

Cahier des Charges - Application Festiplan 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> F005 "Client mobile Android Festiplan"	2
4. Exigences techniques	4
4.1. Architecture de l'application	4
4.2. Contraintes de développement	4
4.3. Environnements techniques	5
5. Livrables produit	5
5.1. Audit qualité de l'application	5
5.2. Diaporama de la restitution en anglais du travail sur l'impact environnemental	5
5.3. Les exigences produit	6
5.4. Le code source	6
5.5. Les machines virtuelles	6
5.6. Les documentations (hors VM)	6

1. Introduction

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

La version 1 de l'application permet la gestion de la planification des spectacles (pièces de théâtre, concerts) sur les différentes scènes mise à disposition par les organisateurs du festival. 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 consultation des festival planifiés.

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 *Festiplan*.

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-Festiplan-V1-<Noms membres équipe>".
- Le document sera structuré en 3 parties en plus de l'introduction et de la conclusion :
 1. Évaluation
 2. Plan d'amélioration
 3. 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.
- 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 - 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.

AUD006 - Restitution en anglais du travail sur l'impact environnemental

L'évaluation de l'impact environnemental et le plan d'amélioration associé à cette partie de l'évaluation feront l'objet d'une restitution orale en anglais. Cette soutenance en anglais s'appuiera sur un diaporama rédigé en anglais.

Le document contenant le diaporama aura pour nom "SAE-S4-Environment-Impact-<Noms membres équipe>"

3. Améliorations fonctionnelles

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

3.1. *Feature* F005 "Client mobile Android Festiplan"

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 Festiplan Mobile sera désignée par le label FestiplAndroid.



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

F005US001 - Authentification

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 FestiplAndroid,
Pour avoir accès à certaines fonctionnalités de Festiplan depuis mon mobile avec une expérience utilisateur optimale.

F005US002 - Consultation des festivals programmés

En tant qu'utilisateur authentifié,
Je veux consulter la liste des prochains festivals programmés,
Pour être informé des festivals qui pourraient m'intéresser.

F005US003 - Consultation du détail d'un festival programmé

En tant qu'utilisateur authentifié,
Je veux consulter la liste des spectacles d'un festival programmé,
Pour avoir une idée plus précise sur un festival qui pourrait m'intéresser.

F005US004 - Mise en favori d'un festival

En tant qu'utilisateur authentifié sur FestiplAndroid,
Je veux mettre en favori un festival,
Pour le retrouver plus rapidement lors de ma consultation des festivals.

F005US005 - Consultation des festivals programmés mis en favoris

En tant qu'utilisateur authentifié sur FestiplAndroid,
Je veux consulter la liste de mes festivals favoris,
Pour retrouver plus rapidement les festivals que j'avais repéré.

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. Contraintes de développement

DEV001 - Tests automatisés sur les nouveaux services

Le code des nouveaux 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 - 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.

DEV003 - Projet FestiplAndroid "virtualisé"

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

DEV004 - Projet Festiplan "virtualisé"

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

4.3. 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 virtualisation

Virtual Box version 7.0.14.

TECH006 - Environnement de programmation *front-end* mobile

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

5. Livrables produit

5.1. Audit qualité de l'application

Voir exigence AUD001 du cahier des charges.

5.2. Diaporama de la restitution en anglais du travail sur l'impact environnemental

Voir exigence AUD007 du cahier des charges.

5.3. Les exigences produit

Identifiant livrable	Libellé court	Description
SPEC001	Backlog produit	Le backlog pour les user stories et autres exigences non transverses.
SPEC002	Définition de fini	Le document de "définition de fini" pour les exigences transverses.
SPEC003	Diagrammes de cas d'utilisation	Les diagrammes de cas d'utilisation de la ou des applications à développer.

5.4. Le code source

Identifiant livrable	Libellé court	Description
SRC001	Code source	Le code source hébergé sur Github.

5.5. Les machines virtuelles

Identifiant livrable	Libellé court	Description
VM001	Machines virtuelles	Les images des machines virtuelles exigées par le cahier des charges sur clé USB fournie par l'équipe de développement.
VM002	Documentation VM	Le mode d'emploi pour la configuration de VirtualBox (si besoin) et l'installation des machines virtuelles.

5.6. Les documentations (hors VM)

Identifiant livrable	Libellé court	Description
DOC001	Documentation utilisateur	La documentation utilisateur peut être multiple, administrateur, utilisateur, À chaque sprint ayant ajouté des fonctionnalités utilisateurs, la documentation utilisateur doit être maintenue à jour.

Identifiant livrable	Libellé court	Description
DOC002	Documentation technique	<p>La documentation technique doit être composée :</p> <ul style="list-style-type: none"> • d'une description de l'architecture technique du projet • d'une description des technologies utilisées • d'une description de points spécifiques techniques particuliers dépendants du projet (par exemple la sécurité sur un projet web, la présentation d'un algorithme complexe, etc.). • Et de tous les éléments demandés par les enseignants évaluateurs. <p>À chaque sprint, la documentation technique doit être maintenue à jour.</p>



La documentation technique sera évaluée sur sa capacité à permettre à un nouvel arrivant sur le projet de comprendre et de s'appropriier la solution sur le plan technique.