

Qualité Logiciels Racine Carrées

Version <1.0>

2023-03-26

Loïc DUMAY

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Historique des modifications du document

Date	Version	Description	Auteur
2023-03-02	1.0	Création de la documentation de test	Loïc Dumay
2023-03-26	1.0	Complétion de la documentation de test	Loïc Dumay

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Table des matières

1.	Introduction	4
1.1	Objectif du document	4
1.2	Portée du document	4
1.3	Historique	4
1.4	Définitions, acronymes et abréviations	6
1.5	Identification des artefacts	7
2.	Exigences à tester	8
3.	Stratégie de tests	8
3.1	Types de tests	8
3.1.1	Tests fonctionnels	8
3.1.2	Tests d'interface utilisateur	8
3.1.3	Tests de données et d'intégrité de base de données	8
3.1.4	Profilage de performance	8
3.1.5	Tests de configuration	8
3.1.6	Tests d'installation	8
3.2	Outils	8
4.	Ressources	8
4.1	Travailleurs	9
4.2	Système	9
5.	Jalons du projet	10
6.	Biens livrables	10
6.1	Modèle de test	10
6.2	Journaux de test	10
6.3	Rapports d'anomalies	10
7.	Glossaire	11
7.1	UML	11
7.2	IHM	11
7.3	IDE	11
7.4	OS	11
7.5	Kit de développement	11

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Plan de tests

1. Introduction

Ce plan de test permet de diriger tous les différents tests qui ont, sont et seront effectués sur le projet Qualité Logiciels Racine Carrées en intégrant les rapports de tests et les fiches de test passé, en cours et futur.

1.1 Objectif du document

Le but de ce plan de test est de produire une application de qualité.

1.2 Portée du document

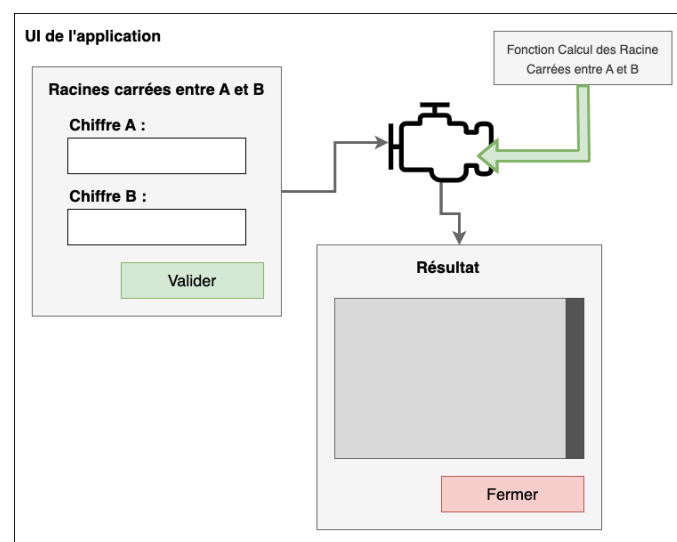
La portée de ce document est de mettre en place et recenser les étapes d'évaluation, les tests unitaires, les tests d'intégration, les tests système et les types de tests, les tests fonctionnels et les tests de performance, les risques et contingences, contraintes, etc.

Pour ce petit projet, seuls les tests fonctionnels sont traités.

1.3 Historique

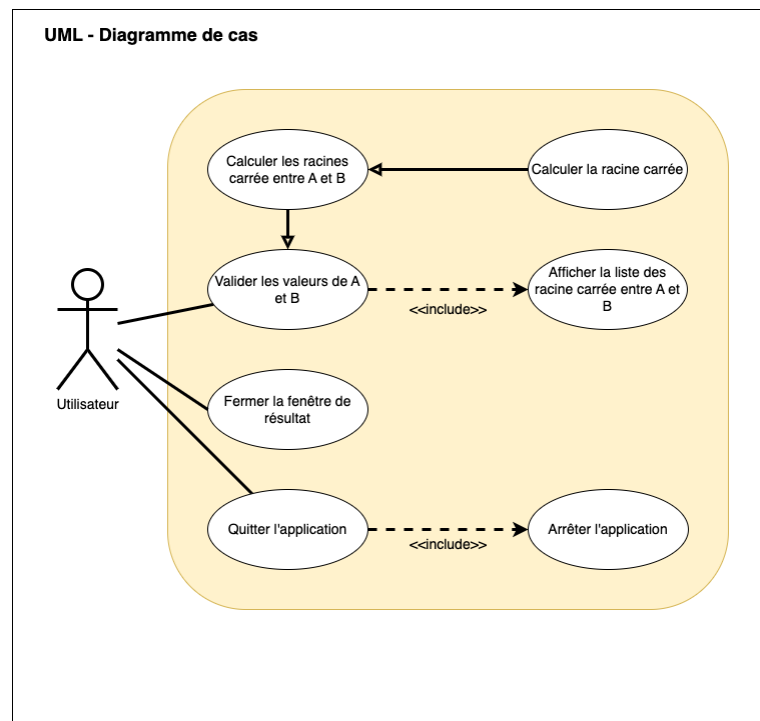
L'idée principale du projet est la conception d'une petite application sur **les racines carrées**, permettant de calculer et afficher toutes les racines carrées entre deux nombres A et B.

Il a donc été évident de créer un prototype d'interface pour guider le développement de celle-ci.

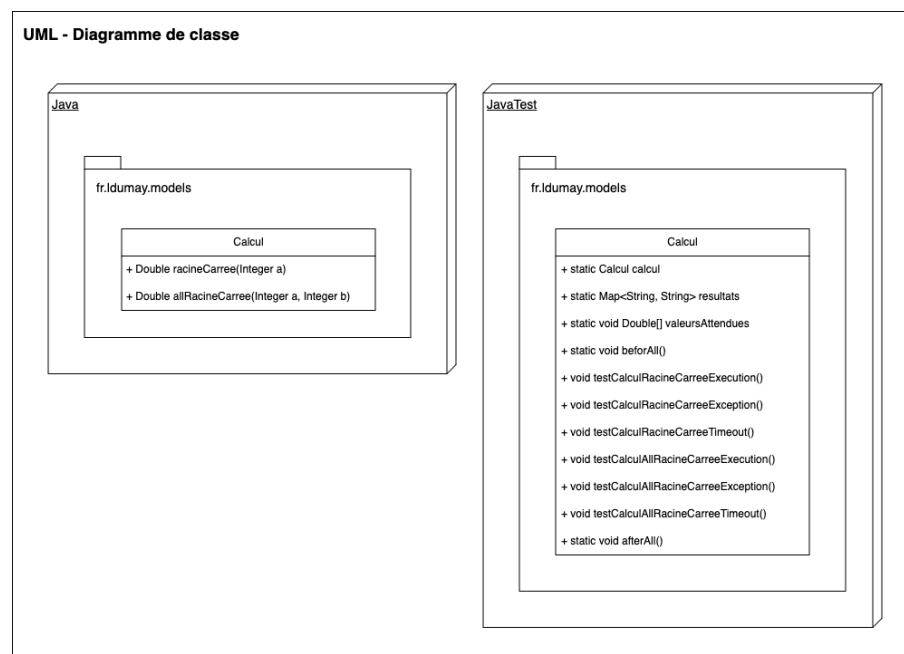


Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Grâce à ce prototype, il a été possible d'en déduire un diagramme UML de cas d'utilisation.



Ce dernier a aussi permis de monter un diagramme UML de classe.



Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Ce n'est qu'à la suite de ces trois schémas de conception que l'application a pu être développée.

Les racines carrées entre A et B

Les racines carrée comprises entre A et B

Nombre entier A

1

Nombre entier B

10

Valider A et B

Résultats

Les racines carrées entre A=1 et B=10 sont :

- > racine carrée de 1 = 1.0
- > racine carrée de 2 = 1.4142135623730951
- > racine carrée de 3 = 1.7320508075688772
- > racine carrée de 4 = 2.0
- > racine carrée de 5 = 2.23606797749979
- > racine carrée de 6 = 2.449489742783178
- > racine carrée de 7 = 2.6457513110645907
- > racine carrée de 8 = 2.8284271247461903
- > racine carrée de 9 = 3.0
- > racine carrée de 10 = 3.1622776601683795

NB : Le projet ne possède pas de version compilée exécutable. Il est donc nécessaire de le recompiler avec une IDE pour le faire fonctionner et effectuer les tests.

1.4 Définitions, acronymes et abréviations

Dans le cas où des termes ne seraient pas compréhensibles, un glossaire du projet est disponible à la fin de ce document.

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

1.5 Identification des artéfacts

Le tableau suivant identifie les artéfacts et leur disponibilité pour évaluation :

Document (version / date)	Disponibilité	Ressource	Notes
<i>Spécification d'exigences logicielles</i>	✓ Oui	Dépôt GitHub	Disponible sur le README principal du dépôt git du projet
<i>Spécification de l'interface utilisateur</i>	✓ Oui	Image	Image dans le Plan de test et le dépôt GitHub
<i>Spécifications des cas d'utilisation</i>	✓ Oui	Image	Image dans le Plan de test et le dépôt GitHub
<i>Spécifications des classes</i>	✓ Oui	Image	Image dans le Plan de test et le dépôt GitHub
<i>Réalisation de l'application</i>	✓ Oui	Dépôt GitHub	Le code de l'application est disponible dans le dépôt GitHub
<i>Réalisations des tests</i>	✓ Oui	Documentations de test	Le code et les rapports des tests fonctionnels est disponible dans le dépôt GitHub

Lien du dépôt GitHub du projet : https://github.com/ldumay/esiee_2023_qualite_logiciels

NB :

- Les prérequis de l'application sont disponibles dans le README principal du dépôt.
- La documentation du projet est disponible dans [le dossier /docs de l'application 3](#).

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

2. Exigences à tester

Pour ce plan de test, il a été décidé de tester toutes les méthodes de la classe Calcul de l'application.

3. Stratégie de tests

3.1 Types de tests

3.1.1 Tests fonctionnels

Tableau récapitulatif des tests fonctionnels est disponible dans le fichier PDF : **Rapport des tests FON.**

3.1.2 Tests d'interface utilisateur

Comme il a été retenu de tester toutes les méthodes de la classe Calcul de l'application, le plan de test ne prévoit pas de tests IHM. Donc, il n'y a pas de tableau récapitulatif des tests IHM.

3.1.3 Tests de données et d'intégrité de base de données

Il n'y a pas eu de tests de données et d'intégrité de base de données pour le développement de ce petit projet.

3.1.4 Profilage de performance

Il n'y a pas eu de profilage de performance pour le développement de ce petit projet.

3.1.5 Tests de configuration

Il n'y a pas eu de test de configuration pour le développement de ce petit projet.

3.1.6 Tests d'installation

Il n'y a pas eu de test d'installation pour le développement de ce petit projet.

3.2 Outils

Il n'y a pas eu d'AGILE SCRUM pour le développement de ce petit projet.

4. Ressources

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

Il n'y a pas eu d'AGILE SCRUM pour le développement de ce petit projet.

4.1 Travailleurs

Il n'y a pas eu d'AGILE SCRUM pour le développement de ce petit projet.

4.2 Système

Le tableau suivant identifie les ressources du système nécessaire à l'exécution du plan de tests.

Ressources du système	
Ressource	Nom – Type
Poste de développement et de tests	
• Cross-palteforme	OS
• IntelliJ IDEA 2022.3.3	IDE
• Java JDK 17 ou ultérieur	Kit de développement

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

5. Jalons du projet

Il n'y a pas eu d'AGILE SCRUM pour le développement de ce petit projet.

6. Biens livrables

Une application et une liste de test a été effectué pour **ce petit projet**.

6.1 Modèle de test

Un rapport et des fiches de test ont été réalisé :

- Le rapport de test :
 - o PDF : Tests fonctionnels de la classe **Calcul** pour l'application **qualite_logiciel_app_3** – nommé : **Rapport des tests FON**
- Les fiches de test :
 - o PDF : **Fiche de tests N°1 de type FON** (*disponible*)
 - o PDF : **Fiche de tests N°2 de type FON** (*non disponible*)
 - o PDF : **Fiche de tests N°3 de type FON** (*disponible*)
 - o PDF : **Fiche de tests N°4 de type FON** (*non disponible*)
 - o PDF : **Fiche de tests N°5 de type FON** (*non disponible*)
 - o PDF : **Fiche de tests N°6 de type FON** (*non disponible*)

6.2 Journaux de test

Il n'y a pas de journal de test pour le développement de ce petit projet.

6.3 Rapports d'anomalies

Il n'y a pas de rapport d'anomalies pour le développement de ce petit projet.

Qualité Logiciels Racine Carrées	Version : <1.0>
Plan de tests	Date : 2023-03-26

7. Glossaire

7.1 UML

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Source : [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

7.2 IHM

L'interaction homme-machine, appelé IHM, s'intéresse à la conception et au développement de systèmes interactifs en prenant en compte ses impacts sociétaux et éthiques.

Source : https://fr.wikipedia.org/wiki/Interactions_homme-machine

7.3 IDE

En programmation informatique, un environnement de développement est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels¹. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction². Certains environnements sont dédiés à un langage de programmation en particulier.

Source : https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement

7.4 OS

En informatique, un système d'exploitation (souvent appelé OS — de l'anglais operating system) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs.

Source : https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation

7.5 Kit de développement

Un kit de développement logiciel, aussi appelé trousse de développement logiciel, est un ensemble d'outils logiciels destinés aux développeurs, facilitant le développement d'un logiciel sur une plateforme donnée (par exemple, iOS, Android, BlackBerry 10, Symbian, Bada, Linux, OS X, Microsoft Windows).

Source : https://fr.wikipedia.org/wiki/Kit_de_d%C3%A9veloppement