



PLAN DE TEST

Sommaire

Introduction	2
I. Tests Unitaires	2
II. Tests Acceptances/Bout en Bout.....	3
III. Tests de Securite	3
IV. Tests de Performance.....	4

Introduction

Au sein de ce document, nous allons présenter notre stratégie de test.

Tout d'abord, même si l'idée de la méthodologie de développement TDD a été évoquée, nous avons préféré nous baser sur une méthode de développement basé sur les fonctionnalités (FDD).

Cela ne doit pas nous empêcher de nous concentrer sur la création des tests. Pour cela, nous utiliserons plusieurs types de tests différents, chacun ayant pour fonction de valider un certain aspect du produit fini, cela tout du long du processus de développement.

Les tests sont aussi essentiels afin de permettre le travail d'équipe tout en conservant la cohérence du code et permettant aux différents développeurs de s'assurer de ne rien briser au sein de l'application.

Pour l'automatisation de ces tests, nous utiliserons possiblement Monkey Runner, qui est pratique afin de tester l'application sur les différentes versions d'Android, il peut aussi être utile plus tard afin de simuler un input sur l'application, rendant possible une simulation de test utilisateur.

I. Tests Unitaires

Au vu de notre approche fonctionnelle, nos tests unitaires répondront à chaque fonction et devront valider son fonctionnement, Cela en testant ladite fonction à l'aide d'un résultat prévisible correct.

Ces tests unitaires seront regroupés et exécutés de façon automatique, procurant un premier dépistage d'erreurs ou de bugs, aidant à la cohérence du développement et à la stabilité de l'application et plus particulièrement de son code.

Ces tests sont simples à écrire et exécuter, ils sont néanmoins vitaux pour le développement.

II. Tests Acceptances/Bout en Bout

Comme mentionne au cours de l'introduction, l'utilisation de Monkey Runner pourrait grandement nous faciliter la tâche en vue des tests d'acceptation, en effet, ils nous permettraient d'exécuter directement toute la suite de tests.

Nous rapprochons les tests d'acceptations des tests de bout en bout car il nous paraît ici pertinent, au vu du périmètre de l'application, de réunir les deux au sein du même objectif.

En effet, le test de bout en bout reprendra les différents tests d'acceptation réussis de façon commune, et cela afin de valider et d'accomplir tout notre parcours utilisateur.

Ces tests sont pensés pour valider la possibilité d'accomplir des users stories, ils interviennent après les tests unitaires, plus tard dans le développement.

III. Tests de Sécurité

Nous avons pensé au fait d'utiliser Firebase, ce qui nous permettra de nous assurer grandement de l'aspect sécuritaire du projet. Les tests devront être effectués, mais seront grandement simplifiés par le fait que nous ferons appel à une entité externe pour le processus d'authentification et de stockage.

Ces tests sont à effectuer à plusieurs moments dans le processus de développement, particulièrement lorsqu'une nouvelle fonctionnalité est implémentée qui touche à la gestion des données.

IV. Tests de Performance

Pour les tests des performances, il nous faudra dans un premier temps fixé des objectifs, mais aussi des points de référence ou facteurs de réussite permettant de mesurer les performances de notre application.

En effet, sans référentiel, impossible d'évaluer correctement les performances de notre application. A première vue, dans notre application, il s'agira surtout de la gestion de la charge, du nombre d'utilisateurs se connectant simultanément ou du nombre d'écritures à l'intérieur de la base de données.

Ainsi, en évaluant correctement nos attentes, nous serons à même d'optimiser l'application et ses performances en mesurant son degré de réactivité, sa capacité et sa sécurité, car bien souvent la sécurité est aussi liée aux performances (Déni de Service, etc).