

Cahier des Charges - Application CheckYourMood v2

Franck Silvestre

Table of Contents

1. Introduction	1
2. Audit qualité de l'application	1
3. Améliorations fonctionnelles	2
3.1. <i>Feature</i> F004 "Client mobile Android CheckYourMood"	3
4. Exigences techniques	3
4.1. Architecture de l'application	3
4.2. Environnements techniques	4
4.3. Contraintes de développement	4

1. Introduction

La startup *CheckYourMood* souhaite développer une deuxième version de son application Web éponyme.

La version 1 de l'application permet à un utilisateur enregistré sur la plateforme d'indiquer son humeur à différents moments de la journée afin de pouvoir disposer de visualisations sur son humeur au fil du temps en fonction de différents paramètres comme la période, le type d'humeur, etc. Pour plus de détails sur la version 1 de l'application, vous pouvez consulter le cahier des charges de la version 1 (cf. SAE du semestre 3).

La version 2 de l'application vise 2 objectifs :

- Améliorer la qualité de l'application (code, tests, impact environnemental, sécurité).
- Fournir une version mobile Android de la saisie des humeurs.

Ce document présente les exigences produits permettant de livrer à la startup la deuxième version de son projet en temps et en heure.

Dans la suite du document, les termes "application", "plateforme" ou "logiciel" désignent de manière indifférente l'application Web *CheckYourMood*.

2. Audit qualité de l'application

Le premier travail de l'équipe concerne la réalisation d'un audit de la version 1 de l'application. Cet audit permettra d'identifier les points faibles de l'application et ainsi de cartographier les points d'amélioration de l'application sur différentes caractéristiques. Une démonstration de la faisabilité de la version 2 de l'application constituera la dernière partie de l'audit.

AUD001 - Livrable audit

À l'issue de l'audit, un document contiendra la restitution des résultats de l'audit.

- Le document aura pour nom "SAE-S4-Audit-CheckYourMood-V1-<Noms membres équipe>".
- Le document sera structuré en 4 parties en plus de l'introduction et de la conclusion :
 1. Évaluation
 2. Plan d'amélioration
 3. Choix de la configuration des machines virtuelles
 4. Démonstration de la faisabilité

Le document, en tant que livrable produit, sera accessible en ligne avec accès en mode suggestion à toutes les parties prenantes.

AUD002 - Caractéristiques à évaluer

La version 1 de l'application devra faire l'objet d'une évaluation sur les caractéristiques suivantes :

- La documentation technique et fonctionnelle du code.
- La qualité du code.
- La couverture du code par les tests et la qualité des tests.
- La qualité du dépôt Git et de l'historique des commits.
- L'impact environnemental de l'application (<https://www.website-footprint.com/fr/>).
- Les performances de l'application.
- La sécurité de l'application.

AUD003 - Utilisation de PHPStan

L'outil d'analyse statique de code [PHPStan](#) devra être mis en place pour soutenir l'évaluation de la qualité du code.

AUD004 - Plan d'amélioration

L'évaluation de la version 1 de l'application doit conduire à un plan d'amélioration décrivant précisément les actions qui seront menées sur chacune des caractéristiques évaluées dans le cadre du développement de la version 2 de l'application.

Ce plan d'amélioration prendra la forme d'une liste d'exigences techniques, de qualité, etc.

AUD005 - Choix de la configuration des machines virtuelles

Les contraintes de développement décrivent des exigences relatives à l'utilisation de la virtualisation pour faciliter la recette du projet (DEV003 et DEV004). La partie "Choix de la configuration des machines virtuelles" doit contenir les spécifications des machines virtuelles à mettre en place pour répondre aux exigences DEV003 et DEV004.

AUD006 - Démonstration de la faisabilité du projet

La partie "Démonstration faisabilité" fera l'objet d'une présentation des arguments avancés par l'équipe pour démontrer sa capacité à réaliser le plan d'amélioration dans les délais impartis.

3. Améliorations fonctionnelles

Les exigences fonctionnelles sont décrites au format User Story. Elles sont regroupées en *features*.

3.1. *Feature* F004 "Client mobile Android CheckYourMood"

Cette feature comprend les fonctionnalités relatives à l'utilisation d'un client mobile Android pour utiliser l'application. Dans la suite du document, l'application mobile CheckYourMood Mobile sera désignée par le label CYMDroid.



CYMDroid est un client mobile s'interfaçant à l'application CheckYourMood. CYMDroid ne stocke pas de données sur le client mobile. Les données sont récupérées ou modifiées via l'application CheckYourMood via une API REST comme cela est rappelé dans les exigences techniques.

US041 - Authentication

En tant qu'utilisateur ayant un compte sur la plateforme,
Je veux m'authentifier avec un login et un mot de passe depuis mon application CYMDroid,
Pour avoir accès à certaines fonctionnalités de CheckYourMood depuis mon mobile avec une expérience utilisateur optimale.

US042 - Consultation des dernières humeurs saisies

En tant qu'utilisateur authentifié,
Je veux consulter la liste des 5 dernières humeurs saisies,
Pour prendre connaissance de la tendance de mes dernières humeurs.

US043 - Saisie d'une humeur

En tant qu'utilisateur authentifié sur CYMDroid,
Je veux ajouter une humeur à tout instant de la journée,
Pour obtenir à terme le plus d'information possible sur mon humeur au fil du temps.

4. Exigences techniques

4.1. Architecture de l'application

ARC001 - Architecture 3-tiers

L'application est une application Web s'appuyant sur une architecture 3-tiers.

ARC002 - Modele Vue Contrôleur

L'application est une application conçue sur la base du *design pattern* MVC.

ARC003 - API REST

L'application fournit une API REST permettant au client mobile de communiquer avec le *back end* pour réaliser les fonctionnalités attendues du client mobile.

4.2. Environnements techniques

TECH001 - SGBD Relationnel

MySQL version 8 ou supérieure.

TECH002 - Langages de programmation *back-end*

Php version 8 ou supérieure.

TECH003 - Langages de programmation *front-end* web

HTML 5, librairie Bootstrap version 5 ou supérieure pour les apports CSS et Javascript.

TECH004 - Gestion de version de code source

Git version 2.32 ou supérieure

TECH005 - Environnement de programmation *front-end* mobile

Android version 9. Le *front-end* mobile devra fonctionner sur une machine virtuelle Android (cf. DEV003).

4.3. Contraintes de développement

DEV001 - Tests automatisés sur les services

Le code des services métier développés dans le Modèle de l'application fait l'objet d'une couverture de code par les tests automatisés supérieure à 80%.

DEV002 - Tests automatisés sur les contrôleurs

Le code des contrôleurs de l'application fait l'objet d'une couverture de code par les tests automatisés supérieure à 80%.

DEV003 - Amélioration des tests automatisés

La répartition entre les tests unitaires et tests d'intégration doit se conformer à l'esprit de la pyramide des tests de M. Cohn.

DEV004 - Utilisation de PHPStan

L'outil d'analyse statique de code [PHPStan](#) devra être mis en place pour maintenir la qualité du code tout au long du projet.

DEV005 - Projet CYMDroid "virtualisé"

Le projet CYMDroid doit être testable sur un système Android déployé dans une machine virtuelle sous VirtualBox.

DEV006 - Projet CheckYourMood "virtualisé"

Le projet CheckYourMood V2 doit être testable sur un système LAMP^[1] déployé dans une machine virtuelle sous VirtualBox.

[1] Linux Apache MySQL PHP