

simon.daguenet@epitech.eu

Music Sheet Writer

Music Sheet Writer

[Sujet du document]

Objectifs du document

Résumé

Ce document est le Template des documents pour le projet MusicSheetWriter. Il comprend les éléments de base à rendre au Lab EIP : Résumé, Glossaire, Description du document, Sommaire ; et définie également les styles utilisé pour la présentation du document.

Glossaire

***– M –***

Mot : Définition

***– S –***

Signification de l’Acronyme (SDA) : Définition

Description du document

|  |  |
| --- | --- |
| Titre | Music Sheet Writer : [Sujet du document] |
| Date de création | [Date de création] |
| Date de publication | [Date de publication] |
| Auteur | simon.daguenet@epitech.eu |
| Responsable | Jonathan Racaud |
| E-mail | musicsheetwriter\_2017@labeip.epitech.eu |
| Sujet | [Sujet du document] |
| Version du modèle |  |

Tableau des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Auteur | Section(s) | Commentaire |
|  |  |  |  |

Sommaire

[1. Partie 1 2](#_Toc434592146)

[1.1. Sous-partie 1.1 2](#_Toc434592147)

[1.2. Sous-partie 1.2 2](#_Toc434592148)

[1.3. Sous-partie 1.3 2](#_Toc434592149)

[2. Partie 2 2](#_Toc434592150)

Liste des Tableaux

[Tableau 1: Tableau en Row-Heading 2](#_Toc434592579)

[Tableau 2: Tableau en Column-Heading 2](#_Toc434592580)

[Tableau 3: Tableau en Double-heading 2](#_Toc434592581)

Liste des Figures

[Figure 1: Nom de l'image 2](#_Toc434592447)

# 

# Livrables

## Android

### Cas de tests

#### L’arborescence

Le livrable des cas de tests pour l’application Android suivra la même arborescence que dans leur implémentation.

Les cas de tests seront regroupés par type de test (Acceptation, Intégration, Unitaire, etc.), campagne de test (Campagne, Activité, Fragment, Composant, etc.), et suite de test (Scenario, UI, appelAPI, Class, etc.). Chaque cas de test étant une méthode de la classe de test.

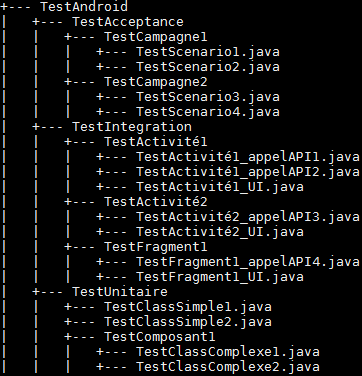


Figure 1: Exemple d’arborescence des fichiers de tests Android

Cette organisation permettra tout d’abord de comprendre les enjeux et le contexte Android dans lequel sont joués les tests. De plus, il permettra aux développeurs de retrouver facilement le code de chaque test au sein du code source.

#### Tests Unitaire

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre l’élément testé |
| Les données utilisées en temps qu’entré pour le test | Savoir quelle valeur sont testés et par substitution, lesquels ne le sont pas |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

#### Test Intégration et d’acceptation

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre la fonctionnalité testé et comprendre le contexte dans lequel se trouve le user s’il devait faire le test manuellement |
| Les données utilisées en temps qu’entré pour le test | Savoir quelles valeurs sont utilisées pour le test |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| Le type de scénario qu’il simule (voir ci-dessous) | Connaitre la nature du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

Le type de scénario peut être :

* Basic : c’est le flow d’exécution tel qu’il est censé se produire en condition d’utilisation normale
* Alternative : c’est le flow d’exécution tel qu’il est censé se produire en condition d’utilisateur particulière
* Exception : c’est le flow d’exécution tel qu’il est censé se produire en condition d’erreur

Le livrable des cas de tests devra être mis à jour aussitôt qu’une campagne de test est terminé d’être codé, même si tous les tests ne sont pas en succès.

### Rapport d’exécution de tests

Le livrable présentant les cas de tests pour Android sera sous la forme d’un fichier HTML. Il présentera tous les tests joués, regroupé par suite et campagne de tests. Les résultats de ces derniers sera données et, pour les tests ayant échoué, le fil d’exécution du test sera affiché afin de montrer exactement les raisons qui ont amené le test à échouer.



Figure 2: Résultats des tests Android

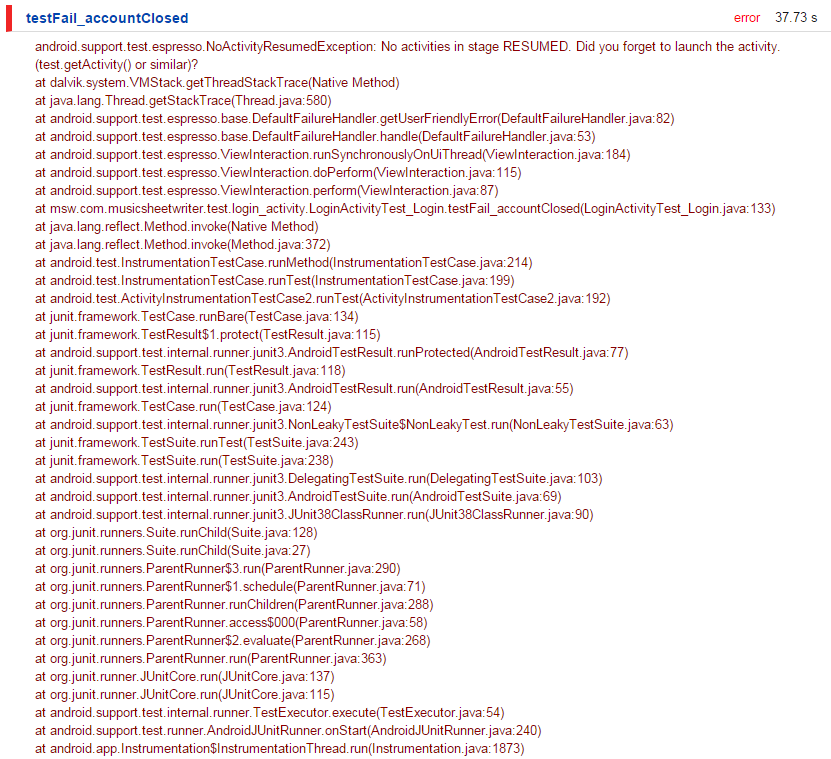


Figure 3: Fil d'exécution des tests en échec

### Rapport sur le suivi global de la qualité

## IPhone

### Cas de tests

Les cas de tests pour l’IPhone seront chaque nouvelle implémentation de fonctionnalité (le login, la création de compte, ..). Ce système permet une fréquence de mise à jour des tests corrects.

#### Tests Unitaire

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre l’élément testé |
| Les données utilisées en temps qu’entré pour le test | Savoir quelle valeur sont testés et par substitution, lesquels ne le sont pas |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

#### Test Intégration et d’acceptation

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre la fonctionnalité testé et comprendre le contexte dans lequel se trouve le user s’il devait faire le test manuellement |
| Les données utilisées en temps qu’entré pour le test | Savoir quelles valeurs sont utilisées pour le test |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| Le type de scénario qu’il simule (voir ci-dessous) | Connaitre la nature du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

Le type de scénario peut être :

* Basic : c’est le flow d’exécution tel qu’il est censé se produire en condition d’utilisation normale
* Alternative : c’est le flow d’exécution tel qu’il est censé se produire en condition d’utilisateur particulière
* Exception : c’est le flow d’exécution tel qu’il est censé se produire en condition d’erreur

Le livrable des cas de tests devra être mis à jour aussitôt qu’une campagne de test est terminé d’être codé, même si tous les tests ne sont pas en succès.

### Rapport d’exécution de tests

Il y a deux formes de rapport :

* Avec export
* Sans export

La version sans export est celle lancé directement depuis *xcode*. Les résultats sont clairs à la fin de l’exécution et si une erreur se produit, la ligne d’où provient l’erreur est clairement mise en valeur avec une explication de l’erreur.

Le problème est qu’il n’y a aucun export et donc après fermeture de *xcode* aucune preuve.

Pour le test avec export, il faut lancer depuis un terminal en ligne de commande. Le seul problème est qu’on ne peut pas lancer de test unique, ils seront tous exécutés mais en générant un rapport clair pour prouver le bon fonctionnement de l’application.

Voici Un exemple d’export possible suite à la réussite de tous les tests :

▸ Building MSW/MSW2 [Debug]

▸ Check Dependencies

▸ Building Pods/KIF [Debug]

▸ Check Dependencies

▸ Building Pods/Pods-MSW2Tests [Debug]

▸ Check Dependencies

▸ Building MSW/MSW2Tests [Debug]

▸ Check Dependencies

▸ Running script 'Check Pods Manifest.lock'

▸ Running script 'Embed Pods Frameworks'

▸ Running script 'Copy Pods Resources'

All tests

Test Suite MSW2Tests.xctest started

LogTest

✓ test00LogFail (3.399 seconds)

✓ test01MotDePasseOublie (8.548 seconds)

✓ test02Inscription (7.422 seconds)

✓ test03Log (3.926 seconds)

✓ test04Deconnexion (5.903 seconds)

✓ test05ChangerDeMotPasse (14.272 seconds)

✓ test06ChangeFirstname (5.414 seconds)

✓ test07Menu (6.868 seconds)

✓ test08Swipe (5.708 seconds)

✓ test09SwipeMenu (2.007 seconds)

✓ test10Ajouter (5.572 seconds)

✓ test11 (0.000 seconds)

LoginTest

✓ testConnexion (2.451 seconds)

✓ testConnexionfausse (1.171 seconds)

✓ testGetMethodWithId (0.687 seconds)

✓ testGetMethodWithString (0.704 seconds)

✓ testOptionsMethodWithString (0.678 seconds)

✓ testPostMethodForSuscribeWithUsername (0.778 seconds)

✓ testPostMethodWithIdentifier (1.185 seconds)

✓ testPostMethodWithString (1.170 seconds)

✓ testPutMethodWithString (0.740 seconds)

testApiMethod

✓ testGetMethodWithId (0.697 seconds)

✓ testPostMethodForSuscribeWithUsername (0.707 seconds)

✓ testPostMethodWithIdentifier (1.173 seconds)

✓ testPostMethodWithString (1.170 seconds)

Executed 25 tests, with 0 failures (0 unexpected) in 82.353 (82.365) seconds

### Rapport sur le suivi global de la qualité

## Windows phone

### Cas de tests

#### L’arborescence

Le livrable des cas de tests pour l’application Windows phone suivra la même arborescence que dans leur implémentation qui se trouvera dans le projet « test » de la solution Music Sheet Writer.

Les cas de tests pour l’application Windows phone seront chaque nouvelle implémentation de fonctionnalité (le login, la création de compte, ..). Ce système permet une fréquence de mise à jour des tests corrects.

#### Test Unitaire

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre l’élément testé |
| Les données utilisées en temps qu’entré pour le test | Savoir quelle valeur sont testés et par substitution, lesquels ne le sont pas |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

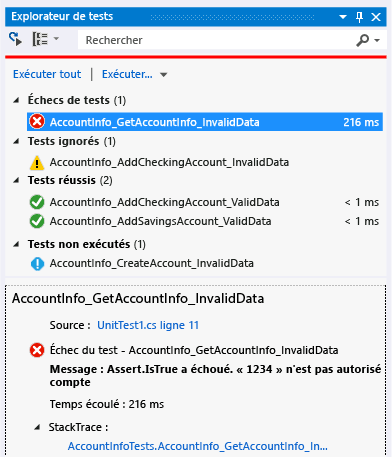
#### Test UI

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre la fonctionnalité testé et comprendre le contexte dans lequel se trouve le user s’il devait faire le test manuellement |
| Les données utilisées en temps qu’entré pour le test | Savoir quelles valeurs sont utilisées pour le test |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

### Rapport d’exécution de tests

Le rapport d’exécution de tests s’effectuera via l’interface de Microsoft Visual Studio 2015 dans un projet « unit test » qui sera ajouté à la solution projet. Dans cette interface nous sommes en mesure d’observer le résultat des tests effectués auparavant. Ainsi, nous pouvons observer le résultat d’un test réussi et un échoué.

Le livrable présentant les cas de tests pour Windows phone sera sous la forme d’un fichier (.rtf). Il présentera tous les tests joués, regroupé par suite et campagne de tests.



### Rapport sur le suivi global de la qualité

Le rapport sur le suivi global de la qualité s’effectuera grâce au test de non-régression qui permettra au chef de groupe d’avoir une vision des tests effectués précédemment par le développeur. Ainsi il sera en mesure de connaitre la qualité du code en visualisant les tests effectué. Le chef de groupe pourra également s’appuyer sur les tests fonctionnels qui lui permettront de vérifier la conformité de l'application développée avec le cahier des charges initial.

## Logiciel

### Cas de tests

L’ensemble des tests pour le logiciel se trouveront dans un répertoire Test/Logiciel. Dans ce répertoire, les tests seront classé par fonctionnalités, par exemple nous aurons un répertoire, Test/Logiciel/Gérer un projet, qui regroupe l’ensemble des tests unitaires qui sont utilisés pour Gérer un projet.

#### Tests Fonctionnels

Les tests fonctionnels visent à tester le bon déroulement d’une fonctionnalité. Cela implique un ensemble de tests unitaires qui, une fois tous réalisés avec succès, permet au test d’être validé.

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre la fonctionnalité testée |
| La liste des tests unitaires utilisés | Savoir quelle valeur est testé, les conditions à succès. |

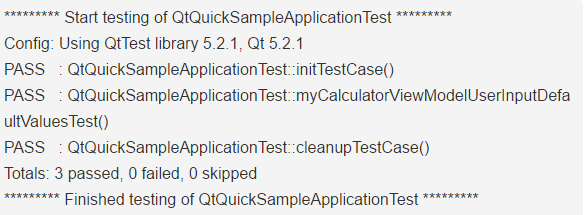
#### Tests Unitaire

Un test unitaire est une simple fonction C++, qui fait appel à la librairie QTestLib, pour valider la valeur de retour de la fonction testé.

|  |  |
| --- | --- |
| *Chaque test comportera…* | *Afin de…* |
| Une courte description présentant le test | Connaitre l’élément testé |
| Les données utilisées en temps qu’entré pour le test | Savoir quelle valeur sont testés et par substitution, lesquels ne le sont pas |
| Les conditions à succès | Connaitre quelle et comment est la sortie du test |
| La durée approximative du test | Connaitre le temps nécessaire à son exécution normale. |

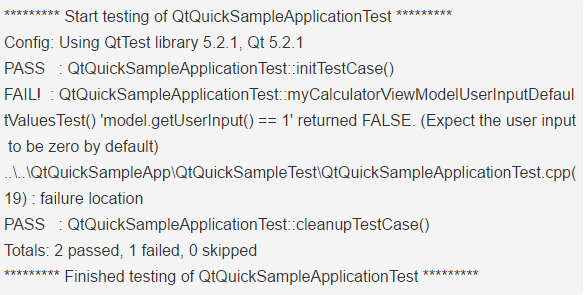
### Rapport d’exécution de tests

Les rapports d’exécution de tests apparaissent dans une console lorsqu’on lance le test. Néanmoins, la librairie QTestLib fournit la possibilité d’exporter ces résultats au format XML.



Sortie console d’un test avec succès

En cas d’erreur, la fonction ayant échoué est mentionné par un FAIL ! et un message correspondant à l’erreur est indiqué ainsi que le fichier et la ligne à laquelle l’erreur s’est produit.



Sortie console d’un test avec échec

### Rapport sur le suivi global de la qualité

## Site web

### Cas de tests

#### Tests Unitaires

Les tests unitaires seront exécutés par de simple fonction PHP qui vont vérifier la valeur de retour d’une fonction.

Chaque test unitaire sera regroupé par fonctionnalité dans une même méthode qui représentera le test pour une seule fonctionnalité.

#### Tests Fonctionnels

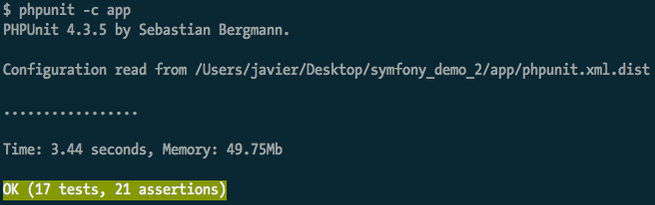
Le Framework Symfony est utilisé pour le développement du site web les différents cas de test fonctionnels seront regroupé par module, i.e. une classe de test pour la parties gestion des utilisateurs, une autre pour la gestion des partitions…

Chaque classe de test sera regroupée dans le bundle correspondant sous le répertoire /Test du bundle.

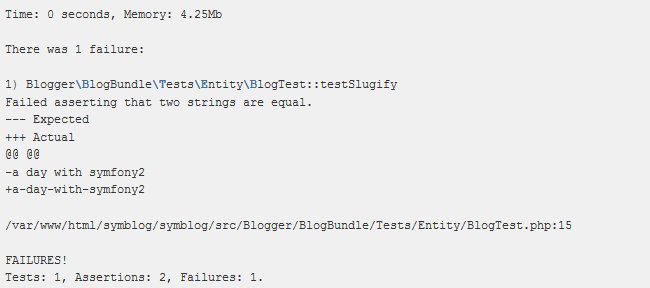
### Rapport d’exécution de tests

Tous les rapports d’exécution de tests seront fournis dans des fichiers au format texte.

Dans le cas où il n’y a pas d’erreur le nombre de test passé s’affiche avec le message OK.



Si un test échoue un message d’échec apparait avec la ligne et la classe dans lesquels le test a échoué.



### Rapport sur le suivi global de la qualité

Avant chaque mise à jour du projet sur le SVN il est impératif que le projet passe tous les tests de fonctionnalités cela veut dire qu’il ne doit y avoir aucun échec lors du lancement des commandes des tests sur les fonctionnalités.