

Sébastien Corbin et François-Guillaume Ribreau  
CSII 2<sup>e</sup>année  
Le 21 Septembre 2011

---

# Dossier de spécifications

Génie Logiciel Embarqué

# Introduction

---

## 1.1 Matériel

### 1.1.1 Description

L'iPhone est une famille de smartphones conçue et commercialisée par Apple Inc. depuis 2007. Les modèles, dont l'interface utilisateur a été conçue avec le multi-touch, disposent d'un appareil photo, d'une fonction baladeur numérique, d'un client Internet (pour naviguer sur le Web ou consulter son courrier électronique), et de fonctions basiques telles que les SMS/MMS (messages texte et multimédia); mais disposent aussi de la messagerie vocale visuelle et de l'App Store, qui permet de télécharger des applications, allant des jeux aux réseaux sociaux, en passant par les GPS, la télévision, la presse électronique ou encore les bandes-dessinées. Au mois de mai 2010, on compte plus de 225 000 applications.

### 1.1.2 Spécifications

Les spécifications suivantes sont relatives à l'iPhone 4 (génération Juin 2010)

**Système d'exploitation** iOS 4.3.5 (build 8L1, sortie le 25 juillet 2011)

**Alimentation** Batterie lithium-polymère

**Processeur** Puce apple A4 1 GHz

**Stockage** 16/32 Go de mémoire flash

**Mémoire** 512 Mio de DRAM

**Écran** Écran multi-touch de 3,5 pouces

**Résolution** 960 x 640 px (326 ppp)

**Carte graphique** PowerVR SGX 535 GPU

**Caméra** *Arrière* : 5 mégapixels, vidéo 720p avec Flash LED / *Avant* : VGA

## 1.2 Le système d'exploitation 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.

## 1.3 L'environnement de développement

Le kit de développement, disponible uniquement pour Mac OS X, propose les outils nécessaires à la création d'une application pouvant tourner sous iOS. Si son téléchargement et son utilisation sont gratuits, la publication de telles applications requiert d'adhérer au programme des développeurs Apple, pour la somme de 99\$ par an. Il n'en demeure pas moins que cette offre peut s'avérer intéressante pour bon nombre de développeurs, étant donnée la taille du marché créé par iOS. En plus d'offrir aux développeurs exactement les mêmes API que celles d'Apple, le SDK contient de nombreux outils facilitant le développement et le test d'applications pour iOS.

### Outils de développement

La plupart des outils de développement du SDK étaient déjà présents dans Mac OS avant son arrivée. Cependant, ils gèrent désormais l'utilisation de l'iPhone, en tant que plate-forme de développement :

**Xcode** Environnement de Développement Intégré par défaut sur Mac OS X. Il permet l'écriture, la gestion et la compilation de projets de développement, écrits notamment en Objective-C. L'iPhone SDK y ajoute les bibliothèques de développement pour iOS. Il est donc possible pour le développeur de créer des projets d'applications pour ce système. Pour tester l'application, deux possibilités existent : le développeur peut brancher un iPhone ou iPod Touch à son ordinateur Mac, puis y lancer l'application comme test, ou lancer l'application en test dans iPhone Simulator.

**Interface Builder** permet de construire une interface pour Cocoa Touch manuellement, à l'aide de glisser-déposer. Il permet également de traduire facilement une application dans plusieurs 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 au préalable, à l'aide d'un glisser-déposer. Finalement, le fichier d'interface ainsi créé est ajouté au projet Xcode. 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 système dans iPhone Simulator.

**iPhone Simulator** 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 quelle version du firmware il désire utiliser.

# Architecture serveur

---

Entry point (point d'entrée) http ://IP :3000/api/

**/check/userId/ POST**

- Paramètres : lat, lon, userId

-

# Base de données

---

Collections en BDD :

**users** collection des utilisateurs

**checks** collection des checks

Ajout d'une première entitée dans Checks :

**loc** [1,1]

**userId** ObjectId("4e7e0614bd99e29380000000")

**data** new Date()

Ajout d'une première entitée dans Users

**login** FG

**password** "098f6bcd4621d373cade4e832627b4f6" ("test" en MD5)

**checks** [ObjectId("4e7e0614bd99e098900000000")]

Ajout d'un index sur les checks via le CLI \$mongo :

---

Listing 1.1 – Commandes à entrer dans le CLI mongo

---

```
> use geobbs  
> db.checks.ensureIndex({ loc : "2d" })
```

---

# Ce qu'il faut traiter

---

Intro Les os de l'embarqué ->(liste et ce qu'il font chacun)

Notion d'entrée/sortie

Les Env. de dev (avec particularité du remote debugging)

Résumé des différentes méthodes (agile, xp etc..., tdd etc...)

Projet -> Equipe -> Cahier des charges

Choix de la méthode : -> Méthode Agile -> + TDD

Languages : Objective-C / JavaScript

Libraries : ExpressJS (serveur http) Mongoose <http://mongoosejs.com/> (mongodb)

La sécurisation est faite par OAuth

# Matériel existant

---

iPhone

MacBookPro 13 et 15 pouces



# Analyse des librairies

---

Guide de programmation iOS

Géolocalisation :

Introduction au développement relatif à la géolocalisation

# Interface utilisateur

---

## Accueil

**Cas 1 :** GeoBbs est lié à Twitter, Foursquare, Facebook.

Affichage des derniers Tweets/Check-in (avec algorithme d'importance) de ses amis dans le secteur où se trouve l'utilisateur

**Cas 2 :** Aucun compte lié

Affichage des dernières notifications envoyées dans le secteur où se trouve l'utilisateur.  
L'UI de la home ressemble à Foursquare :

**Liste** Une liste des notifications

**Menu en bas** Liste | Notifier | Réalité augmentée