

Plan de développement

*Développement d'un outil pour
l'apprentissage des positions des fœtus au
moment de l'accouchement.*

Plan de développement

Développement d'un outil pour l'apprentissage des positions des fœtus au moment de l'accouchement.

Les informations d'identification du document

Référence du document :	Cahier des Charges
Version du document :	2
Date du document :	19/01/2024
Auteur(s) :	Miguelangel Espinal Coemgen Genevey

Les éléments de vérification du document

Validé par :	Lucile Vadcard
Validé le :	//
Soumis le :	//
Type de diffusion :	Document électronique (.pdf)
Confidentialité :	

Les éléments d'authentification

Date / Signature :

Date / Signature :

Sommaire

Sommaire.....	3
1. Introduction (ou préambule).....	3
1.1. Objectifs et méthodes.....	3
1.2. Documents de référence.....	3
2. Concepts de base.....	4
3. Organisation.....	4
3.1. Décomposition en tâches.....	4
1. Planification.....	5
2. Cycle de vie.....	5
3. Méthodes et outils.....	6
4. Documentation.....	6
4.1. Présentation.....	6
4.2. Standard et outils.....	8
5. Qualité.....	9
6. Annexes.....	9
7. Glossaire.....	9
8. Références.....	9

1. Introduction (ou préambule)

Le Plan de Développement a pour objectif de de détailler les étapes de conception de de ce projet.

1.1.Objectifs et méthodes

Présenter le plan de développement du logiciel dans son ensemble avec les tâches spécifiques et sa distribution.

1.2.Documents de référence

- Cahier des Charges
- Cahier de Recette

2. Concepts de base

3. Organisation

On présente les tâches spécifiques qui vont rendre possible la création du projet.

3.1.Décomposition en tâches

1. Création de l'espace de travail avec framework et git.
2. Création et remplissage de base de données.
 - 2.1. Recueillir les données.
 - 2.2. Modélisation et création de la base de données.
 - 2.3. Remplir la base de données.
3. Ecriture du code.
 - 3.1. Programmation d'une interface graphique accorde aux maquettes présentées et aux besoins du client côté étudiant.
 - 3.1.1. Montrer la page d'accueil.
 - 3.1.2. Montrer la page de login ou inscription.
 - 3.1.3. Montrer la page type pour choisir un exercice.
 - 3.1.4. Montrer la page type pour une étape d'un exercice.
 - 3.1.5. Montrer la page type de correction d'une étape d'un exercice.
 - 3.1.6. Montrer la page de résultat d'un exercice.

- 3.2. Programmation d'une interface graphique accorde aux maquettes présentées et aux besoins du client côté enseignant.
 - 3.2.1. Montrer la page d'accueil.
 - 3.2.2. Montrer la page de login ou inscription.
 - 3.2.3. Montrer la page avec les cohortes avec des données disponibles.
 - 3.2.4. Montrer la page de résultat d'une cohorte.
- 3.3. Programmation des fonctions et méthodes pour:
 - 3.3.1. Le chargement des données.
 - 3.3.2. Lancer un exercice type et faire la boucle pour passer dans toutes les étapes de l'exercice.
 - 3.3.3. Pouvoir tourner les images et/ou modèles en 3D dans l'appli pour que l'étudiant puisse donner une réponse.
 - 3.3.4. Vérifier les réponses données par l'utilisateur avec les réponses correctes et fournir la page de vérification correctement remplie.
 - 3.3.5. Enregistrer les données d'un exercice.

3.4. Structure des équipes

L'équipe est formée par 2 développeurs qui vont se répartir les tâches équitablement.

1. Planification

Etape 1 : Création de l'espace de travail

Date prévisionnelle : 01/02/2024

Etape 2 : Création des bases de données

Date prévisionnelle : 01/02/2024

Etape 3.1 : Programmation de l'interface étudiant

Période de temps prévisionnelle : 01/02/2024 - 31/05/2024

Etape 3.2 : Programmation d'une interface enseignant

Période de temps prévisionnelle : 01/02/2024 - 31/05/2024

Etape 3.3 : Programmation des fonctions et méthodes

Période de temps prévisionnelle : 01/05/2024 - 07/06/2024

Etape 4 : Phase de tests

Période de temps prévisionnelle : 27/05/2024 - 07/06/2024

Etape 5 : Soutenance Finale

Date prévisionnelle : 20/06/2024

2. Cycle de vie

Vu la taille restreint du nombre des développeurs, on va travailler avec la méthode agile comme SCRUM, bien adapté à nos besoins, où on aura des réunions et des échanges très souvent entre les deux développeurs, mais aussi, plus comme dans les méthodes agiles, on aura des réunions semanales avec le tuteur pour suivre le progrès chaque semaine et quand c'est le cas pouvoir adapter le développement.

On aura un phase plutôt dédiée à la création de l'interface graphique soit de l'utilisateur, l'admin ou l'enseignant. Une autre dédiée aux bases de données. Pour avoir aussi un phase plutôt de programmation avec la logique du système.

3. Méthodes et outils

Les langages de programmation utilisés seront Javascript, php, html, avec un framework comme React ou Vue JS.

Nos productions seront réalisées sur VSCode et seront sauvegardées sur github, on travaille aussi quelques documents sur un drive.

Nous disposerons également d'un ensemble d'images et de schémas représentant des les différentes positions du nouveau-né durant l'accouchement.

4. Documentation

4.1.Présentation

4.1.1. Architecture de l'application

Pour le côté interface utilisateur ça va commencer avec un header qui affichera à gauche le logo et à droite le bouton log in/s'enregistrer.

Ensuite un menu avec plusieurs boutons qui pourront varier en fonction de l'utilisateur de la session actuel:

- Accueil
- Exercices
- Résultats

Pour finir avec l'espace ou on va afficher le contenu de chaque page:

1. Page d'accueil et explication de fonctionnement.
2. Page des exercices disponibles à réaliser par l'étudiant.

Affiche une liste de liens vers les différents exercices proposés à l'étudiant.

3. Page pour afficher une question d'un exercice.
 - 3.1. Avec chaque étape à passer dans chaque question.

Chaque étape correspond à une représentation du nouveau-né à compléter. Quand toutes les représentations sont remplies, l'étudiant peut passer à la page suivante (page de vérification des réponses).

4. Page pour afficher le retour d'un exercice après avoir été répondu par l'étudiant.

Pour chaque question l'ensemble des réponses données par l'étudiant seront affichés ainsi que les réponses attendues.

5. Page pour afficher le résumé et le résultat final après avoir passé toutes les questions d'un exercice.
6. Page pour afficher le résumé de chaque cohorte.

Disponible uniquement pour les enseignants et leur permet de voir pour chaque cohorte quels exercices ont posé le plus de problème.

7. Page pour la création des exercices.

Disponible uniquement pour les enseignants et leur permet de créer un exercice en sélectionnant question qu'il souhaite ou en les mettant aléatoire.

Côté logique de programmation on va avoir les particularités suivantes:

1. Le chargement des données sur la page des exercices.

Ce point doit permettre de charger les paquets d'exercices, les données pour remplir les étapes d'une question et les données de la réponse correcte de chaque question.

2. L'enregistrement des réponses.

Pour permettre vérifier les réponses données par l'étudiant et pour a la fin de chaque exercice rajouter les résultats d'une étudiante a sa cohorte correspondante.

3. La vérification des réponses données.
4. Création aléatoire d'exercices.
5. Création aléatoire d'exercices mais avec difficultés et conditions.

4.1.2. Plan de test

Le plan de test sera rédigé durant la phase de test. Son développement sera similaire au plan suivant :

A) Présenter un ensemble de tests ou recettes ainsi que leur ordonnancement

B) Description des tests d'intégration. Pour chaque test :

B.1) Description du test : Décrire le test de façon externe

B.2) But du test : Décrire ce que prouve le test

B.3) Principe de réalisation : Décrire la procédure de test ainsi que les paramètres et

C) Description des tests unitaires : Pour chaque module/classe :

C.1) Mise en œuvre

C.2) Pour chaque jeu d'essai : Données d'entrées, résultats attendues, critère d'évaluation

4.1.3. Compte-Rendu

Tout au long du projet devront être rédigé un compte-rendu des réunions ainsi qu'une documentation interne détaillant pour chaque module :

A) Ses Objectifs

B) Ses Relations avec d'autres Modules

C) La Liste des Modules dont il est dépendant ainsi que ceux qui sont dépendant de lui

- D) Définir les types et attributs de l'objet correspondant au module*
- E) Lister les procédures visibles par un autre module externe*
- F) Lister les variables visibles par un autre module externe*

4.2. Standard et outils

Ces documents seront fournis au format .pdf.

5. Qualité

Définir les critères déterminant la qualité du logiciel

6. Annexes

7. Glossaire

Définit l'ensemble des termes spécialisés du document

8. Références

Indique les références bibliographiques vers d'autres documents apportant des informations complémentaires