

Compte Rendu n°3 du Génie Logiciel

Nom : DERMOUCHE

Prénom : Mohammed Ryad

Groupe : TD3

1)-Première partie :

1)- Les activités effectuées et non effectuées :

- **Définition des objectifs** : Les objectifs du projet ont été définis pour résoudre le problème de coloration des graphes à l'aide de la logique propositionnelle.
- **Analyse des exigences** : L'analyse des exigences a été insuffisante, les besoins n'ont pas été clairement définis dès le début du projet.
- **Analyse de faisabilité** : Aucune analyse de faisabilité formelle n'a été réalisée, ce qui a conduit à des problèmes d'intégration et de compatibilité.
- **Spécifications des exigences** : Les exigences fonctionnelles et non fonctionnelles n'ont pas été correctement spécifiées, ce qui a complexifié le développement.
- **Conception** : Une conception initiale du projet a été réalisée, mais elle s'est avérée inadéquate pour l'intégration des nouvelles fonctionnalités.
- **Implémentation** : Le code a été implémenté en Python, avec des bibliothèques utilisées pour visualiser les graphes.
- **Tests unitaires** : Des tests unitaires ont été partiellement envisagés, mais ils n'ont pas couvert tout le code.
- **Intégration** : L'intégration a été réalisée, mais elle a rencontré des difficultés en raison de problèmes de compatibilité entre les bibliothèques.
- **Validation** : Aucune validation finale complète n'a été réalisée pour s'assurer que le projet répondait aux exigences définies.
- **Déploiement** : Le déploiement n'a pas été réalisé.
- **Maintenance** : Aucune mention de la maintenance du projet n'a été faite, car on n'a pas eu le temps de revoir le projet après la soutenance.

*Raisons possibles pour l'absence de certaines activités :

- Manque de connaissances en gestion de projet
- Pression du temps et des ressources
- Focalisation sur l'aspect technique
- Manque d'expérience avec les méthodologies de développement

*Conséquences de l'absence de ces activités :

- Problèmes techniques imprévus
- Retards dans le projet
- Qualité et performance du logiciel
- Sans déploiement ni maintenance, le logiciel risque de ne pas être durable. Mauvaise satisfaction des utilisateurs

2)- Le temps personnes en heures :

Activité	Heures	Normes
Définition des objectifs	6	Conforme
Analyse des exigences	3	Sous-estimée
Analyse de faisabilité	0	Non réalisée
Spécifications des exigences	2	Sous-estimée
Conception	12	Sous-estimée
Implémentation	50	Conforme
Tests unitaires	9	Sous-estimée
Intégration	16	Légèrement sous-estimée
Validation	0	Non réalisée
Déploiement	0	Non réalisée
Maintenance	0	Non réalisée

3)- Le modèle de cycle qu'on a utilisé : **modèle en cascade (Waterfall)****Pourquoi ce modèle a-t-il été utilisé ?**

- **Approche simple** : Il est facile à gérer et à suivre, surtout pour les petits projets avec des exigences stables.
- **Manque de flexibilité** : Notre projet n'a pas connu de modifications fréquentes des exigences, ce qui correspond à l'approche du modèle en cascade.
- **Absence d'itérations** : Il n'y avait pas de retours en arrière ou de révisions continues, ce qui est typique du modèle en cascade.

Aurait-il été possible d'en utiliser un autre ? : **Oui , Modèle en V**

Pourquoi c'est adapté à notre projet ? :

- Il aurait permis une validation rigoureuse à chaque étape.
- Il aurait réduit les erreurs liées aux graphes complexes, limitant les sessions de débogage intensives.
- Chaque phase de développement étant liée à une phase de test, les problèmes auraient été détectés plus tôt.
- Cela aurait évité les bugs imprévus pendant l'intégration finale.

2)-Deuxième partie :

Nous avons choisi de travailler sur le premier projet, et notre équipe est composée de :

- OUBERKA Mohamed
- DERMOUCHE Mohammed Ryad
- BOUCHELLAL Imad-Eddine

Exigences fonctionnelles :

- Le logiciel doit permettre au clients de passer facilement leurs commandes
- Le logiciel doit permettre au clients de choisir la langue de commandes
- Le logiciel doit permettre au clients de suivre l'état d'avancement de sa commande
- Le logiciel doit permettre au client fidèle de suivre leurs points accumulés à chaque dépenses et les utiliser pour appliquer des réductions
- Le logiciel doit afficher le prix, la photo et les ingrédients de chaque articles ainsi que des remarques concernant les allergies
- Le logiciel doit permettre au client de créer un compte de fidélité
- Le logiciel doit permettre aux serveurs de s'authentifier
- Le logiciel doit permettre aux manager de s'authentifier
- Le logiciel doit permettre au personnel de mentionner toute arrivées et utilisation du stock en terme de quantité, articles, emplacements dans le stock, prix
- Le logiciel doit permettre le suivi de chaque serveurs en temps de travail, nombre de tables servi
- Le logiciel doit permettre de calculer le chiffre d'affaires, les salaires des employées, les dépenses des matière première
- Le logiciel doit être reliée à la borne de paiement sans contact

Exigences non-fonctionnelles :

- Le logiciel doit être une application mobile/tablette compatible avec l'OS android
- Le logiciel doit être fiable
- Le logiciel doit avoir une disponibilité de 99,5% du temps
- Le logiciel doit être scalable pour des future mise à jour
- Le logiciel doit respecter la conformité au RGPD (Règlement Général sur la Protection des Données)
- le temps de réponse : le temps de réalisation d'une opération sur l'application doit être optimale (pas plus de 5s).
- La capacité à monter en charge quand la quantité d'utilisateurs simultanés augmente.
- le temps de la prise de commande il faut qu'il soit instantané, et une fois que le client a pris sa commande, la cuisine la reçoit directement.
- l'apparition des menus et des produits : Le chargement des menus et des options doit être rapide (idéalement inférieur à 1 seconde).

- En cas de panne ou de perte de connexion, l'application doit pouvoir se reconnecter rapidement.

Exigences Techniques :

- Le logiciel doit être codé en Flutter
- Le logiciel doit être hôte chez AWS

Exigences de sécurités :**1)-Protection des comptes client :**

- **Authentification :**
 - Chaque client, y compris les clients fidèles, doit disposer d'un compte sécurisé accessible via un mot de passe fort ou une authentification OAuth (Google, Facebook, etc.).
 - Un système de réinitialisation de mot de passe sécurisé doit être en place.
 - L'authentification à deux facteurs (2FA) doit être proposée pour renforcer la sécurité.

2)- Sécurité du programme de fidélité :

- **Gestion des points de fidélité :**
 - Les clients doivent accumuler automatiquement des points à chaque commande et consulter leur solde dans un espace sécurisé.
 - Les points peuvent être échangés contre des réductions sur des commandes futures.
- **Protection des données et anti-fraude :**
 - Les données liées aux transactions et au programme de fidélité doivent être chiffrées.
 - Le système doit surveiller toute activité suspecte (tentatives de fraude, manipulation de points) et alerter les clients en cas de comportement inhabituel.