



FishingDay

Dossier de conception fonctionnelle et technique

Auteur
Ronan LE HAY

TABLE DES MATIERES

1 - Versions.....	3
2 - Introduction	4
2.1 - Objet du document	4
2.2 - Références.....	4
2.3 - Besoin du client.....	4
2.3.1 - Contexte.....	4
2.3.2 - Contrainte.....	4
3 - Description générale de la solution.....	5
3.1 - Les Acteurs.....	5
3.2 - Application tierces	5
3.3 - Principe de fonctionnement.....	6
3.4 - Cas d'utilisation général	7
3.4.1 - Cas d'utilisation : Obtention des informations d'un spot.....	8
3.4.2 - Cas d'utilisation : Consultation du compte utilisateur.....	9
4 - Architecture Technique.....	10
4.1 - Application web	10
4.2 - Base de données	10
4.3 - Services Tiers	11
5 - Architecture de déploiement.....	12
5.1 - Déploiement de l'application	12
5.2 - Serveur de déploiement	12
6 - Glossaire.....	13

1 - VERSIONS

Auteur	Date	Description	Version
RLH	13/07/2018	Création du document	1.0
RLH	16/08/2018	Modification du document après début du développement	2.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception de l'application **FishingDay**. Il regroupe à la fois la partie fonctionnelle et la partie technique.

L'objectif du document est de présenter les besoins de l'utilisateur et de décrire la solution qui va être implémentée pour répondre à ces besoins.

Les éléments du présent dossier découlent :

- Du cahier des charges réalisé en amont.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **Cahier des charges de l'application FishingDay**

2.3 - Besoin du client

2.3.1 - Contexte

L'application FishingDay a pour but de fournir à son utilisateur des informations utiles pour la préparation d'une sortie de pêche en mer. Elle a donc pour objectif de collecter des informations concernant la météo, les horaires des marées et les fonds marins d'un endroit donné et de les présenter à l'utilisateur sous forme d'un tableau de bord.

Le projet est présenté de façon plus complète dans le cahier des charges cité ci-dessus.

2.3.2 - Contrainte

Afin d'être fonctionnelle, l'application a besoin de récolter un certain nombre de données. La principale difficulté de ce projet sera donc de trouver des sources de données qui soient exploitables. Cette problématique est traitée au paragraphe **4.3**.

3 - DESCRIPTION GENERALE DE LA SOLUTION

D'après l'analyse des besoins énoncés précédemment, il en est ressorti de ma réflexion un découpage de la solution en un seul package avec lequel l'utilisateur pourra interagir.

3.1 - Les Acteurs

L'application telle que présentée pour le moment ne concerne qu'un seul type d'utilisateur. Elle permettra l'inscription et la connexion pour l'utilisation d'une fonction « favoris », mais en dehors de cette fonctionnalité, il n'y a pas de différence de fonctionnement pour un utilisateur inscrit ou non inscrit.

3.2 - Application tierces

L'application **FishingDay** a pour principal objectif de centraliser des données venant de différentes sources. Pour que celle-ci fonctionne, elle a donc besoin de collecter ces données via des applications tierces.

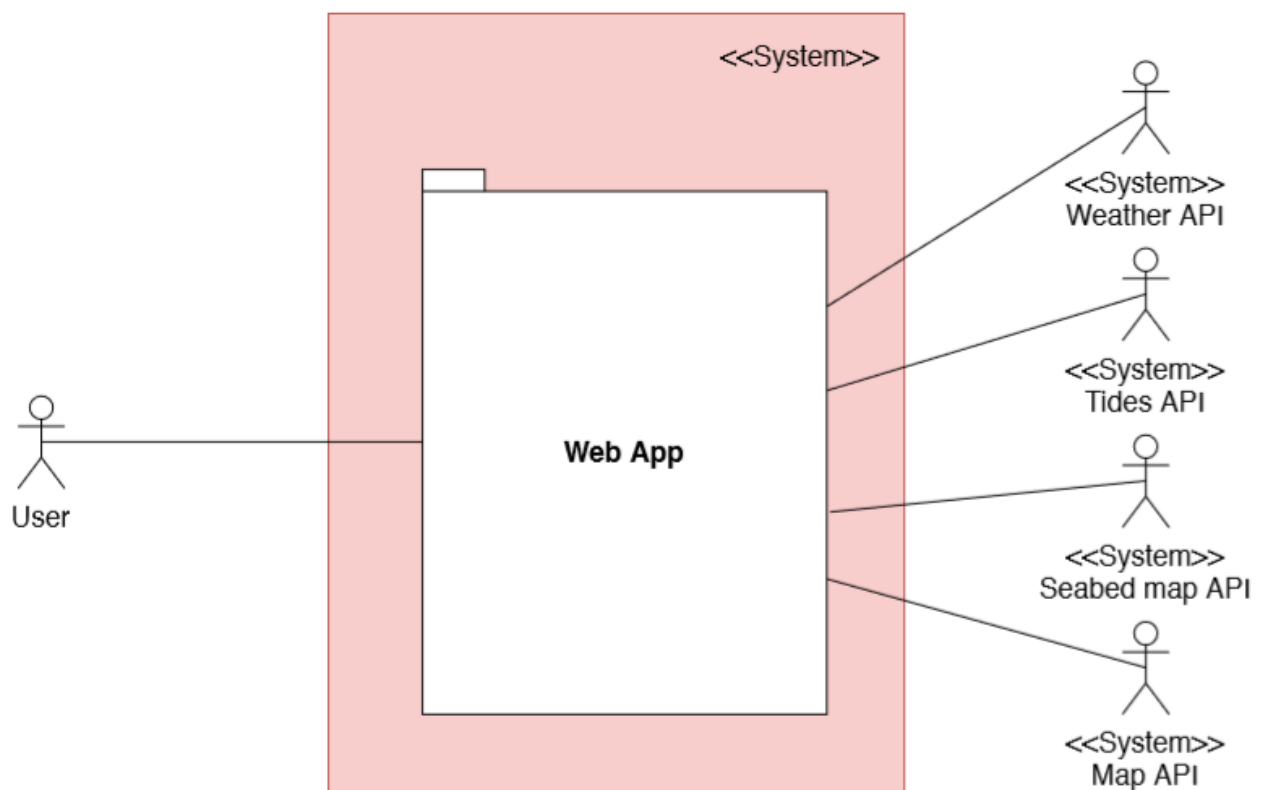
L'application aura recours à 4 applications tierces :

- Une application fournissant les données météorologiques
- Une application fournissant les données concernant les horaires des marées
- Une application permettant d'obtenir la cartographie des fonds du spot sélectionné.
- Une application permettant d'obtenir une carte de la France affichant les villes ayant déjà fait l'objet d'une recherche par un utilisateur identifié.

Pour chacun de ces besoins, plusieurs sites web mettant à disposition une API peuvent être utilisés. Le choix de la solution à utilisé sera présenté par la suite.

3.3 - Principe de fonctionnement

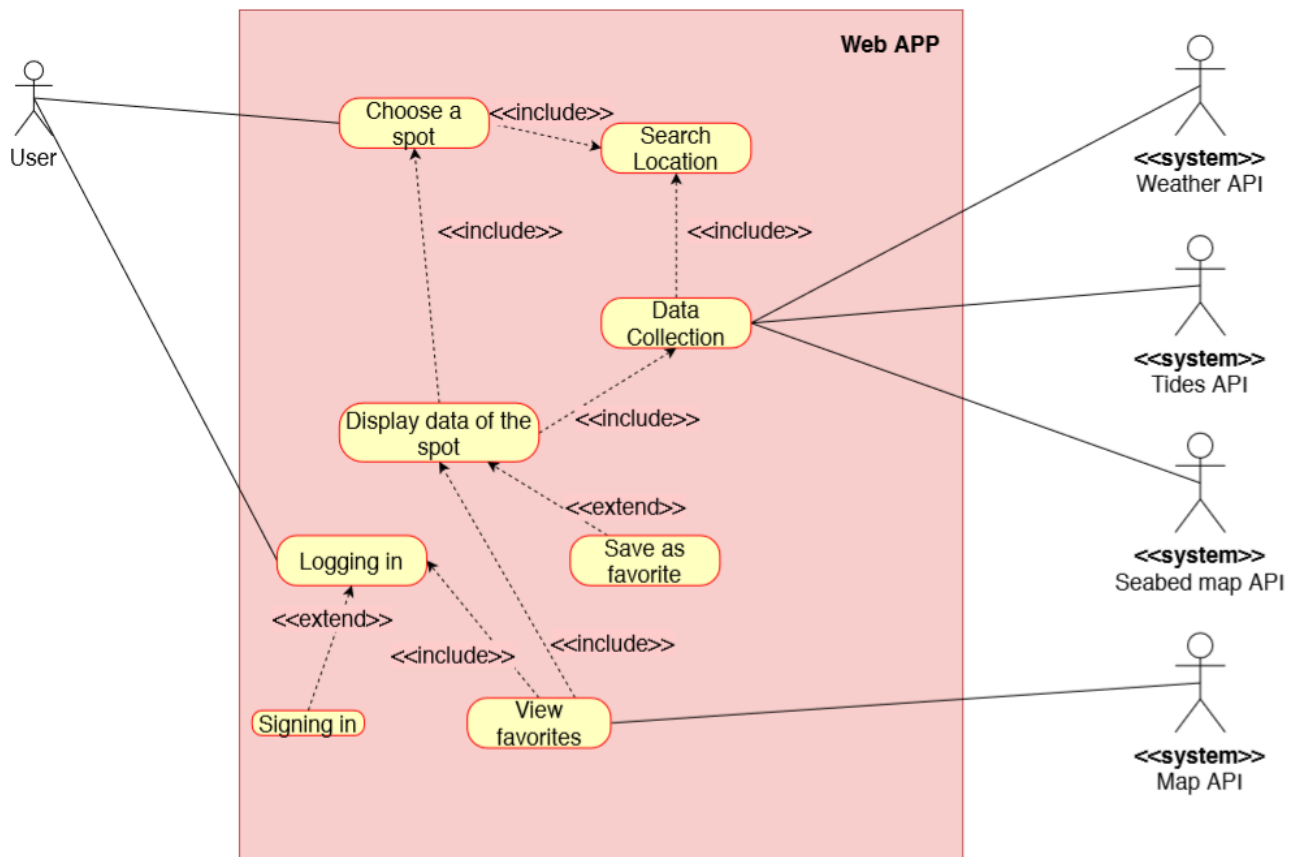
L'esprit de cette application est d'être le plus simple possible d'utilisation. L'utilisateur se connecte au site, entre le lieu désiré et obtient les données. Cette simplicité se retrouve également dans la conception de l'application comme présenté sur le diagramme de package suivant :



1. Diagramme de package

3.4 - Cas d'utilisation général

Le fonctionnement de l'application est détaillé ci-dessous :



2. Diagramme de cas d'utilisation

3.4.1 - Cas d'utilisation : Obtention des informations d'un spot

Nom : Consultation des données climatique d'un spot

Acteur(s) : Utilisateur

Description : L'utilisateur cherche à obtenir les informations utiles concernant un spot de pêche.

Auteur : Ronan LE HAY

Date(s) : 17/07/2018

Pré-conditions : Aucune.

Démarrage : L'utilisateur s'est connecté sur la page d'accueil de l'application.

DESCRIPTION

Le scénario nominal

1. **L'utilisateur** entre un lieu dans le champ de recherche.
2. **Le système** fait appel au cas d'utilisation interne « Choose a spot »
3. **Le système** fait appel au cas d'utilisation interne « Search location »
4. **Le système** fait appel au cas d'utilisation « Data Collection »
5. **Le système** fait appel aux services externes « Weather API », « Tides API » et « Seabed map API »
6. **Le système** affiche les données récoltées sur la page web.

Les scénarios d'exception

- 7 L'utilisateur est connecté, la recherche est automatiquement sauvegardée en base de données.

FIN

Scénario nominal : Sur décision de l'utilisateur après avoir consulté les données souhaitées.

Scénario d'exception : Idem scénario nominal.

POST CONDITION

Scénario nominal : Les données se sont affichées sur la page de résultat.

Scénario d'exception : Le spot a été ajouté aux favoris de l'utilisateur.

PROBLEME NON RESOLU

Que faut-il afficher lors qu'un ou plusieurs des services tiers ne renvoi aucune données pour le spot recherché ?

-> Si aucune donnée n'est renvoyée, la page d'accueil affiche un message invitant l'utilisateur à réessayer avec une autre commune. (Ajouté le 16/08/2018)

3.4.2 - Cas d'utilisation : Consultation du compte utilisateur

Nom : Consultation des favoris

Acteur(s) : Utilisateur

Description : L'utilisateur cherche à obtenir les informations sur un de ses spots préalablement recherchés.

Auteur : Ronan LE HAY

Date(s) : 17/07/2018

Pré-conditions : L'utilisateur doit être connecté.

Démarrage : L'utilisateur a demandé la page « Mon Compte »

DESCRIPTION

Le scénario nominal

1. **Le système** fait appel au cas d'utilisation interne « View favorites »
2. **Le système** affiche la liste des spots déjà recherchés par l'utilisateur.
3. **L'utilisateur** choisi un spot.
4. **Le système** fait appel au cas d'utilisation « Display data of the spot »
5. **Le système** fait appel au cas d'utilisation « Data Collection »
6. **Le système** fait appel aux services externes « Weather API », « Tides API » et « Seabed map API »
7. **Le système** affiche les données récoltées sur la page web.

FIN

Scénario nominal : Sur décision de l'utilisateur après avoir consulté les données souhaitées.

POST CONDITION

Scénario nominal : Les données se sont affichées sur la page de résultat.

4 - ARCHITECTURE TECHNIQUE

4.1 - Application web

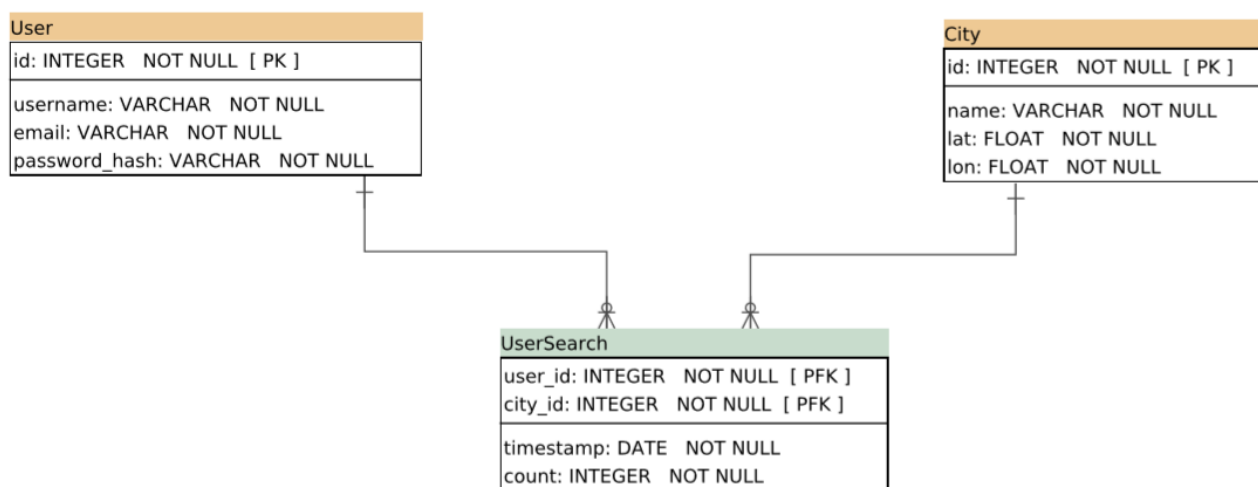
Le développement de l'application web sera fait en langage python (version 3.5) avec l'utilisation du Framework Django (version 2.0).

4.2 - Base de données

L'utilisation d'une base de données est nécessaire afin de stocker les informations des utilisateurs souhaitant créer un compte (Stockage des informations de l'utilisateur, stockage des favoris...).

Le SGBD utilisé sera PostgreSQL afin de faciliter le déploiement sur la plateforme Heroku où sera hébergée l'application (cf 5.1 Déploiement de l'application).

Le modèle physique de données ci-dessous présente en détail les informations qui seront stockées dans la base de données.



3. Modèle Physique de Données

Description des tables :

- **User :**
Cette table contiendra uniquement les éléments essentiels concernant l'utilisateur, à savoir un nom d'utilisateur, une adresse mail et le hash du mot de passe.

- **City :**
Cette table sera remplie à chaque fois qu'un utilisateur recherche une ville. On stockera alors le nom de la ville et ses coordonnées géographiques.
- **UserSearch :**
Cette table aura pour but de stocker les villes recherchées par un utilisateur. On stockera également la date de la recherche pour pouvoir afficher par la suite les dernières villes cherchées par l'utilisateur, ainsi que le nombre de fois où cette recherche a été faite pour pouvoir créer automatiquement une liste de villes favorites.

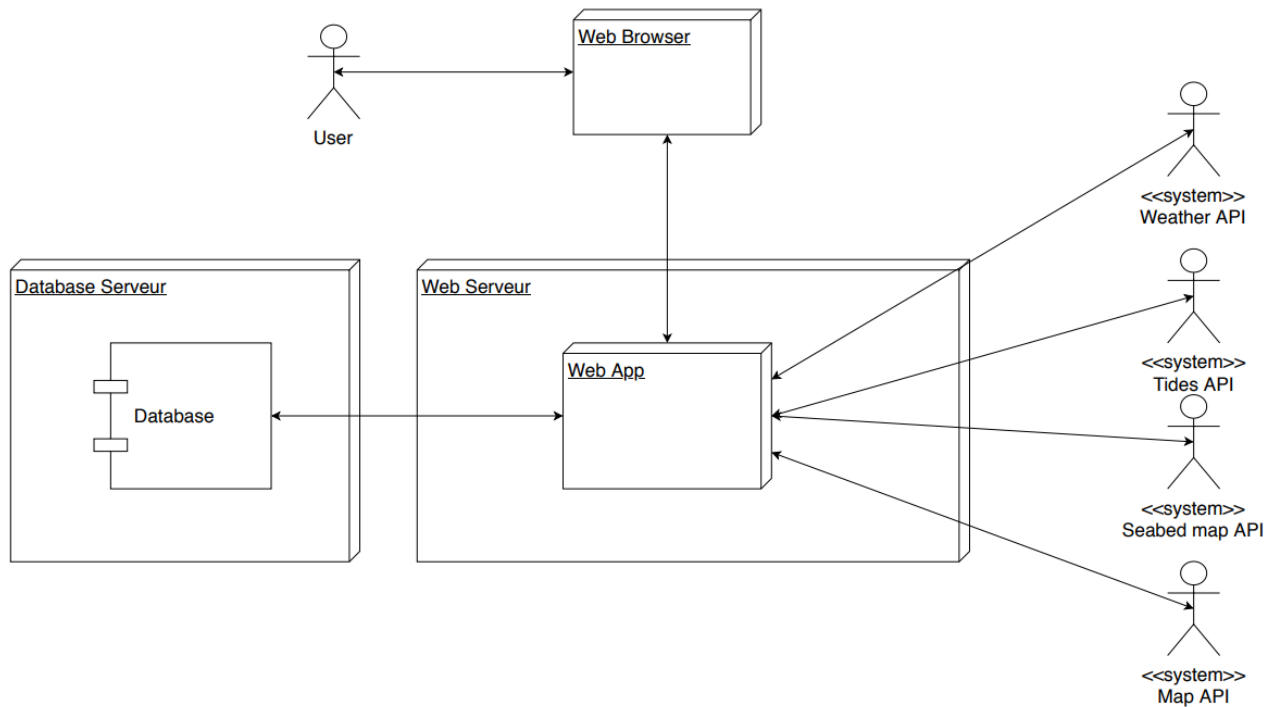
4.3 - Services Tiers

Le fonctionnement de l'application nécessite l'utilisation de services tiers pour obtenir les données à afficher à l'utilisateur. Le tableau ci-dessous résume les besoins de l'application et les solutions qui seront utilisées pour y répondre :

Service	Rôle	Solution Retenue
"Weather API"	-Obtenir les informations météorologiques du spot via les coordonnées géographiques.	API de OpenWeatherMap
"Tides API"	-Obtenir les horaires des mares pour le spot choisi.	API Worldtides.info
"Seabed API"	-Obtenir la carte des fonds marins du spot.	API Navionics
"Map API"	-Afficher la carte de France avec un pointeur sur chaque ville déjà recherché.	API Google maps

5 - ARCHITECTURE DE DEPLOIEMENT

5.1 - Déploiement de l'application



4. Diagramme de déploiement

5.2 - Serveur de déploiement

L'application **FishingDay** sera déployée sur les serveurs de Heroku, un service de cloud-computing de type Plateforme as a Service (PaaS).

Cette solution permet un déploiement facile à mettre en œuvre et gratuit.

6 - GLOSSAIRE

SGBD	Système de Gestion de Base de Données
-------------	---------------------------------------