
Manuel de développement

Boussole sur Smartphone pour l'évacuation de bâtiment en situation d'urgence.



Réalisé par :

BEL HAJ HASSINE Souha
DIONI Oumar
LAHLOU Yousra
OZKAYA Sila
TOUMI Sofiane



Encadré par :

Mr. RACHEDI Abderrezak

Table of Contents

I. Introduction	3
II. Serveur web	3
1. Outils utilisés	3
2. Les différentes classes	3
III. La base de données MySQL	4
IV. Application Android	4
1. Outils utilisés	4
2. Activités.....	4
3. Connexion au serveur	5

I. Introduction

Dans le cadre de notre projet de fin d'études, nous avons développés l'application BEST : Boussole pour l'évacuation en situation d'urgence.

Ce manuel de développement est destiné à exprimer en grandes lignes la méthode utilisé pour mettre en place l'application et à permettre à d'autres étudiants de poursuivre le développement de notre application.

Nous disposons d'un serveur web, d'une base de données et d'une application Android que nous allons décrire dans ce document.

II. Serveur web

1. Outils utilisés

L'environnement de développement utilisé pour mettre en place le serveur WEB est Eclipse. C'est un IDE très complet et qui permet de faire développer notre serveur en Java EE.

Le conteneur de notre serveur est Apache Tomcat, c'est un serveur puissant et très facile d'utilisation.

Nous avons également eu la possibilité de mettre en place un projet Maven. Il s'agit d'un outil de gestion de projet logiciel pour Java, il permet de gérer les dépendances vers les bibliothèques tierces et assembler tout cela pour en faire un livrable correct. L'utilisation d'un projet Maven nous a permis de faciliter la mise en place de notre serveur.

De plus, nous avons fait appel au Framework open-source Hibernate qui facilite l'accès aux bases de données.

Hibernate apporte une solution aux problèmes d'adaptation entre le programme orienté objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

2. Les différentes classes

Notre serveur est structuré selon plusieurs classes. Les principales sont celles qui programment les bâtiments, zones, sorties. Il faut également gérer les utilisateurs, et les admins.

Ensuite, des classes sont nécessaires pour enregistrer les informations liées à la localisation tels que les signaux, les points. Une connexion à la base de données est nécessaire pour enregistrer ou manipuler les données.

De plus, des fichiers « .jsp » permettent de décrire les interfaces web, les servlets permettent de contrôler ces vues.

Des requêtes REST sont également mises en place pour récupérer les différentes actions des utilisateurs du serveur web, ces requêtes sous forme d'URI sont ensuite traitées à l'aide de la base de données. Les requêtes peuvent être du type GET pour récupérer des informations ou POST pour envoyer des données.

III. La base de données MySQL

Pour mettre en place la base de données, nous avons utilisé PHPMyAdmin. Les tables principales qui composent la base de données :

- Bâtiment
- Zone
- Sorties
- Utilisateurs

Les différentes zones et sorties sont évidemment liées au bâtiment où elles se trouvent.

La connexion à la base de données est faite une seule fois à l'aide d'Hibernate et d'un fichier de configuration : « hibernate.cfg.xml ».

Cette connexion renvoie une session qui est utilisée par la suite pour accéder à la base et y prendre des informations ou faire des modifications.

IV. Application Android

1. Outils utilisés

Les outils utilisés sont Android Studio pour développer l'application et la librairie Volley qui permet de construire très facilement des applications réseaux.

2. Activités

Les principales activités sont premièrement l'activité principale de l'application, où s'affichent la liste des bâtiments. C'est à partir de cette activité qu'on a accès à la partie Fingerprinting pour configurer le plan du bâtiment, et aussi à la partie Navigation pour se localiser et trouver la sortie la plus proche.

L'activité Fingerprint permet de configurer la position des zones et sorties réelles du bâtiment.

De plus, l'activité Navigation permet de récupérer le plan du bâtiment, se localiser et d'afficher le chemin pour l'évacuation.

Une classe Singleton est également nécessaire pour la mise en place de la connexion à la base de données.

La classe `VuePlan` est une `View` qui est appelée au sein de la navigation et du fingerprinting pour afficher le plan et pourvoir récupérer les points cliqués ou encore dessiner le chemin.

Une classe `WifiScanner` permet d'écouter les réseaux Wifi et d'identifier leur signal en fonction du RSSI.

Enfin, différentes classes telles que bâtiment, zone, sortie, signal et les points sont nécessaires à l'application Android.

3. Connexion au serveur

La connexion au serveur s'effectue à l'aide de la librairie `Volley`, qui permet à notre application Android de communiquer avec le serveur à l'aide de services REST. Les données sont présentées sous forme JSON.

Les URI permettent d'accéder aux services du serveur web, les données sont renvoyées sous forme JSON, et l'application fait une conversion des informations pour les manipuler selon son programme.

C'est de cette manière que nous récupérons la liste des bâtiments de la base de données. Par exemple : l'URI pour accéder aux bâtiments est :

@adresseIP:8080/BestServeur/webapi/batimentrest/findallbatiment

Cette requête GET est traitée dans le serveur, ensuite le résultat est envoyé à l'application.