

Sébastien Corbin et François-Guillaume Ribreau
CSII 2^eannée
Le 21 Septembre 2011

Dossier de spécifications

Génie Logiciel Embarqué

But du document

Le but de ce document est de lister toutes les fonctionnalités du projet GeoBBS ainsi que de donner un aperçu des interfaces et des interactions la composant. Ce document fournit aussi une description des utilisateurs et leur rôle dans le logiciel.

Documents applicables et de références

2.0.1 Documents applicables

.

2.0.2 Documents de références

.

Diagramme de cas d'utilisation

TODO Diagramme global des cas d'utilisation ici

Description globale

4.0.3 Environnement

L'application est hébergée sur le serveur internet de la société GEOBBS CORP. Et est accessible depuis l'extérieur via l'application mobile.

4.0.4 Profils des utilisateurs

GeoBBS accepte deux types d'utilisateurs, les visiteurs et les membres.

Les visiteurs

Toute personne qui arrive sur le site (ou l'application) pour la première fois et qui ne possède pas de compte ou n'est pas connecté est appelée visiteur.

Les membres

Un membre est une personne qui s'est identifiée auprès du site (ou de l'application mobile). Elle est distinguée des autres membres par un identifiant unique et peut utiliser l'intégralité des services qu'offre GeoBBS.

Spécifications générales

5.0.5 Description des services attendus

Les services attendus sont répartis dans les différents modules de l'application. Ces modules sont accessibles ou non en fonction du type d'utilisateur, visiteur ou membre.

Les visiteurs ont uniquement accès au formulaire de création de compte et au formulaire de connexion. Aucun autre écran n'est accessible au grand public.

Les membres peuvent agir sur les 4 grands modules :

Near répertories les membres proches géographiquement de l'utilisateur actuel

Check permet au membre de "checker" c'est à dire informer proactivement GeoBBS de sa position en y ajoutant s'il le souhaite une description et une image (ou vidéo)

AR Affiche en réalité augmentée les points d'intérêts autour de l'utilisateur

Profil permet à l'utilisateur d'éditer son profil

5.0.6 Approche utilisée dans ce document

Plutôt que de disperser les spécifications (diagramme de cas d'utilisation, diagramme de classe, diagramme de séquence, maquettes) d'un même module à diverses endroits du document. La spécification par composant a été préférée. Elle permet une lecture plus aisée et rapide de l'intégralité du fonctionnement d'un composant d'un module.

5.0.7 Description générale des fonctions

Le visiteur s'inscrit

Un visiteur peut effectuer 2 actions, s'inscrire et se connecter.

Cas d'utilisation : le visiteur s'inscrit

But : Enregistrer les informations du visiteur en base

Résumé : /

Acteurs :

- Visiteur (acteur principal)

Description des enchaînements

Pré-conditions :

- Le site est accessible, la base de données est disponible, le visiteur est sur le site

Enchaînements nominaux :

- Évènement déclencheur : Lorsque le visiteur souhaite se créer un compte
TODO : faire tableau avec les différentes étapes | Visiteur | Logiciel |

Enchaînements alternatifs :

alternatif 1 : Le visiteur souhaite s'inscrire via son compte Facebook, Twitter. TODO :

Enchaînements d'erreurs :

Exception 1 : Les informations obligatoires ne sont pas toutes saisies

- Afficher un message d'erreur récapitulant les champs restant à saisir

Exception 2 : Certaines informations saisies sont incorrectes (Email, nom d'utilisateur, code postal)

Exception 3 : Le nom d'utilisateur est déjà présent en BDD

- Afficher un message d'erreur : « Ce nom d'utilisateur est déjà présent en BDD, veuillez en choisir un autre »

Post-conditions : /

6.1 Ecrans

TODO : Insérer les écrans ici

TODO : Liste des cas d'utilisation que l'on POURRAIT traiter (tu vas quand même pas tous les faire, ça prendrait trop de temps...) - Connexion du visiteur (on s'en branle) - Visualiser

et editer profil (on s'en branle) - Ajouter un checker (ok) - Visualiser les checks autour (ok) - Consulter le profil d'un amis (on s'en branle) - Ajouter une application - Lancer une application - - ...

Au pire, tu peux en faire quelque un (genre juste "ajouter check" et "visualiser checks", en fait juste ceux dont j'ai fait des mockups/capture d'écran, t'embête pas pour le reste) et peut-être citer les autres cas comme étant réalisable mais dans le futur.

Ecrans supplémentaires

Diagramme de classes

L'application serveur et l'application mobile étant des prototypes développés suivant le paradigme fonctionnel, ils n'utilisent aucune classe. Il n'y a donc pas de diagramme de classe dans ce document. Cependant les spécifications futures (post-prototype) inclueront classes et un diagramme de classe devra donc être présent.

Spécifications technologiques et opératoires

9.1 Matériel

9.1.1 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.

9.1.2 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

9.2 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.

9.3 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.

9.3.1 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

Port par défaut : 3000

L'adresse du serveur et le port sont défini directement dans le code de l'application

10.1 Points d'entrée

POST /check/create/ :

Description Ajoute un check

Paramètres :

userId (string) Id de l'utilisateur

lat (float) Latitude

lon (float) Longitude

description (string) (optionel) Description du check

imgUrl (string) (optionel) Url de l'image à lier

GET /check/list/ :

Description Retourne la liste des checks aux alentours de lat,lon.

Paramètres :

userId (string) Id de l'utilisateur

lat (float) Latitude

lon (float) Longitude

distance (entier) (optionel) Distance maximum par rapport à lat,lon.

count (entier) (optionel) Nombre maximum de données à retourner

GET /user/profil/ :

Description Affiche une page HTML d'édition du profil et des préférences de l'utilisateur.

Paramètres :

userId (string) Id de l'utilisateur

Base de données

L'administration de la base de donnée se fera par l'utilisateur via RockMongo ou MongoHub.

Collections en BDD :

users collection des utilisateurs

checks collection des checks

Ajout d'une première entitée dans Checks :

```
loc [1,1]
__userId ObjectId("4e7e0614bd99e29380000000")
date new Date()
description test
User login: FG
```

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 11.1 – Commandes à entrer dans le CLI mongo

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

Délimitation de l'application

TODO : expliquer que nous sommes parti sur un prototype et que nous ne réaliserons pas l'intégralité de la plateforme pour des raisons de temps.