

Rapport du Quatrième livrable

Table des matières

Arborescence du quatrième livrable :	1
Choix du livrable.....	3
Quatrième livrable.....	5

Arborescence du quatrième livrable :

1. Analyse_des_besoins :

○ Cahier_des_charges

- Un (quatrième) cahier des charges minimal
- Un cahier des charges cumulé

○ Organisation :

- 1.WBS
- 2.Les matrices d'antériorité
- 3.GANTT

○ Recueil_des_besoins

- Recueil_des_besoins

2. Spécification :

○ Maquette_logo

- Maquette :
 - Maquettes
- Logo :
 - Logos
- **Cas d'utilisation**
 - Cas_d_utilisation
 - Scenarii
- Etude des technologies web

3. Conception :

- **Diagramme**
 - Diagramme_composant-connecteur
 - Diagramme_deploiement
- Implementation_monolithique_technologies
- **Conception_architectural_PagePhp**
 - Conception_architectural_Php
- **Conception_détaillée**
 - Conception_Page

4. Programmation :

- Simfast-Php
- **Logos**
 - Logo
- **Videos**
 - Video_presentation
- **Css**
 - Css.css

5. Test :

- **Dossier_de_test**
 - Document_test_validation
 - Document_test_intégration
-

Choix du livrable

Choix du quatrième livrable

Nous avons fait un changement dans nos livrables : Ce livrable va être sur les pages PHP de l'utilisateur mais aussi sur les pages du gestionnaire. Cependant le prochain livrable sera sur les modules.

Nous avons choisi dans ce quatrième livrable de faire un processus de développement pour finaliser la partie php du projet : la création des pages en php. Ce quatrième livrable est tourné vers le choix de création des pages en php avec leur implémentation sur le serveur du Rasperrybi.

Choix de l'arborescence du quatrième livrable

Nous avons choisi cette méthode de livrable pour la raison suivante, afin de mener à bien un projet informatique de cette envergure, nous devons rassembler les activités du génie logiciel pour respecter les critères de qualités. Nous avons donc proposé de faire plusieurs livrables avec une caractéristique : les dossiers seront divisés en selon les activités du génie logiciel : Analyse des besoins, Spécification, Conception, Programme, Test. Cela permet une meilleure clarté et aussi de suivre un ordre linéaire pour la validation de notre livrable.

Choix du rendu en plusieurs livrables, Cycle de vie :

Notre équipe suit un processus de développement. Le processus de développement indique la forme dans laquelle les activités sont connectées entre elles. L'ordre dans lequel s'enchaînent les activités s'appelle le cycle de vie du produit logiciel.

Ici ce premier livrable représente un cycle de vie du produit logiciel. Le produit logiciel est décomposé en plusieurs cycles de vie et sera complété par de nouveaux livrables au fil du déroulement du projet.

Cette partie du rapport est rédigé après avoir fait l'activité du génie logiciel : **Analyse des besoins**. Avec cette activité nous avons pu déterminer la méthode de décomposition de nos livrables qui définiront notre cycle de vie.

Nous avons proposé un cycle de vie en Cascade. (Voir figure) C'est un diagramme d'activité, utilisé pour montrer l'enchaînement des tâches effectuées par notre équipe. Chaque livrable se basera sur cette méthode. Lorsque ces activités seront terminées on passera à la suivant dans un nouveau livrable.

Cette méthode de cascade nous permet d'avoir un contrôle sur chaque étape. Elle permet dès que possible de pouvoir créer les tests, ainsi lors du développement, à chaque étape on réfléchit aux erreurs/problèmes que l'on rencontre. Une fois la programmation terminée nous exécutons les tests de validation qui vont venir valider les besoins fonctionnelles définis dans le cahier des charges puis construire à partir des maquettes un plan répondant aux exigences du client. Les tests d'intégrations vont venir valider la relation des composants (pages html). Nous sommes donc dans un cycle linéaire.

Quatrième livrable

Analyse des besoins

Le (quatrième) cahier des charges minimal :

Nous proposons un quatrième cahier des charges qui n'est pas complet et sera complété au fil des livrables. Nous pouvons le retrouver dans le dossier document sur le gitlab du projet

Nous proposons un cahier des charges cumulés, c'est-à-dire tout ce qui a été vu dans le cahier des charges jusqu'à ce livrable.

Organisation :

Dans le dossier organisation disponible sur le premier livrable de notre dépôt GitLab, nous pouvons retrouver le diagramme WBS des activités à réaliser, les matrices d'antériorité et enfin le diagramme de GANTT pour pouvoir proposer un cycle de vie.

Recueil des besoins :

Nous listons et reformulons les exigences définies par le client de manière claires et concises tout en restant réaliste. Nous pouvons retrouver le recueil des besoins dans le dossier Document du dépôt GitLab.

Spécification

Maquette_logo :

La maquette que le client a choisi et les logos se trouve dans le dossier « Maquette_logo » situé dans le dossier « Document ». Pour cette partie nous avons réalisé nos maquettes sur le même principe de couleur et de visibilité. Nous avons utilisé un outil se nommant Figma.

Le client a choisi le logo SIMFAST en bleu en cohérence avec les maquettes.

Cette partie a permis de créer les tests de validation avec la partie du cahier des charges afin de satisfaire les demandes du client. Les tests de validation sont validés si la visibilité des pages lors de l'exécution des pages codes sont identique aux maquettes et corresponde à la demande du client

Cas d'utilisation :

Nous avons réalisé les cas d'utilisation.

Rapport scénarii :

Nous retrouvons la rédaction des scénarii que nous avons proposé

Études des technologies utilisées :

On définit toutes les technologies utilisées lors de ce projet.

Conception

Conception_architectural :

La conception architecturale permet de rendre compréhensible le modèle avec les interactions (relations) entre les pages php. Cela permet de faciliter le passage à la programmation par la représentation de ce modèle.

Dans la conception architecturale il y a la représentation des interactions entre toutes les pages que nous devons créer.

Conception_détaillée :

La conception détaillée est un détail de chaque page en termes de composant balistique de l'html. Elle permet de rendre pour chaque page php un aperçu des grandes parties du code.

Dans la conception détaillée il y a la représentation des grandes parties « balises » html de toutes les pages html que nous devons créer.

Diagramme :

Le diagramme composant-connecteur nous permet d'avoir un aperçu des composants et des liens qu'ils vont avoir à la fin du projet, cela nous guidera pour l'interaction des composants.

Le Diagramme de déploiement permet de répartir les composants de la vue où l'utilisateur va observer devant son écran et de séparer au contrôle avec tous les composants et code qui permettra de rendre une page web dynamique.

Implémentation monolithique technologique

Est une représentation des interactions entre les différentes technologies que nous devons utiliser.

Programmation

Page_php :

Nous retrouvons toutes les pages php que nous devons créer, et réaliser à partir des maquettes. Nous avons mis les interactions entre les pages à l'aide du php. Avec le php nous avons pu faire la connexion login et mot de passe, ainsi que l'inscription et l'interaction avec la base de données que nous avons créée dans le livrable antérieur (3^{ème})

Nous avons ajouté les vidéos et images pour les pages PHP.

Test

Test d'acceptation :

C'est un type de test qui permet de vérifier si toutes les exigences client, décrites dans le cahier des charges correspondent aux attentes du produit défini dans chaque livrable. Elle est aussi décrite à partir de l'activité de spécification grâce à la réalisation des maquettes.

Nous retrouvons les tests d'acceptation réalisés à partir des attentes des clients et des maquettes, et valider après la programmation si cette dernière correspond bien.

Test d'intégration :

C'est un type de test qui permet de vérifier si les exigences du client sur la navigabilité entre les composants correspondent bien. Les tests sont réalisés à partir du cahier des charges et approfondi dans la conception architecturale dans la partie conception.

Nous retrouvons les tests d'intégration réalisé à partir des attentes du clients et de la conception architecturale. La conception architecturale permet de voir si une page appelle l'autre. Ces tests sont validés après la programmation de toutes les pages html et si toutes les interactions entre les pages correspondent bien.

Test unitaire :

C'est un type de test qui permet de vérifier si les attendues sont bon. Ils sont réalisés à partir des pages PHP et on entre des valeurs pour vérifier si ça s'implémente dans la base de données. Dans l'ancien livrable, nous avons proposé ces tests mais pour le code que nous avons proposé pour les pages php. Dans ce livrable nous avons implémenté le code fait dans le livrable ultérieur et nous vérifions sur nos pages si cela marche.