

Stellar Sentry TD



Lien vers GitHub : https://github.com/ValentinLoiseau1/stellar_sentry_td

Lien vers O2Switch : https://greta-bretagne-sud.fr/stagiaires-kercode/valentin-loiseau/stellar_sentry_td

Lien vers mon cahier des charges : https://github.com/ValentinLoiseau1/SS_TD/blob/main/Cahier_des_charges18_03_24.pdf

Sommaire :

1 Contexte du projet

2 Maquettage

3 Conception

3.1 MCD

3.2 MVC

4 Choix techniques

5 Sécurité

6 Chronologie du travail

7 Amélioration possible

1 Contexte du projet

Le projet "Stellar Sentry TD" est né de la vision partagée de deux développeurs passionnés, désireux de créer un jeu vidéo qui reflète leurs intérêts et leur identité.

Ils ont choisi de se lancer dans le développement d'un tower defense se déroulant dans un univers de science-fiction.

L'objectif principal de ce site web est de fournir aux futurs clients/joueurs une plateforme centralisée pour découvrir et explorer le jeu "Stellar Sentry TD".

Il vise à offrir aux visiteurs la possibilité de s'informer sur le jeu, d'essayer différentes versions disponibles, de partager leurs avis et de consulter les avis d'autres utilisateurs.

En outre, le site a pour ambition de présenter une feuille de route détaillée, permettant aux utilisateurs de visualiser les futures mises à jour et fonctionnalités prévues à travers les différents trimestres de développement.

Ceci servira à orienter les priorités de développement en fonction des besoins et attentes des utilisateurs. Le projet vise un public diversifié, incluant des joueurs de tous âges. Étant donné la longue existence et la popularité continue des jeux de ce type, il existe un marché ouvert et une demande certaine pour des nouveautés, ce jeu vise donc à répondre à ces attentes.

Initialement développée en français, l'interface utilisateur du jeu et du site web sera accessible dans cette langue.

Toutefois, une version en anglais sera également envisagée dans le but d'atteindre un public plus large, offrant ainsi une accessibilité internationale.

En ce qui concerne le support de développement, le jeu sera prioritairement destiné aux ordinateurs de bureau (desktop).

Quant au site web, il sera conçu pour offrir une expérience optimale sur toutes les tailles d'écran, assurant ainsi une accessibilité maximale sur les différents appareils utilisés par les joueurs.

2 Maquettage

Pour la conception de la maquette de notre site web, j'ai opté pour l'utilisation de Figma, un outil que je considère comme très performant. J'ai suivi une approche "mobile first" lors de sa réalisation, mettant ainsi l'accent sur l'expérience utilisateur sur les appareils mobiles.

Bien que la maquette ait servi de base pour le développement du site, quelques ajustements ont été nécessaires en cours de route, soit pour répondre à des priorités changeantes dans le développement, soit pour ajuster les contraintes de temps.

Liens vers la maquette Figma : <https://www.figma.com/file/c12G3iNm2PRAEKsiq9NZff/Untitled?type=design&node-id=0-1&mode=design&t=Ue6MO4qOaNVcja8B-0>

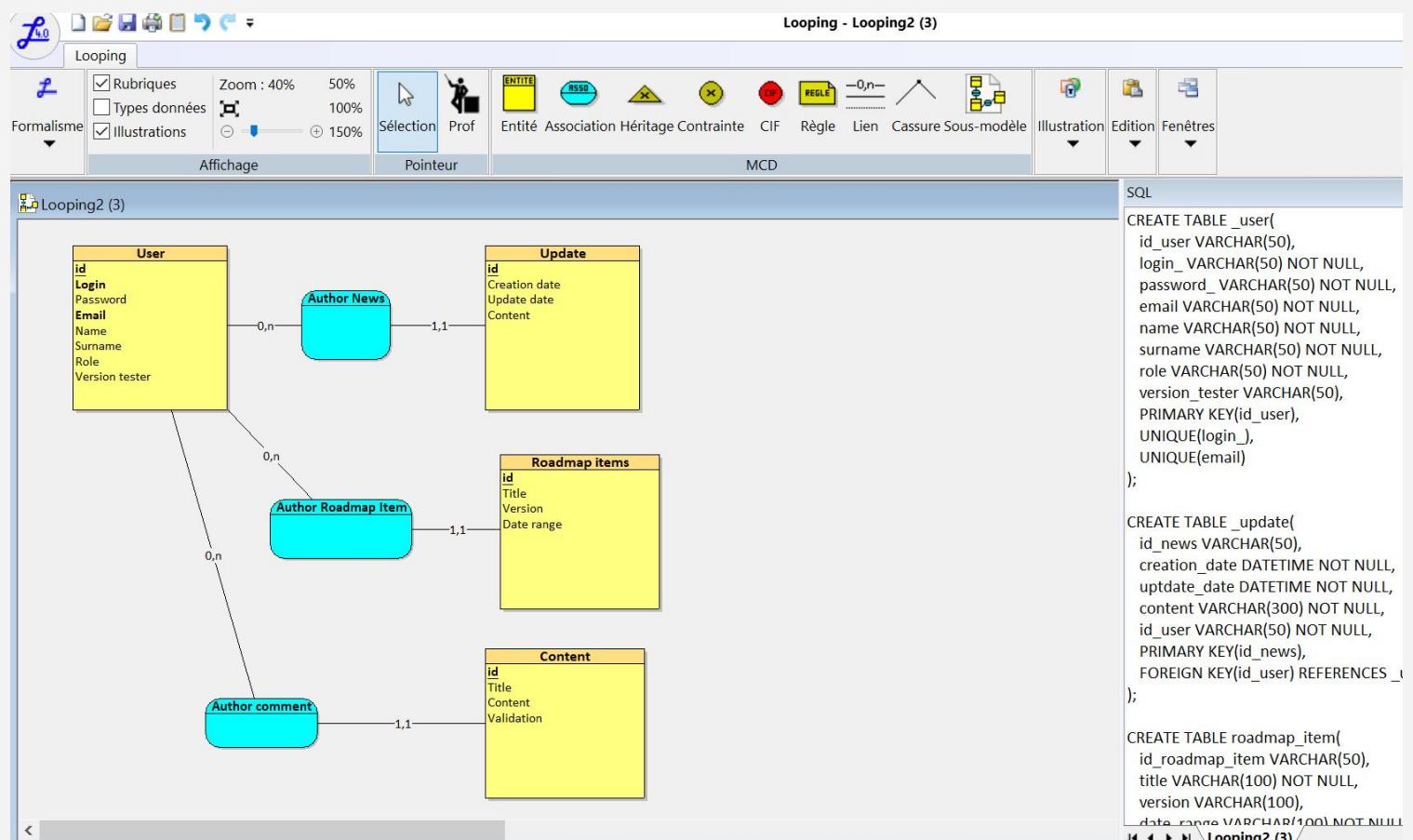
3 Conception

3.1 MCD

Pour la conception du Modèle Conceptuel de Données (MCD), j'ai utilisé l'outil Looping, qui m'a permis de créer efficacement mes tables, leurs associations et de définir leurs paramètres en amont.

Une conception solide du MCD est essentielle pour pouvoir se concentrer pleinement sur le développement, en disposant d'une base de données structurée et réfléchie.

Une fois la conception achevée, Looping m'a fourni un fichier SQL que j'ai ensuite importé dans phpMyAdmin pour procéder à la mise en place de la base de données.



3.2 MVC

Concernant la conception et l'organisation de mon code dans Visual Studio Code, j'ai opté pour une structuration en Modèle-Vue-Contrôleur (MVC).

Model : Le modèle représente la structure des données et gère leur accès. Il encapsule la logique métier et interagit avec la base de données.

- View : La vue est responsable de l'interface utilisateur. Elle affiche les données du modèle à l'utilisateur et collecte les actions de l'utilisateur pour les transmettre au contrôleur.
- Contrôleur : Le contrôleur agit comme un intermédiaire entre le modèle et la vue. Il reçoit les requêtes de l'utilisateur, traite les données nécessaires en utilisant le modèle approprié, puis sélectionne la vue appropriée pour afficher les résultats.

Cette méthodologie de travail offre une structure claire à l'ensemble du projet, ce qui facilite la collaboration entre les membres de l'équipe, améliore la lisibilité du code et permet une maintenance et une évolution plus aisées de l'application.

4 Choix techniques

Frontend :

Langages de balisage : HTML

Utilisé pour la structure et le contenu de la page web, y compris la création de différentes sections pour afficher les détails du jeu, les avis des utilisateurs, etc.

Langages de style : CSS, SCSS

Utilisation d'un préprocesseur pour la présentation visuelle du site web, y compris la mise en page, les couleurs, les polices, etc. SASS (Syntactically Awesome Stylesheets) peut être utilisé pour une gestion plus efficace et organisée du style.

Langages de programmation : JavaScript

Utilisé pour ajouter de l'interactivité au site web, comme la validation des formulaires, la manipulation dynamique du contenu, les animations, etc.

Backend : Langage de programmation : PHP

Utilisé pour implémenter la logique métier du site web, y compris la récupération et la manipulation des données du jeu depuis la base de données, la gestion des utilisateurs, la gestion des sessions, etc.

Base de données : Système de gestion de base de données (SGBD) : MySQL.

Utilisé pour stocker les informations spécifiques au jeu, telles que les articles, les formulaires de contact, les développeurs, ainsi que les informations utilisateur.

Développement et optimisation :

Design architectural du back-end : MVC (Modèle-Vue-Contrôleur)

Utilisé pour structurer le code de manière à séparer clairement la logique métier, la présentation et le contrôle.

Outils de développement : IDE (Environnement de Développement Intégré), outils de débogage, outils de gestion de version. (Vs Code, console de navigateur).

Utilisé pour faciliter le processus de développement, la collaboration et la gestion du code source.

Utilisé pour garantir des temps de chargement rapides, une expérience utilisateur fluide et une bonne optimisation des ressources.

5 Sécurité

Hashage des mots de passe : Pour garantir la sécurité des mots de passe des utilisateurs, nous utilisons des algorithmes de hachage robustes. Le hachage des mots de passe permet de les stocker de manière sécurisée dans la base de données, rendant ainsi les informations des utilisateurs inintelligibles en cas de compromission de la base de données.

Protection contre les injections de code : Afin de contrer les attaques par injection de code, nous mettons en place des techniques qui empêchent les attaquants de manipuler les données saisies par les utilisateurs pour injecter du code malveillant dans nos requêtes SQL.

Utilisation de htmlspecialchars : Pour éviter les attaques et garantir la sécurité des données affichées dans les pages web, nous utilisons la fonction htmlspecialchars pour convertir les caractères spéciaux en entités HTML.

Vérification des champs saisis : Nous mettons en place des mécanismes de validation côté serveur pour vérifier les données saisies par les utilisateurs. Cela inclut la vérification de la longueur, du format et de la validité des données avant de les traiter ou de les stocker dans la base de données. En validant les entrées utilisateur, nous réduisons les risques d'injections de code malveillant ou d'autres types d'attaques.

6 Chronologie du travail

Finalisation du Cahier des Charges et des Spécifications :

Objectif : Clôturer la phase de définition des besoins, des fonctionnalités et des exigences du site web.

Responsable : Chef de projet / Équipe de développement.

Délai : Le cahier des charges doit être finalisé pour le 15 mars 2023.

Livrables attendus : Cahier des Charges approuvé et Spécifications détaillées.

Conception du Backend :

Objectif : Élaborer l'architecture système, définir la structure de la base de données et planifier les fonctionnalités côté serveur.

Responsable : Développeurs Backend / Architecte logiciel.

Délai : La conception base de donnée doit être finalisé pour le 23 mars 2023 . Livrables attendus : Schéma d'architecture, modèle de base de données.

Conception du Frontend :

Objectif : Créer la partie visible du site web en rassemblant les éléments graphiques, en élaborant les wireframes des pages et en définissant l'interface utilisateur.

Responsable : Développeurs Frontend / Designer UI/UX.

Délai : La conception Frontend doit être finalisé pour le 23 mars 2023.

Livrables attendus : Wireframes, éléments graphiques validés.

Assemblage :

Objectif : Intégrer le backend et le frontend pour former une version fonctionnelle du site.

Responsable : Équipe de développement / Intégrateurs.

Délai : L'assemblage doit être finalisé pour le 12 avril 2023.

Livrables attendus : Version bêta du site web assemblé.

Tests Fonctionnels :

Objectif : Vérifier que toutes les fonctionnalités du site web répondent aux spécifications et sont exemptes de bugs.

Responsable : Équipe de test / Développeurs. Délai : Les tests à l'issue de l'assemblage devront être aboutis pour le 19 avril 2023.

Livrables attendus : Rapports de tests, corrections des anomalies.

Déploiement :

Objectif : Mettre le site web en ligne et le rendre accessible au public.

Responsable : Administrateur système / Équipe de développement.

Délai : Le déploiement doit être opéré pour le 10 mai 2023. Livrables attendus : Site web déployé et fonctionnel.

7 Amélioration possible

- Renforcer le style des avis postés par les utilisateurs pour offrir une expérience visuelle plus attrayante.
- Rendre la page administrateur plus intuitive en simplifiant les processus de gestion et de surveillance du site.
- Ajouter la possibilité pour les utilisateurs de supprimer leur compte ainsi que tous les commentaires qu'ils ont publiés, s'ils le souhaitent, afin de garantir leur contrôle sur leurs données personnelles.
- Améliorer le traitement des erreurs en mettant en place des messages d'erreur plus descriptifs et en offrant des solutions alternatives aux utilisateurs en cas de problème.
- Optimiser la taille et le poids des contenus tels que les images pour réduire la consommation de ressources et favoriser une empreinte environnementale réduite, dans une démarche d'éco-responsabilité.