

Cahier des charges Finale

Table des matières :

I – Introduction

II – Énoncé

III – Pré-requis

IV – Priorités

I / Introduction

À l'aide du cahier des charges que vous nous avez délivré sur le projet d'une réalisation d'une application Web, nous proposons en retour un cahier des charges minimal technique avec le vocabulaire de l'informatique. Il devra être validé par le client afin que son contenu réponde à ses besoins et exigences.

Nous retrouvons sur ce cahier des charges les exigences demandées par le client, ainsi que les contraintes qui sont imposées à l'équipe de développement. Ces exigences sont fonctionnelles ou non fonctionnelles. Le cahier des charges est exprimé de manière concise et sans ambiguïté, il sera complété au cours de la réalisation du projet.

II / Énoncé

Domaine du projet : Le système est un site web, qui permet de faire des simulations de différents calculs (regroupés dans des modules) qui dépendent de différents domaines.

On applique l'abstraction sur le domaine (Cf méthode COO).

On réalise un tableau avec OBJET | ETAT | COMPORTEMENT

Objet	État	Comportement
Application Web	Aura plateforme	Installé sur un serveur (RaspberryPi)

Plateforme	<p>a 3 types d'utilisateur a un gestionnaire à un utilisateur inscrit a un visiteur</p> <p>a au moins trois simulation dans différents domaines</p>	Pour faire simulation
Visiteur		<ul style="list-style-type: none"> • Peut accéder à la page d'accueil • Peut voir une vidéo de démonstration • Peut s'inscrire grâce au formulaire d'inscription
Utilisateur Inscrit	<ul style="list-style-type: none"> • A un login, mot de passe, email 	<ul style="list-style-type: none"> • Peut se connecter/déconnecter • Peut utiliser les outils de simulation • Peut accéder à son profil • Peut changer de mot de passe
Gestionnaire	<ul style="list-style-type: none"> • A un login, mot de passe, email 	<ul style="list-style-type: none"> • Peut se connecter/déconnecter, • Peut visualiser la liste des utilisateurs, -> supprime son historique • Peut visualiser les données statistiques • Peut supprimer un utilisateur, ~visionner les données mentionnées dans la partie utilisateur~ • Peut établir une statistique des visites et des modules utilisés par les visiteurs inscrits

Page d'accueil	<ul style="list-style-type: none"> • A un texte explicatif • A une vidéo de démonstration 	
----------------	---	--

Page d'inscription	<ul style="list-style-type: none"> • A un formulaire avec login(=identifiant), mot de passe • A un captcha 	
Page de connexion	<ul style="list-style-type: none"> • A un formulaire avec login, mot de passe • A un lien « oublie de mot de passe » 	<ul style="list-style-type: none"> • Accède à une page en construction
Plateforme de simulation	Possède des outils de simulation	
Fichier log	<ul style="list-style-type: none"> • A un login, MDP tenté, adresse IP, date 	Enregistrer sur serveur web
Outils de simulation	<p>Informatique</p> <ul style="list-style-type: none"> • a conversion binaire en hexa • a conversion hexa en binaire • a conversion octal en décimal • ... <p>Mathématiques</p> <ul style="list-style-type: none"> • a résultat de probabilité <p>Gestion a calcul d'amortissement</p>	
Application web		Installé sur un serveur (RaspberryPi 4)

RaspberryPi 4	<ul style="list-style-type: none"> • A une application web • A un serveur web (à installer) • A un serveur SGBD (Mysql) • A une base ou des bases de données 	
Données de l'historique	<ul style="list-style-type: none"> • A utilisateur et module utilisé 	
Données des statistiques	<ul style="list-style-type: none"> • A un nombre de visite • A un nombre de module utilisé 	
Données de l'utilisateur inscrit	<ul style="list-style-type: none"> • A un login, mot de passe, un email 	
Données du gestionnaire	<ul style="list-style-type: none"> • A un seul gestionnaire • A un login (gestion), mot de passe (à définir), un email 	

II.1 La base de l'application web

Nous devons faire une application web, qui comportera un menu avec des simulations de calculs. C'est-à-dire qu'elle pourra, par exemple faire des conversions de binaire à l'hexadécimal, calculer les coûts d'amortissement d'un projet...

Nous devons réaliser une page d'accueil avec un texte explicatif et une vidéo qui incitera les visiteurs (= utilisateurs non-inscrits) à s'inscrire.

Nous devons créer une page d'inscription qui comportera un formulaire où le visiteur devra créer son identifiant (ou 'login') ainsi que son mot de passe et son email. Il devra ensuite remplir un captcha qui s'assurera qu'il s'agit bien d'un être-humain. Cela finalisera la création de son compte et il pourra utiliser l'application.

Nous devons réaliser une page de connexion qui contiendra un formulaire demandant le login et le mot de passe de l'utilisateur avec son email ainsi qu'un lien d'oubli de mot de passe qui redirigera sur une page en construction. La page de connexion sera accessible à tous et donnera l'accès à l'application aux utilisateurs inscrits ainsi qu'au gestionnaire.

II.2 Serveur

L'application web sera installée sur un RaspberryPi4 qui aura un serveur web un serveur SGBD (Mysql), et une ou des bases de données.

II.3 Base de données

Nous devons réaliser une base de données qui permettra le stockage des données de l'application web. L'application web doit avoir une base de données de l'historique avec un utilisateur et le module qu'il a utilisé.

Elle doit avoir une table de données pour les statistiques avec le nombre de visites et le nombre de modules utilisés.

Elle doit avoir une table de données de l'utilisateur inscrit avec son login, le mot de passe, un email.

Elle doit aussi avoir une table de données pour le gestionnaire. Il y aura un seul gestionnaire avec son login : gestion, son mot de passe : à définir, un email. à l'hexadécimal, calculer les coûts d'amortissement d'un projet...

II.4 Utilisateur inscrit

Dans cette application web, il devra y avoir un utilisateur inscrit (dans la base de données de l'application, grâce à son login et son mot de passe) qui pourra utiliser la plateforme de simulation pour faire différents calculs.

Il pourra accéder au menu de son profil pour y changer son mot de passe. L'utilisateur devra se déconnecter.

Lors de la connexion il devra y avoir un lien pour “oublie de mot de passe” qui se dirigera vers une page en construction.

II.5 Utilisateur non inscrit (visiteur)

Si l'utilisateur n'est pas enregistré dans la base, appelé “visiteur”, il n'a pas accès aux outils de simulation,

mais il pourra accéder à la page d'accueil qui a pour but d'expliquer pourquoi la plateforme a été conçue, avec une vidéo de démonstration incitant le visiteur à s'inscrire.

Il pourra s'inscrire en remplissant un formulaire d'inscription avec login et mot de passe conforme et devra valider un captcha pour devenir utilisateur. Ses données entrées seront enregistrées dans une base de données.

II.6 Gestionnaire

Parmi les autres acteurs, on retrouve 1 seul gestionnaire inscrit dans la base de données, et se connectera à la plateforme avec un login “gestion” et un mot de passe à définir et à transmettre au client pour qu'ils puissent faire des tests.

Le gestionnaire pourra visualiser la liste des utilisateurs inscrits dans l'application sans voir leur mot de passe et pourra aussi en supprimer, ce qui entraînera la suppression de l'historique de celui-ci. Aussi le gestionnaire pourra visualiser les données des utilisateurs. Il pourra ensuite établir une statistique sur les visites et les simulations utilisées par les utilisateurs inscrits.

Le gestionnaire devra se déconnecter.

III / Pré-requis

Utilisation de gitlab afin de catégoriser, centraliser et partager les documentations avec le client.

Les langages de développement utilisés pour faire ce projet sont PHP et Mysql, ainsi qu'avec le html, le shell, le css.

Le HTML pour les pages statiques avec le css pour leurs rendre un style, Shell pour le serveur, Sql pour les requêtes et la base de données et le php pour rendre les pages dynamiques avec un serveur et une base de données.

IV / Priorités

Le cahier des charges finales