



Université de Yaoundé I
Faculté des sciences

ICT304: SOFTWARE TESTING

PLAN DE TEST

Réalisé Par :

- TIENTCHEU TAKOU Maurice Donald 21Q2439
- HAPPI NOUWE Morel Alban 21Q2523
- NDJOCK Liam Jordan 21Q2349
- ESSEME Audrey 21L2309
- TALLA Donald Rodrigue 21U2344

Encadré Par :

- Dr. KIMBI

Table des matières

1. Scope

1.1. Fonctionnalités principales

1.2. Modules à tester

2. Schedule

2.1. Phase 1 : Tests unitaires

2.2. Phase 2 : Tests d'intégration

2.3. Phase 3 : Tests fonctionnels end-to-end

3. Resource Allocation

3.1. Équipe de test

3.2. Outils de test

4. Tools

4.1. Vitest pour les tests unitaires

4.2. Sinon.js pour les tests d'intégration

4.3. Puppeteer pour les tests end-to-end

4.4. Istanbul pour la couverture de code

4.5. ESLint pour l'analyse statique

5. Risk Management

5.1. Risques techniques

5.2. Risques organisationnels

6. Exit Criteria

6.1. Couverture de code

6.2. Réussite des tests

6.3. Nombre de défauts critiques

7. Conclusion

Scope

Fonctionnalités principales de l'application de restaurant local :

- Gestion du panier d'achat (module "basket")
- Gestion des promotions (module "promotions")
- Gestion des utilisateurs (module "users")
- Gestion des événements (module "events")
- Gestion des erreurs (module "error-handling")

Le plan de test couvre les tests unitaires, d'intégration et fonctionnels de ces différentes fonctionnalités.

Schedule

Phase 1 (2 semaines) : Tests unitaires des modules individuels

- Basket Test Cases
- Promotions Test Cases
- Users Test Cases
- Events Test Cases
- Error Handling Test Cases

Phase 2 (1 semaine) : Tests d'intégration entre les modules

- Integration Test Cases

Phase 3 (1 semaine) : Tests fonctionnels end-to-end

- End-to-End Test Cases

Resource Allocation

Équipe de test :

Nom	Rôle	Taches
TIENTCHEU DONALD	Analyste test senior	Supervise Basket testcase , error-handling and event
ESSEME AUDREY	Testeur junior	Basket-Test case
NDJOCK JORDAN	Testeur junior	Account Testcase

Nom	Rôle	Taches
HAPPI NOUWE MOREL ALBAN	Testeur junior	Event Testcase
TALLA DONALD	Testeur junior	User Testcase

Outils de test :

- Vitest pour les tests unitaires
- Js pour les tests d'intégration
- Pipeter pour les tests end-to-end

Environnement	Outils	Configuration
Développement	- Visual Studio Code version 1.78.0- Node.js version 20.11.0- Dépendances Node.js à jour	- Extensions VS Code installées : - Vitest - ESLint - Prettier- Projet configuré avec Vitest pour les tests unitaires
Intégration	- Visual Studio Code version 1.78.0- Node.js version 20.11.0- Dépendances Node.js à jour	- Extensions VS Code installées : - Vitest - Sinon .js - ESLint - Prettier - Projet configuré avec Vitest pour les tests unitaires et Sinon.js pour les tests d'intégration - Configuration matérielle et logicielle identique à l'environnement de production- Données de test représentatives
Production	- Visual Studio Code version 1.78.0- Node.js version 20.11.0- Dépendances Node.js à jour	- Extensions VS Code installées : - Vitest - Sinon.js - Puppeteer - ESLint - Prettier

Environnement	Outils	Configuration
		<ul style="list-style-type: none"> - Projet configuré avec Vitest pour les tests unitaires, Sinon .js pour les tests d'intégration et Puppeteer pour les tests end-to-end - Configuration matérielle et logicielle identique à l'environnement de test- Données réelles des utilisateurs

Tools

- Vitest pour les tests unitaires
- Sinon.js pour les tests d'intégration
- Puppeteer pour les tests end-to-end
- Istanbul pour la mesure de la couverture de code
- ESLint pour l'analyse statique du code

Risk Management

Risques techniques :

- Complexité des interactions entre modules
- Performances de l'application
- Sécurité des données utilisateur

Risques organisationnels :

- Disponibilité des ressources de test

Respect du planning

- Couverture de code supérieure à 90% pour les tests unitaires
- Réussite de tous les tests d'intégration
- Réussite de tous les scénarios de test end-to-end
- Nombre de défauts critiques ouverts inférieur à 5

Conclusion

Ce test plan détaillé fournit une approche structurée pour tester de manière approfondie l'application de restaurant local, en couvrant les différentes phases de test et en tenant compte des aspects clés tels que les ressources, l'environnement, la gestion des défauts et des risques. Il devra être affiné au fur et à mesure de l'avancement du projet.

