.h>
@interface ViewController: UIViewController<ABPeoplePickerNavigationController Delegate>

[3] Dans le fichier d'implémentation du ViewController, on crée une méthode de PeoplePicker, met le ViewController comme son delegate et affiche le Picker comme un Modal View Controller. Après on connecte ce méthode à un bouton pour qu'on puisse appeler la méthode.

```
- (IBAction)showPicker:(id)sender{
        ABPeoplePickerNavigationController *picker =
        [[ABPeoplePickerNavigationControlleralloc] init];
        picker.peoplePickerDelegate = self;
        [selfpresentModalViewController:pickeranimated:YES];
        }
```

[4]Le People Picker appels la méthode delegate quand l'utilisateur clique sur le bouton qu'on a créé au-dessus. Si l'utilisateur annule, la première méthode est appelée pour disparaitre le People Picker. S'il sélectionne une personne, la seconde méthode est appelé pour copier le nom, prénom, adresse, et les autres informations avant de disparaître le People Picker. Le People Picker appelle la troisième méthode quand l'utilisateur tape sur la propriété de la personne sélectionné. Mais dans notre application, le People Picker est toujours disparaître quand l'utilisateur sélectionne une personne, donc, cette méthode n'est pas appelée. En revanche, pour compléter l'implémentation du protocole, on doit la remplir avec cette dernière méthode.

```
- (void)peoplePickerNavigationControllerDidCancel:
 (ABPeoplePickerNavigationController *)peoplePicker{
        [selfdismissModalViewControllerAnimated:YES];
}

-
(BOOL)peoplePickerNavigationController:(ABPeoplePickerNavigationController*)people
PickershouldContinueAfterSelectingPerson:(ABRecordRef)person {
        [selfdisplayPerson:person];
        [selfdismissModalViewControllerAnimated:YES];
return NO;
}

-
(BOOL)peoplePickerNavigationController:(ABPeoplePickerNavigationController*)people
PickershouldContinueAfterSelectingPerson:(ABRecordRef)personproperty:(ABPropertyI
D)propertyidentifier:(ABMultiValueIdentifier)identifier
{return NO;}
```

[5] On doit obligatoirement implémenter la méthode ci-dessous pour qu'on puisse récupérer les informations et les traiter.

```
- (BOOL)peoplePickerNavigationController: (ABPeoplePickerNavigationController *)
    peoplePicker
      shouldContinueAfterSelectingPerson:(ABRecordRef)person {


    NSString *str =  (__bridge_transfer NSString *)ABRecordCopyValue(person,
        kABPersonFirstNameProperty);
    NSString *st1 =  (__bridge_transfer NSString *)ABRecordCopyValue(person,
        kABPersonLastNameProperty);
    if (st1 == nil) st1 = @"";
    else str = [str stringByAppendingString:@" "];
    str = [str stringByAppendingString:st1];

    ABMultiValueRef addresses = ABRecordCopyValue(person, kABPersonAddressProperty
        );
    int count = ABMultiValueGetCount(addresses);
    if (count == 0) {
        self.name.text = str;
        self.addressLine1.text = @"";
        self.addressLine2.text = @"";
        self.postalCode.text = @"";
        self.city.text = @"";
        self.country.text = @"";
        [self dismissModalViewControllerAnimated:YES];
        return NO;
    }
    CFDictionaryRef dict = ABMultiValueCopyValueAtIndex(addresses, 0);
    NSString *countryTmp = (__bridge_transfer NSString *)CFDictionaryGetValue(dict
        , kABPersonAddressCountryKey);

    NSDictionary *address = (__bridge_transfer NSDictionary*)
        ABMultiValueCopyValueAtIndex(addresses, 0);
    NSString *address1 = [address objectForKey:@"Street"];
    addressLine2.text = @"";
    if (count == 2) {
        NSDictionary *tmp = (__bridge_transfer NSDictionary*)
            ABMultiValueCopyValueAtIndex(addresses, 1);
        self.addressLine2.text = [tmp objectForKey:@"Street"];
        NSLog(@"address line 2 %@",self.addressLine2.text);
    }
    NSString *zipCode = [address objectForKey:@"ZIP"];
    NSString *town = [ address objectForKey:@"City" ];

    if (str == nil || [str isEqualToString:@""]) self.name.text = @"";
    else self.name.text = str;

    if (address1 == nil || [address1 isEqualToString:@""]) self.addressLine1.text
        = @"";
    else self.addressLine1.text = [address objectForKey:@"Street"];
    if (zipCode == nil || [zipCode isEqualToString:@""]) self.postalCode.text = @
        "";
    else self.postalCode.text = [address objectForKey:@"ZIP" ];

    if (town == nil || [town isEqualToString:@""]) self.city.text = @"";
    else self.city.text = [ address objectForKey:@"City" ];

    if (countryTmp == nil || [countryTmp isEqualToString:@""]) self.country.text =
        @"";
    else self.country.text = countryTmp;

    [self.name becomeFirstResponder];
    [self dismissModalViewControllerAnimated:YES];
    return NO;
}
```

4. L'internationalisation

D’après le besoin non-fonctionnelle, l’application multi-langues est aussi une partie important dans notre application. On permet l’application de savoir 2 langues principales : français et anglais, mais l’utilisateur n’a pas le droit de régler dans l’application directement, l’application va changer automatiquement selon la langue d’iPhone. Pour faire ça, dans la production du code on doit bien préparer et suivre quelques étapes suivantes :

[1] Tous les chaînes des caractères dans notre application qui doivent être traduites, on doit faire la localisation en utilisant la fonction NSLocalizedString.

```
NSLocalizedString (@"Click Here!", @"Label for button");
```

Cette fonctionprend 2 paramètres : le texte par défaut et le commentaire pour décrire comment le texte est utilisé.

[2] On doit exécuter un script dans le Terminal pour tirer tous les NSLocalizedStrings à partir de vos fichiers de classe, et les mettre dans un fichier texte qui peut être traduit et remplacé. Pour faire ça vous devez :

- Ouvrir le Terminal et changer le répertoire où votre projet situe
- Créer un répertoire en.lproj
- Générer le fichier des chaînes de caractère par une ligne de commande

  genstrings –o en.lproj Classes/*.m
- Importer le répertoire en.lproj à votre projet d’Xcode

Cela va créer un fichier texte appelé Localizable.strings dans le dossier en.lproj, qui contient toutes les chaînes avec des commentaires :

```
/* Label for button */
"Click Here!" = "Click Here!";
```

[3] Pour le localiser aux autres langues, vous aurez besoin d’ajouter les langues dans le tab d’Info de notre projet.

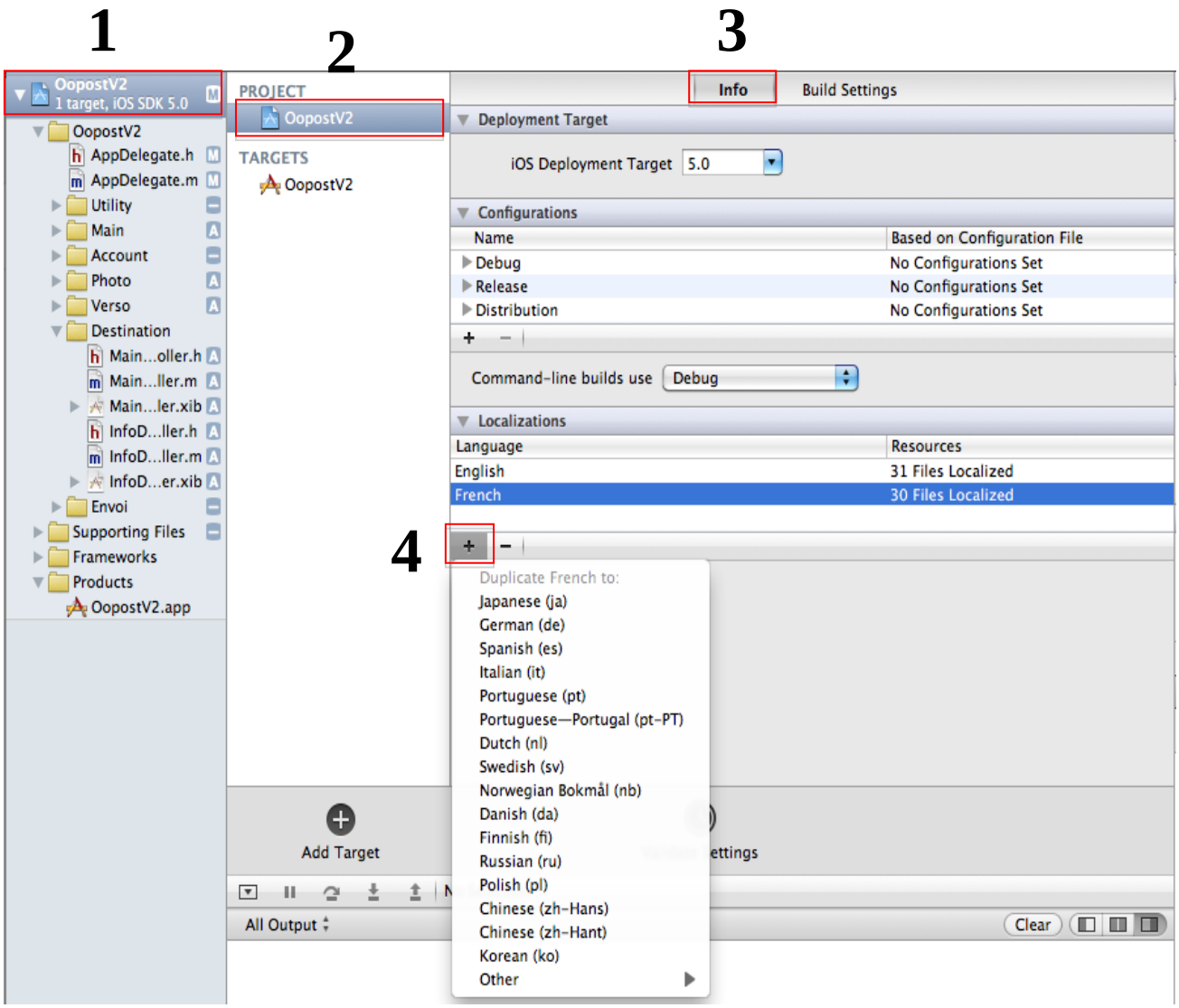


Figure 21: La page d'ajouter des langues localisés

[4] Après avoir ajouté les langues, l'Xcode va créer les fichiers des langues dans le répertoire en.lproj. Donc vous pouvez alors remplacer le texte dans les fichiers avec des chaînes localisées. Par exemple:

/ * Label for button * /

"Click Here!" = "Cliquez ici!";

5. Le service de Push Notification

L'Apple Push Notification Service (APNS) est le service qui est utilisé pour notifier les informationsà l'application iOS qui dispose une connexion d'internet. La notification peut être l'alerte detexte avec ou sans son et nombrede bagages mise à jour sur l'icône d'application. Dans notre application, il est utilisé pour dire à l'utilisateur que sa carte postale est bien envoyée à destinataire ou aux destinataires.

Pour permettre notre application de pouvoir utiliser ce service, les étapes au-dessous doivent être bien configurées :

[1] Générer un CertificateSigningRequest de Keychain de votre Mac et le sauvegarder dans le disc.

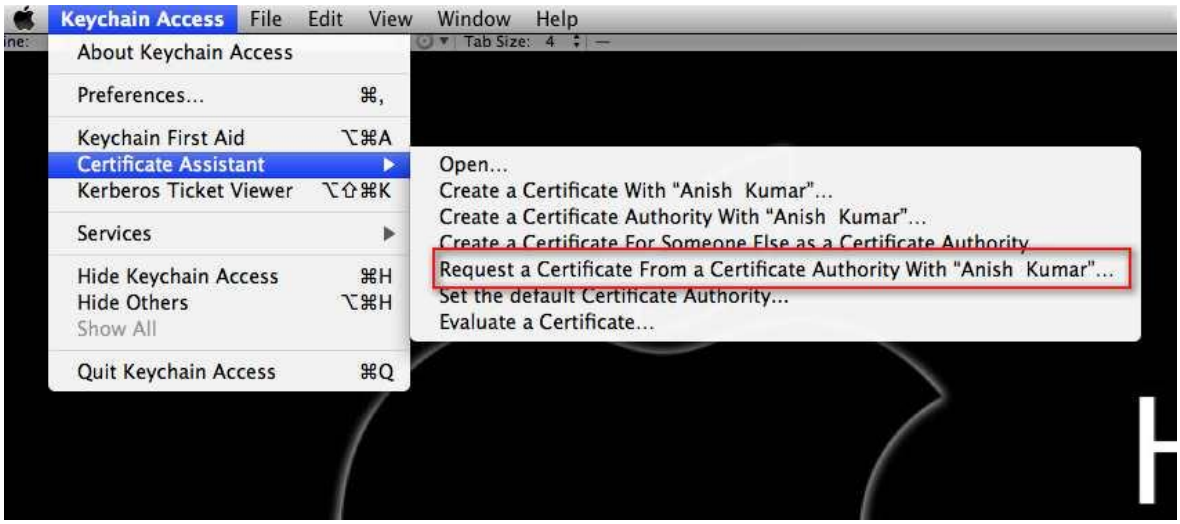


Figure 22: Génération du CertificateSigningRequest

[2] Vous devez connecter à votre compte du développement,cliquer suriPhone Developer Program Portal et cliquer sur App ID sur la gauche. Après, vous devez créer un nouvel identifiant com.mydomain.applicationName. Ce nom sera utilisé lors de la mise en place de votre application qui doit être signé avec un certificat de développement.

Figure 23: Le panier pour configurer « Development Push SSL Certificate ».

Après avoir soumis l'App ID, vous serez guidé à la page de la liste. Cliquez sur Configure pour régler. Cochez « Enable for Apple Push Notification Service »pour permettre APNS, puis cliquez sur Configurepres de «Development Push SSL Certificate».

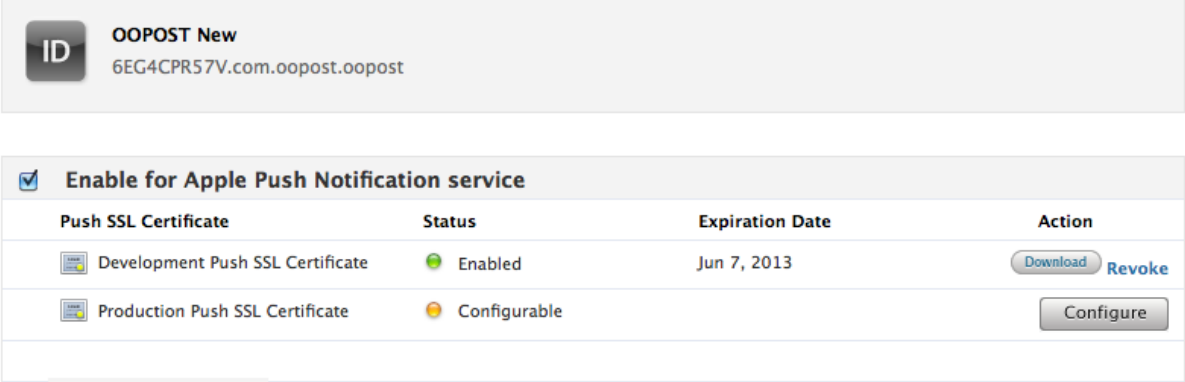


Figure 24: La page pour télécharger le Development Push SSL Certificate

Vousêtre demandé de certificat généré dans la première étape et télécharger le certificat (aps_developer_identity.cer) à partir du Program Portal en cliquant sur le bouton Download. Ensuite, vous devez double-cliquer sur ce certificat pour l'enregistrer dans votre Keychain et exporter cette clé en cliquant sur ce nouvel certificat. La clé exportée est enregistré avec le nom Certificate.p12 sur votre système.

De plus, vous devez exécuter 3 lignes des commandes pour générer les clés. La dernière ligne de commande est pour combiner les 2 clés à une clé globale qui a besoin pour les certificats différents.

```
openssl pkcs12 -clcerts -nokeys -out cert.pem -in Certificate.p12
openssl pkcs12 -nocerts -out key.pem -in Certificate.p12
catcert.pemkey.unencrypted.pem>ck.pem
```

[3] Après ça, vous devez cliquer sur Provisioning dans la barre de gauche, et créer un nouveau Provisioning Profile et utiliser le nouvel App ID et sélectionner le périphérique que vous souhaitez utiliser pour le développement. Aussi, vous devez télécharger le nouveau Provisioning Profile et double clique pour l'enregistrer au KeyChain. Après avoir le nouveau Provisioning Profile, on doit changer le Provisioning Profile dans votre Xcode au nouveau Provisioning Profile.

[4] En plus, vous devez implémenter les 3 méthodes ci-dessous. La première est pour dire à l'application de permet le service de Push Notification. La deuxième est pour savoir si

l'iPhone est déjà enregistré dans le serveur d'Apple. La troisième est appelé quand il y a le Push Notification à notre iPhone.

```
- (void)applicationDidFinishLaunching:(UIApplication *)app {
        [[UIApplicationsharedApplication] registerForRemoteNotificationTypes:
        (UIRemoteNotificationTypeAlert | UIRemoteNotificationTypeBadge
        |UIRemoteNotificationTypeSound)];
        …….
}

- (void)application:(UIApplication *)app
didRegisterForRemoteNotificationsWithDeviceToken:(NSData *)deviceToken {
        NSLog(@"device Token=%@",deviceToken);
}

- (void)application:(UIApplication *)application
didReceiveRemoteNotification:(NSDictionary *)userInfo {
if ( application.applicationState == UIApplicationStateActive ){
        UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Oopost V2"
                message:@"Veuillez consulter votre mail pour les nouveautés!"
                delegate:self
                cancelButtonTitle:@"OK"
                otherButtonTitles:nil, nil];
      [alert show];
}
```

[4] Finalement, on doit savoir comment on peut faire le Push Notification depuis le serveur en utilisant le code PHP qui a mis dans le serveur Oopost ci-dessous :

```php
<?php
// Put your device token here (without spaces):
$deviceToken = '8536e3bc0a79f6947619ee537afd76d57b158839961b08a7eb4b932a6e62d6d3';

// Put your private key's passphrase here:
$passphrase = '1234';

// Put your alert message here:
$message = 'Veuillez consulter votre mail pour les nouveaut&eacute;s!';

////////////////////////////////////////////////////////////////////////////////
$ctx = stream_context_create();
stream_context_set_option($ctx, 'ssl', 'local_cert', 'ck.pem');
stream_context_set_option($ctx, 'ssl', 'passphrase', $passphrase);

// Open a connection to the APNS server
$fp = stream_socket_client(
    'ssl://gateway.sandbox.push.apple.com:2195', $err,
    $errstr, 60, STREAM_CLIENT_CONNECT|STREAM_CLIENT_PERSISTENT, $ctx);

if (!$fp)
    exit("Failed to connect: $err $errstr" . PHP_EOL);

echo 'Connected to APNS' . PHP_EOL;

// Create the payload body
$body['aps'] = array(
    'alert' => $message,
    'sound' => 'default'
    );

// Encode the payload as JSON
$payload = json_encode($body);

// Build the binary notification
$msg = chr(0) . pack('n', 32) . pack('H*', $deviceToken) . pack('n', strlen($payload)) . $payload;

// Send it to the server
$result = fwrite($fp, $msg, strlen($msg));

if (!$result)
    echo 'Message not delivered' . PHP_EOL;
else
    echo 'Message successfully delivered' . PHP_EOL;

// Close the connection to the server
fclose($fp);
```

6. La sauvegardede la carte

La sauvegarde de la carte est une fonctionnalité pour faciliter l'utilisateur dans la création de la carte parce qu'en tout moment, l'application sauvegarde tout ce qu'il a fait. Donc il peut éteindre l'application et continuer de créer la carte avec toutes les informations qu'il a saisir comme la photo, le message, les informations des destinataires. Les informations sauvegardés reste dans l'iPhone jusqu'à l'utilisateur supprime la carte ou l'application.

Pour permettre notre application de faire ça, on a utilisé la classeNSUserDefaults. Elle est une façon standard que les développeurs d'iOS utilisent pour stocker les informations de leur applications et plupart les réglages.

Lorsque vous créez une application pour iOS, le système créela base de données avec le Key-Value sur votre application.On peut accéder à cette base de données d'ajouter, supprimer et mettre à jour.

Pour récupérer cette base de données, on peut faire comme ci-dessous

```
NSUserDefaults*userDefaults=[NSUserDefaultsstandardUserDefaults];
```

NSUserDefaultsvous permet de stocker les objets qui sont dans la classeNSData, NSString, NSNumber, NSDate, NSArray, NSDictionary avec des méthodessuivantes.

– setBool:forKey:

– setFloat:forKey:

– setInteger:forKey:

– setObject:forKey:

– setDouble:forKey:

EX: [userDefaultssetObject:@”NAPO” forKey:@”name”];

Pour obtenirles valeurs enregistrées, vous utilisez des méthodessuivantes.

– boolForKey:

– floatForKey:

– integerForKey:

– objectForKey:

– doubleForKey:

EX :NSString* name = [userDefaultsobjectForKey:@”name”] ;

Pour supprimern'importe quel objetdans labase de données, vous utilisezla fonction suivante.

– removeObjectForKey:

Pour sauvegarder l’image, on doit le convertir

```
UIImage *img= [info objectForKey: @"UIImagePickerControllerOriginalImage"];
NSData *imageData = UIImageJPEGRepresentation(img, 100);
[[NSUserDefaultsstandardUserDefaults]setObject:imageDataforKey:@"image"];
```

Pour recupérer l’image, on fait comme le suivant.

```
NSData *imgData = (NSData*)[[NSUserDefaultsstandardUserDefaults]
        objectForKey:@"image"];
UIImage *img = [UIImageimageWithData: imgData];
```

V. CONCLUSION

Le stage à CamMob s'est bien passé, on a utilisé toutes nos connaissances et nos expériences qu'on a obtenues de notre école, inclus les recherches sur les nouveaux technologies, les aides de nos enseignants et les indications de notre tuteur du stage pour réaliser le projet. On trouve que le projet est très intéressant qui me donne beaucoup d'expériences concernant notre domaine de comment à réaliser une application.

Finalement, pendant 4 mois du développement du système,même qu'on ne peut pas finir complètement l'application, mais on a réalisé l'application avec presque toutes les fonctionnalités nécessaires.Toutefois, l'application a encore des points qui sont obligés d'améliorer et de modifier pour être parfait. Avec les expériences du stage, on croit qu'il est très important pour notre vie professionnelle après nos études.

Cette partie est la dernière partie du mémoire. Elle présente d'abord le bilan. Ensuite, les difficultés rencontrées et les expériences acquis pendant le stage. Puis, il porte la partie de perspective après le stage.

1. Bilan du projet

1.1. Fonctionnalités réalisées

Les parties réalisées sont :

- Importer les émotions
    - Importer une photo de pellicule
    - Importer une photo d'appareil photo
    - Importer une photoà partir des réseaux sociaux (Facebook, Picasa, Flickr et Instagram)

- La création ducôté recto de la carte
    - Opération sur la photo comme redimensionner, pivoter et déplacer
    - Appliquer les effets (Normale, Noir et blanc, Sépia et Vintage)
    - Appliquer le cadre à la photo choisie

- La création du côtéverso de la carte
    - Changement de police de caractère parmi les 4 polices de caractère (Arial, Segoe Print, Bell MT, Traveling Typewriter)
    - Changement l'alignement (Gauche, Centre et Droit)
    - Changement des couleurs (Noir, Verte, Rouge, Bleu et Orange)
    - Changement de la taille du message dans l'intervalle de 10 à 30

- Gestion du compte d'utilisateur
    - Créer le compte d'utilisateur
    - Modifier le compte d'utilisateur
    - Suivi de commande
    - Connexion / Déconnexion
    - Acheter crédit

- Ajouter plusieurs destinataires manuellement ou à partir du contact et avoir la possibilité d'envoyer aux plusieurs destinataires
- Pouvoir envoyer la carte en utilisant le crédit existant ou le code promo offert par l'Oopost
- Rendre l'application multi-langue (principalement français et anglais)
- Pouvoir transmettre les nouveautés à l'utilisateur d'Oopost par alerter de Push Notification à partir du serveur Oopost
- Sauvegarder la carte qui est déjà personnalisée pour que l'utilisateur puisse continuer lorsqu'il éteint l'application ou bien réutilise la carte pour l'envoyer aux autres destinataires
- Pouvoir évaluer l'application Oopost sur l'App Store
- Envoyer la newsletter par e-mail à tous les clients d'Oopost

1.2. Fonctionnalités non réalisées

A partir des parties réalisées au-dessus, on reste encore un point non complété qui est le parrainage des amis par e-mail.

2. Points forts et points faibles de l'application

2.1. Points forts

L'application qu'on a créée a des points forts comme les suivants :

- l’application est facile à utiliser car son interface graphique est cohérente et agréabledepuis le début jusqu’à la fin
- l’application peut être utilisée dans 2 langues : français et anglais

2.2. Points faibles

L’application possède aussi les points faibles tels que :

- la performance est un peu lente

3. Expérience acquise

Les connaissances acquises pendant ce 4 mois de stage sont :

- La méthodologie de travail du développement
- La recherche de technologies, de techniques, et de documents pour résoudre le problème et pour développer l’application
- La vie professionnelle au marché du travail
- Le développement de l’application sur iPhone

4. Difficultés

En vérité, c’est normal que le développeur toujours rencontre les points difficiles pendant le processus de développement. Ça m’est égal, j’avais aussi la confrontation avec les difficultés pendant mon travail. Quelques problèmes sont mineur mais quelques sont grandes et je dois trouver les solutions appropriés pour les résoudre. Parmi tous, j’ai confronté deux problèmes qui sont les plus graves.

Le premier est que je n’ai jamais travaillé avec iOS. Bien que ce système opératoire ne soit pas très difficile à apprendre, il a pris de temps pour étudier toutes ses fonctionnalités et sa structure.

Le deuxième problèmeporte sur les besoins et la conception. Comme dans le phénomène de développement d’application pour les mobiles, il y a toujours les changements au long de processus d’élaboration d’application qu’ils peuvent être énorme ou petit. Donc on a toujours perdu une partie de temps pour adapter aux nouvelles modifications.

5. Perspective

A cause de la limitation de temps, on n'a mis que les objectifs principaux pour le développement de l'application. Ces objectifs ne sont pas suffisants pour perfectionner le projet. Par conséquence, on souhaite pour l'extensibilité de projet dans les parties suivantes :

- Perfectionner toutes les fonctionnalités
- Finirla fonctionnalité restante
- Améliorer la performance de l'application

REFERENCES BIBLIOGRAPHIQUES

- **Livres (**Ressources d'Apple pour iOS développeur)

[1] Address Book Programming Guide for iOS (2012)

[2] Animation Types and Timing Programming Guide (2010)

[3] Internationalization Programming Topics (2012)

[4] iOS App Programming Guide (2012)

[5] Local and Push Notification Programming Guide (2011)

[6] Object Oriented Programming with Objective-C (2010)

[7] Table View Programming Guide for iOS (2011)

[8] The Objective-C Programming Language (2011)

[9] View Controller Programming Guide for iOS (2012)

- **Internet**

[1] Resource d'Apple pour iOS développement: Guides et Exemples de code
http://developer.apple.com/devcenter/ios/index.action

[2] API pour connecter aux réseaux sociaux
http://picture.io

[3] Les exemples de code pour personnaliser les contrôles d'interface utilisateur pour iOS
http://cocoacontrols.com

[4] L'exemple de code pour évaluer l'application sur l'App Store
http://arashpayan.com

[5] La caractéristique de l'iPhone
http://fr.wikipedia.org/wiki/IPhone

[6] La plate-forme de l'iOS
http://fr.wikipedia.org/wiki/IOS_%28Apple%29

[7] L'Objective-C
http://fr.wikipedia.org/wiki/Objective-C

ANNEXE A : SCHEMA DE DONNEES

❖ **La base de données d'Oopost existante**

No	Attributs	Type de données
1	code_promo	varchar(4)
2	qty	int

Tableau 2: Le tableau code_promo_mobile

No	Attributs	Type de données
1	id	int
2	name	varchar(64)
3	name_fr	varchar(64)
4	countries_iso_code2	char(2)
5	countries_iso_code3	char(3)
6	address_format_id	int

Tableau 3: Le tableau country

No	Attributs	Type de données
1	distribution_channel	enum('facebook', 'oopost', 'mobile', 'dedicate')
2	campaign_id	int
3	url	varchar(100)
4	card_quota	int
5	start_date	date
6	end_date	date
7	sent_card	int

Tableau 4: Le tableau distributionchannel

No	Attributs	Type de données
1	design_id	int
2	design_file	varchar(50)
3	description	varchar(255)
4	photo_width	int
5	photo_height	int
6	collate_x	int
7	collate_y	int
8	effect	varchar(50)
9	creation_date	date
10	credit	varchar(50)
11	style	int
12	thumbnail_design_file	varchar(100)
13	thumbnail_width	int
14	thumbnail_height	int
15	thumbnail_x	int
16	thumbnail_y	int

Tableau 5: Le tableau freedesign

No	Attributs	Type de données
1	style_id	int
2	style_name	varchar(50)
3	description	varchar(255)

Tableau 6: Le tableau freestyle

No	Attributs	Type de données
1	commanded	int
2	login	varchar(50)

Tableau 7: Le tableau generateCommande

No	Attributs	Type de données
1	histoire_id	int
2	user_id	int
3	cardpack	varchar(10)
4	nombre	int
5	data_achat	date
6	montant	float
7	nb_credit	int
8	order_number	varchar(10)

Tableau 8: Le tableau histoire_commande_pack

No	Attributs	Type de données
1	id	int
2	userID	varchar(50)
3	photofile	varchar(50)
4	message	varchar(1000)
5	destinationName	varchar(50)
6	destinationAdr1	varchar(50)
7	destinationAdr2	varchar(50)
8	destinationZIP	varchar(50)
9	destinationCity	varchar(50)
10	destinationCountry	varchar(50)
11	creationDate	datetime
12	sendDate	datetime
13	sendStatus	varchar(50)

Tableau 9: Le tableau iphone_card

No	Attributs	Type de données
1	id	int
2	name	varchar(50)
3	name_fr	varchar(20)

Tableau 10: Le tableau list_font_align

No	Attributs	Type de données
1	id	int
2	name	varchar(50)

Tableau 11: Le tableau list_font_name

No	Attributs	Type de données
1	card_id	int
2	user_id	int
3	recipient_id	int
4	create_date	date
5	send_date	date
6	design_id	int
7	print_date	date
8	is_collection	tinyint
9	created_channal	varchar(50)
10	photo_file	varchar(100)
11	photo_quality	enum('bad','good')
12	file_source	varchar(100)
13	file_source_sample	varchar(100)
14	file_source_small	varchar(100)
15	ip_address	varchar(50)
16	paidby	enum('pack','bank','promotion')
17	price	float
18	order_number	varchar(10)
19	actif	tinyint
20	message	text
21	font	varchar(100)
22	size	varchar(10)
23	align	varchar(20)
24	status	char(1)

Tableau 12: Le tableau paidcard

No	Attributs	Type de données
1	order_key	varchar(10)
2	user_id	int
3	pack_description	varchar(50)
4	price	int
5	buy_date	date
6	expire_date	date

Tableau 13: Le tableau paidcardpack

No	Attributs	Type de données
1	codepromo_card	varchar(4)
2	card_id	int

Tableau 14: Le tableau promocode_card

No	Attributs	Type de données
1	id	int
2	initial	varchar(10)
3	name	varchar(50)
4	street_adr	varchar(50)
5	street_adr_2	varchar(100)
6	postal_code	varchar(50)
7	city	varchar(50)
8	stat	varchar(100)
9	country_id	int
10	sex	varchar(5)
11	job	varchar(50)
12	age	int
13	point_interest	varchar(300)
14	adr_validation	binary
15	user_id	int

Tableau 15: Le tableau recipient

No	Attributs	Type de données
1	user_id	int
2	email	varchar(50)
3	pwd	varchar(32)
4	initial	varchar(10)
5	firstname	varchar(50)
6	familyname	varchar(50)
7	dob	date
8	target_job_id	int
9	target_age_id	int
10	maritalstatus	varchar(50)
11	streetaddress	varchar(50)
12	zipcode	varchar(50)
13	city	varchar(50)
14	country	int
15	facebook	varchar(50)
16	picasa	varchar(50)
17	flickr	varchar(50)
18	telephone	varchar(50)
19	member_date	date
20	newsletter	tinyint
21	point_interest	varchar(500)
22	number_child_5yrs	int
23	number_child_15yrs	int
24	number_child_18yrs	int
25	remain_paid_credit	int
26	paid_credit_expire_date	date
27	remain_free_credit	int
28	admin	tinyint

Tableau 16: Le tableau user

No	Attributs	Type de données
1	email	varchar(50)
2	iphoneUniqueIdentifier	varchar(50)
3	name	varchar(50)
4	password	varchar(20)
5	iphoneModel	varchar(50)
6	iphoneLocalizedModel	varchar(50)
7	iphoneSystemName	varchar(50)
8	iphoneSystemVersion	varchar(50)
9	transfertToWeb	binary
10	lastlogin	date
11	remainCredit	int

Tableau 17: Le tableau useriphone

ANNEXE B : CAPTURE D'ECRAN

Figure 25: L'écran de démarrage

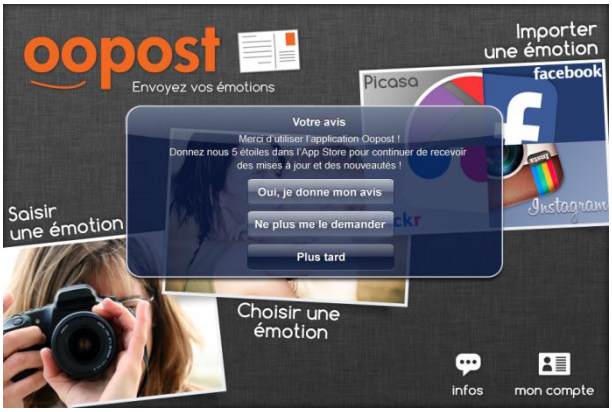


Figure 26: L'évaluation d'app sur l'App Store

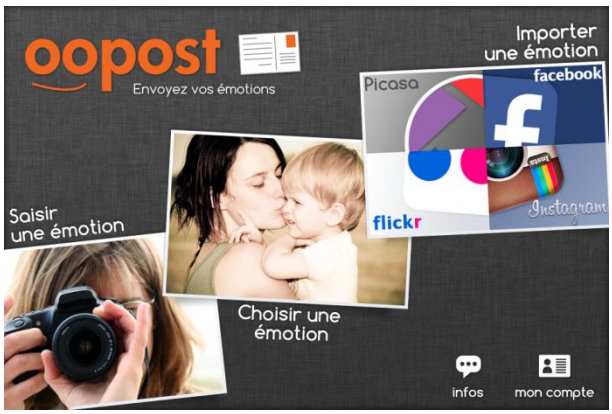


Figure 27: Menu principal

Figure 28: L'ajoute de la cadre

Figure 29: Ajouterl'effet

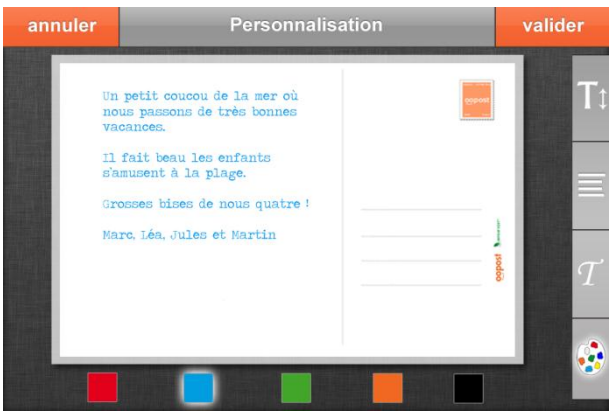


Figure 30: Personnaliser le message avec la taille, alignement, police de caractère et couleur

Figure 31: L'ajoute de destinataire

Figure 32: Connexion

Figure 33: Mon compte

Figure 34: Suivi de commande

Figure 35: L'achat des crédits

Figure 36: La création de compte