Documentation de projet

Phobos

Audergon Vincent, Kolly Florian

Projet du 19.09.2020 au jj.mm.aaaa

Version préliminaire du 19.09.2020

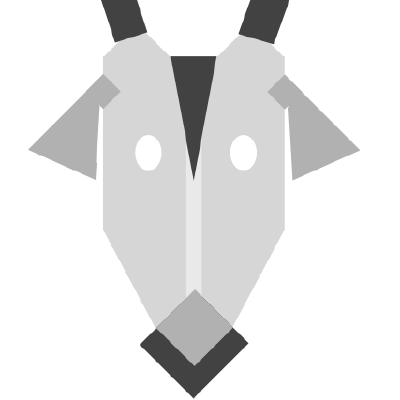


Table des matières

[1 Introduction 3](#_Toc51429613)

[2 Contexte de l’application 4](#_Toc51429614)

[3 Analyse des besoins 5](#_Toc51429615)

[3.1 Use case 5](#_Toc51429616)

[3.2 Maquettes 5](#_Toc51429617)

[3.3 Diagrammes de séquences 5](#_Toc51429618)

[3.4 Diagrammes d’activités 5](#_Toc51429619)

[4 Spécificités matérielles 6](#_Toc51429620)

[5 Spécificités logiciels 7](#_Toc51429621)

[6 Conception 8](#_Toc51429622)

[6.1 Frontend 8](#_Toc51429623)

[6.2 Backend 8](#_Toc51429624)

[6.3 Local Database 8](#_Toc51429625)

[6.4 Global Database (V2) 8](#_Toc51429626)

[7 Tests unitaires et fonctionnels 9](#_Toc51429627)

[7.1 Tests unitaires 9](#_Toc51429628)

[7.2 Tests fonctionnels 9](#_Toc51429629)

[8 Problèmes et solutions 10](#_Toc51429630)

[9 Glossaire 11](#_Toc51429631)

[10 Structure du projet 12](#_Toc51429632)

[11 Bibliographie 13](#_Toc51429633)

# Introduction

Le projet consiste à mettre en place une application mobile permettant d’enregistrer des éléments (nommés items dans la suite du document) faisant parties d’une collection commune. Le but de l’application étant d’être la plus générique possible, ces collections sont entièrement personnalisables par chaque utilisateur, via un système de types. Par exemple, un utilisateur souhaite enregistrer et évaluer les bières qu’il consomme :

* La collection est "bière"
* Les items sont les différentes bières qu’il boit et enregistre
* Les types sont les données qu’il souhaite enregistrer pour ses bières
  + Photo
  + Pourcentage d’alcool
  + Marque
  + Evaluation (de 0 à 5 étoiles)
  + …

L’application est planifiée pour être développée en 2 versions distinctes et distantes dans le temps. La seconde version remplacera la première en y ajoutant de nombreuses fonctionnalités. Cette seconde version n’est pas décrite en longueur ici, mais certaines fonctionnalités sont citées et indiquées avec le terme "V2" entre parenthèses.

## Terminologie : Collection

Une collection représente la catégorie des objets qui y seront enregistrés. On peut citer par exemple une collection "Jeux vidéo" pour un utilisateur souhaitant en faire la liste et les évaluer par catégorie (graphisme, histoire, …). Un autre exemple peut être la liste des musées visités, avec leur location.

Les utilisateurs pourront créer leur propre collection (sans limite de nombre), mais des modèles (templates) seront mis à disposition.

## Terminologie : Item

Un item est un élément de la collection. Cet élément possède des valeurs pour chaque type de la collection (il est possible que la valeur ne soit pas définie sur certains types). Pour reprendre les mêmes exemples que le chapitre précédent, il est possible d’avoir en item le jeu "Super Meat Boy" pour la collection "Jeux vidéo" et le Musée du Louvre pour la collection "Musées visités".

## Terminologie : Type

Le but des types est de permettre à l’application d’être la plus générale possible. De nombreux types sont mis à disposition de base, mais l’utilisateur peut aussi les personnaliser ou en créer des nouveaux. En reprenant les mêmes exemples que précédemment, on peut décider d’avoir par exemple les types suivants pour la collection "Jeux vidéo" :

* Note globale : nombre compris dans [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5] représenté par des étoiles
* Note graphisme : nombre entier, entre 1 et 10
* Jaquette : image
* Editeur : chaîne de caractères
* Catégorie : élément d’une liste [aventure, action, …]

Dans le cas de la collection "Musées visités", on peut retrouver :

* Position géographique : localisation Google Maps
* Liste des œuvres majeurs : liste de chaînes de caractères
* Prix d’entrée : valeur monétaire

## Technologie

### Frontend

Le frontend est réalisé avec la technologie JavaScript moderne React Native, permettant ainsi un code indépendant de la plateforme (Android ou IOS). Le téléphone portable de l’utilisateur est nommé téléphone hôte dans la suite du document.

* Application React Native sur téléphone hôte
* Appareil(s) photo(s) du téléphone hôte
* Système de géolocalisation du téléphone hôte

### Backend

Le backend est principalement réalisé directement sur l’application en React Native, mais un serveur distant est aussi présent pour gérer les utilisateurs et les collections synchronisées (V2).

* Application React Native sur téléphone hôte
* Serveur Node.JS distant

### Databases

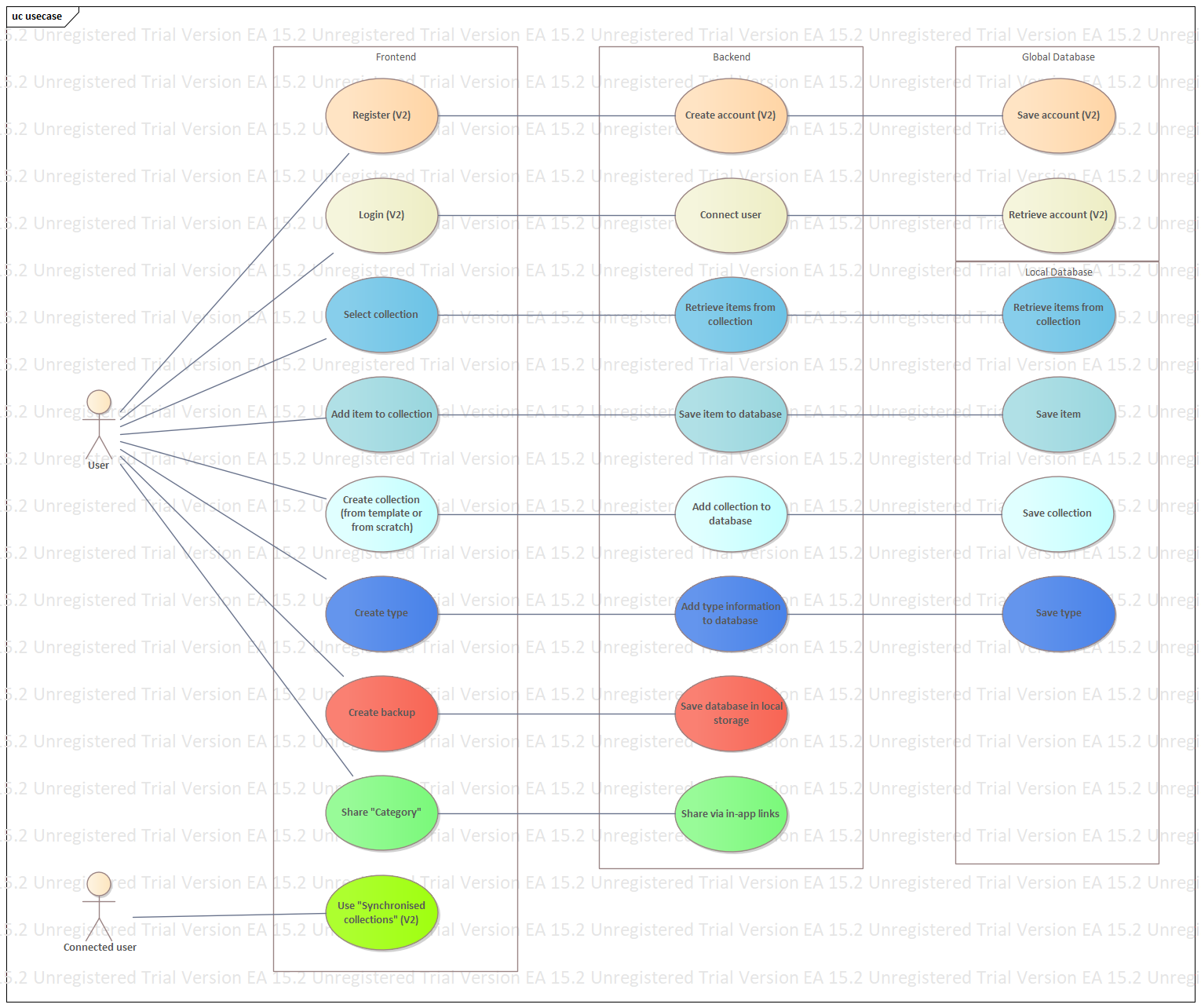
Deux types de bases de données sont utilisées pour ce projet. La première est le stockage local (sur le téléphone) des données (collections, items et types). L’enregistrement de ces informations ne nécessite pas d’envoi à un serveur externe, mais nécessite les permissions d’accès et d’enregistrement des données sur le téléphone hôte.

La seconde base de données permet d’enregistrer les utilisateurs (V2) et les catégories synchronisées (V2).

# Contexte de l’application

# Analyse des besoins

## Use case



## Maquettes

## Diagrammes de séquences

## Diagrammes d’activités

# Spécificités matérielles

# Spécificités logicielles

# Conception

## Frontend

## Backend

## Local Database

## Global Database (V2)

# Tests unitaires et fonctionnels

## Tests unitaires

## Tests fonctionnels

# Problèmes et solutions

# Glossaire

# Structure du projet

# Bibliographie