

Projet XYZ

Table des matières

1	Analyse préliminaire	4
1.1	Introduction	4
1.2	Objectifs.....	5
1.2.1	Livrable	5
1.2.2	Objectif du produit.....	6
1.3	Gestion de projet	6
1.4	Planification initiale	7
2	Analyse / Conception.....	9
2.1	Contexte produit	9
2.1.1	Utilisation	10
2.2	Contexte technique.....	11
2.2.1	Opérationnel	11
2.2.2	Environnement de développement	13
2.2.3	Environnement de production	13
2.2.4	Justification des choix.....	13
2.3	Dossier de conception	14
2.3.1	Zonings.....	14
2.3.2	Maquettes.....	16
2.3.3	Modèle de base de données	20
2.3.4	Analyse fonctionnelle.....	20
2.4	Stratégie de test.....	21
2.5	Risques techniques	23
2.6	Planification	23
3	Réalisation.....	23
3.1	Point de design spécifique.....	23
3.2	Dossier de réalisation	23
3.3	Description des tests effectués.....	24
3.4	Erreurs restantes	24
3.5	Liste des documents fournis	24

4	Conclusions	25
5	Annexes.....	26
5.1	Résumé du rapport du TPI / version succincte de la documentation	26
5.2	Sources – Bibliographie	26
5.3	Journal de travail	26
5.4	Manuel d'Installation	26
5.5	Manuel d'Utilisation.....	26
5.6	Archives du projet.....	26

1 Analyse préliminaire

1.1 Introduction

L'objectif principal de ce projet est de créer une application de simulation de saison de Formule 1. Elle se présentera sous forme de jeu afin de divertir ses utilisateurs. Cette application sera développée sous forme d'application web en utilisant le langage de programmation Javascript.

Ce projet se fait dans le cadre du TPI. Cela veut dire que le projet sera réalisé en suivant les modalités du TPI. Le dossier de projet ainsi que la présentation faisant suite à ce projet seront aussi évalués en suivant la grille d'évaluation TPI VD.

Plusieurs projets ont été réalisés en prévision de ce TPI. Le premier, un projet de monitoring de ruche d'abeille, n'avait aucun lien avec la technologie qui sera utilisée ni aucun rapport avec le thème (La formule 1) de ce TPI. Cependant, ce projet, ayant été évalué avec la grille d'évaluation du TPI, m'a permis de me familiariser avec cette dite grille ainsi que ce que devrait contenir le dossier de projet. Le second projet la création d'un site web de shopping avec un back-office était, lui, un projet non-officiel. Cela veut dire que le projet ne suivait pas de cahier des charges ni d'évaluation finale mais me servait uniquement de moyen d'assimiler la technologie qui sera utilisée dans ce TPI et d'y gagner en aisance.

Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

Ces éléments peuvent être repris des spécifications de départ.

1.2 Objectifs

1.2.1 Livrable

Une planification initiale contenant un diagramme de Gantt devra être livrée aux 2 experts ainsi qu'au chef de projet par courriel électronique sous le format PDF à la fin du premier jour (2 mai 2025).

Le rapport de projet en l'état devra être livré 3 fois par semaine le lundi, le mercredi et le vendredi en fin de journée. Le rapport de projet sera livré sous forme de PDF dans le repo Git dans un dossier dédié à la documentation.

Le livrable finale devra contenir :

- Le journal de travail avec mention de tout ce qui a été accompli durant le projet devra aussi être livrée.
- Le code source de l'application devra être livré aux 2 experts ainsi qu'au chef de projet. La livraison se fera sous la forme de partage de lien au dépôt Git.
- Les scripts qui serviront la création de base de données ainsi que des données de test
- Un guide de déploiement de l'application sur l'infrastructure locale de l'ETML

A la fin du délai imparti pour la réalisation du TPI, c'est à dire avant le 2 juin 2025 à 11h20, une version électronique du dossier de projet devra être transmise aux 2 experts ainsi qu'au chef de projet. En parallèle, une copie papier du rapport devra être fournie sans délai en trois exemplaires. Elle devra être en tout point identique à la version électronique.

1.2.2 Objectif du produit.

Comme énoncé dans l'introduction, l'objectif du projet est de créer une application de simulation de saison de Formule 1. Etant donné que le descriptif peut porter à confusion, je vais décrire, ici, ce à quoi consiste cette application. En effet, il ne s'agit pas de simuler la saison de Formule 1 actuelle, mais de donner la possibilité à un utilisateur de se divertir en choisissant une écurie, choisir les 2 pilotes qui composeront cette écurie et de simuler une saison entière. Une fois la saison lancée, l'utilisateur n'aura aucune influence sur le résultat des courses. Les résultats des courses seront aléatoires en prenant en compte le rating¹ du pilote. Les courses comprendront au moins un accident et un abandon. Cela sera un jeu en solo. Cela veut dire qu'aucun autre joueur pourra prendre part à la session de jeu de l'utilisateur.

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

1.3 Gestion de projet

Une planification initiale sera faite le premier jour de ce projet, directement après la réception du cahier des charges (la planification initiale est trouvable sous le chapitre 1.4).

La méthodologie utilisée lors de ce projet de TPI est celle dite « en cascade ». C'est-à-dire que lors de la planification initiale, le projet a été analysé pour en ressortir toutes les tâches qui seront réalisées. Les tâches ont été mises ensuite dans un ordre logique pour suivre la logique du « Waterfall » qui est qu'une tâche ne peut pas être réalisée

¹ Niveau de compétence du pilote

avant que la précédente ne soit finie. En effet, il sera difficile de s'occuper de la gestion de course avant de s'occuper de la gestion des choix des pilotes et des écuries.

L'entièreté du projet sera disponible sur une repo github (Code source de l'application, planification, journal de travail, rapport, cahier des charges, schémas).

Les personnes impliquées dans ce projet sont :

- Gaël Sonney, chef de projet.
- Yves Bertino, expert n° 1
- Nemanja Pantic, expert n° 2

1.4 Planification initiale

Une planification a donc été faite le premier jour de ce TPI. Les tâches ont d'abord été séparées en plusieurs catégories :

- Entretien, les différents entretiens avec les experts ou le chef de projet.
- Analyse, analyse du projet ou d'une fonctionnalité.
- Réalisation, développement d'une fonctionnalité ou réalisation d'une tâche précise.
- Test, tester une fonctionnalité.
- Documentation, rédiger la partie de la documentation du projet.

Un découpage des tâches plus précises a ensuite été fait. Puis, pour chaque séquence de travail, une estimation des tâches qui seront réalisées ainsi que le temps passé dessus a été faite. Sachant qu'une séquence est définie par une demi-journée de travail.

[Insérer diagramme]

A Garder à l'esprit que cette planification initiale n'est qu'une première idée de comment pourrait se dérouler le projet et qu'elle n'est là que pour donner une idée de l'ordre des tâches à effectuer ainsi que le temps estimé que prendra chacune d'elle. Il est très peu probable que cette planification initiale soit respectée parfaitement. Des imprévus peuvent survenir, de nouvelles idées, des idées de réalisation différentes, ou une mauvaise estimation du temps peuvent en être la cause.

La période de réalisation de ce projet du TPI est du **vendredi 2 mai 2025 à 8h au lundi 2 juin 2025 à 11h25**.

Les horaires de travail sont :

Lundi	08h00 – 11h25	12h20 – 14h45
Mardi	-	-
Mercredi	08h00 – 12h15	13h10 – 16h35
Jeudi	-	13h10 – 16h35
Vendredi	08h00 – 12h15	13h10 – 16h35

Avec comme jours fériés :

- Lundi 26 mai après-midi, Examen eCG
- Jeudi 29 mai, Ascension
- Vendredi 30 mai, Pont de l'Ascension

Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.

Ces éléments peuvent être repris des spécifications de départ.

2 Analyse / Conception

2.1 Contexte produit

Le Produit sera un accès à un site internet. L'utilisateur devra en premier lieu se connecter afin de pouvoir profiter des fonctionnalités. Le site web sera principalement développé pour les navigateurs web via ordinateur. Si un maximum sera fait pour rendre le site web plus modulable, il ne sera pas pensé pour avoir un responsive design complet. Le site sera tout de même accessible sur un navigateur web depuis un téléphone mobile, mais la vue ne sera pas totalement adaptée.

L'application aura 2 types d'utilisateur :

- Les utilisateurs simples, qui pourront jouer et consulter les données
- Les administrateurs, qui pourront modifier les données.

Actions / Rôles	User	Admin
Voir les informations des écuries	✓	✓
Modifier les informations des écuries	✗	✓
Supprimer une écurie	✗	✓
Ajouter une écurie	✗	✓
Voir les informations des pilotes	✓	✓
Modifier les information des pilotes	✗	✓
Supprimer un pilote	✗	✓
Ajouter un pilote	✗	✓
Voir les informations des circuits	✓	✓
Modifier les informations des circuits	✗	✓
Supprimer un circuit	✗	✓

Ajouter un circuit	✗	✓
Lancer une saison	✓	✓

2.1.1 Utilisation

Prenons l'exemple d'utilisation d'un utilisateur simple. Il devra en premier lieu se connecter au site internet avec ses identifiants. S'il n'en a pas, il aura la possibilité de créer un compte. Ensuite, il aura l'occasion de consulter les données des pilotes, des circuits ou des écuries s'il en a l'envie. Pour commencer une partie, il devra se rendre sur la page principale puis cliquer sur le bouton « Lancer une saison ». Le site web affichera ensuite une liste des écuries, il en choisira une. Puis lui sera affiché la liste des pilotes, il fera de même mais en choisira 2. Ce seront ses 2 pilotes qui courront pour lui dans l'écurie qu'il a choisie. Une fois confirmé, la saison débute. Chaque course se lanceront une à une, un classement des pilotes et des constructeurs sera affiché à la fin de chaque course. L'utilisateur devra, néanmoins, confirmer à chaque fin de course avant de passer à la suivante. A la fin de la saison, un classement final sera affiché. Le joueur se retrouvera donc comme au début de sa session de jeu, sur la page principale avec la possibilité de recommencer une partie.

Placer le produit dans son contexte d'utilisation. Par exemple :

- **Site Web Internet**
- **Application, Web, intranet**
- **Application mobile**
- **Infrastructure dupliquer des sites multiples**

Présenter les utilisateurs du système ainsi que les rôles qu'ils peuvent avoir

Présenter les données/information que le système prend en charge

Au minimum :

- *Un ou plusieurs schémas de contexte montrant le produit dans son environnement d'utilisation, ainsi que ses utilisateurs. Ce type schéma doit être accompagné d'explications textuelles*

Le concept complet avec toutes ses annexes:

Par exemple :

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- *...*

2.2 Contexte technique

2.2.1 Opérationnel

Le produit final est divisé en plusieurs parties :

2.2.1.1 Frontend

La partie frontend, c'est-à-dire la partie qui sera visible par l'utilisateur, sera développée en utilisant Vite, React ainsi que TypeScript², plus précisément avec le format TSX. Le TSX nous permettra d'intégrer directement du TypeScript dans nos balise JSX³.

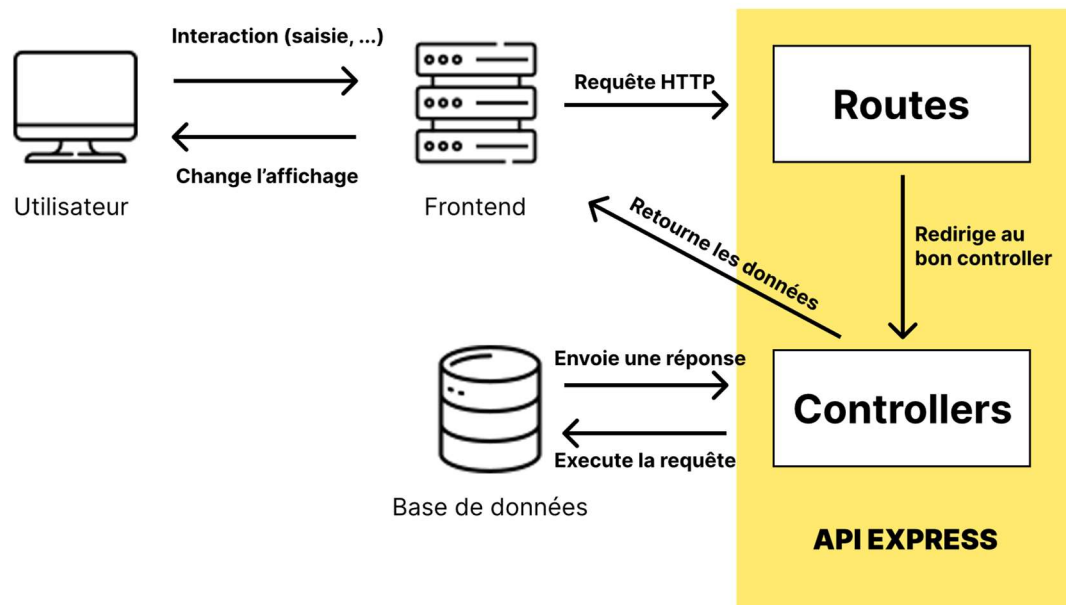
Vite JS est, par définition, un serveur de développement utilisé pour faire tourner localement des applications web. Cela nous permettra aussi de faire un build de notre application afin de le déployer en production.

2.2.1.2 Backend

La partie backend, la partie dont l'utilisateur n'a pas directement accès, sera développée avec Express JS. Express JS est un framework backend pour Node.js, qui nous permet de créer des API, de gérer des requêtes HTTP et faire du routage. Son fonctionnement peut se décrire de cette façon :

² Langage ressemblant au JS mais typé

³ Extension de syntaxe pour JS permettant d'y intégrer du HTML



Prenons l'exemple où l'utilisateur se rends sur la page des pilotes. Lors du chargement de la page, la partie frontend effectue une requête HTTP sous forme de « `http://localhost:3040/api/drivers` » est effectuée (dans le cadre de l'environnement de développement). Dans le fichier `index.js` de notre backend toutes les routes API y sont définies. Parmi elles, la route `/api/drivers` est associé au fichier `routes/drivers.js`.

Ce fichier de routes importe les fonctions définies dans notre Controller et lie chaque routes avec une fonction.

Dans notre exemple, comme notre URL est simplement « `api/drivers/` » (aucun paramètre supplémentaire), c'est la fonction associée à la route « `/` » qui est exécutée. Dans notre cas c'est la fonction « `getAllDrivers` ». La fonction avec ce nom présent dans le `DriversController` va ensuite s'exécuter et va effectuer une requête SQL à notre base de données. Le controller va ensuite retourner les données reçues à la partie frontend qui va mettre à jour l'affichage et afficher la liste des pilotes

Placer le produit dans son contexte d'opérations.

Ce chapitre doit impérativement contenir au moins une représentation graphique

Au minimum :

- *Un ou plusieurs schémas de contexte montrant le produit dans son environnement technique. Ce type schéma doit être accompagné d'explications textuelles*

2.2.2 Environnement de développement

L'environnement de développement sera l'application lancée en local sur un PC. Le frontend sera déployé sur le port 5174 tandis que le backend le sera sur le 3040. Il faudra donc démarrer l'environnement via une console, celle de VSC par exemple, via la commande « npm run dev » pour le frontend puis « node index.js » pour le backend. Cela veut dire que cet environnement n'est pas accessible depuis un poste extérieur. Comment son nom l'indique, c'est sur cet environnement que les fonctionnalités seront d'abord développées puis testées avant d'être remises sur l'environnement de production.

2.2.3 Environnement de production

L'environnement de production sera le serveur disponible à l'ETML. Ce serveur contient un service Nginx et Node ce qui nous permet d'héberger mon application web. Le port ouvert et accessible qui m'a été remis est le port 3040. L'application devra donc être modifiée pour que le backend et le frontend utilisent les 2 ports attribués. L'application, étant hébergée sur le serveur de l'ETML, sera donc accessible par n'importe quel appareil connecté à internet depuis un navigateur web. Le lien pour y accéder est « aki.w3.pm2etml.ch ». Lorsqu'une fonctionnalité a passé tous les tests sur l'environnement de dev, elle pourra être pushée sur le serveur. Un nouveau build de l'application devra être fait pour que les changements prennent effet.

2.2.4 Justification des choix

2.3 Dossier de conception

Dans ce chapitre de concept, nous verrons tout ce qui est zonings, maquettes du site web, les différents schémas dont ceux de la base de données et ceux du fonctionnement des fonctionnalités du site web.

2.3.1 Zonings

Les zonings sont les premières esquisses graphiques des pages du sites web permettant de visualiser l'organisation des différents éléments qui seront présents sur les pages. Les zonings ont été réalisé sur Figma.

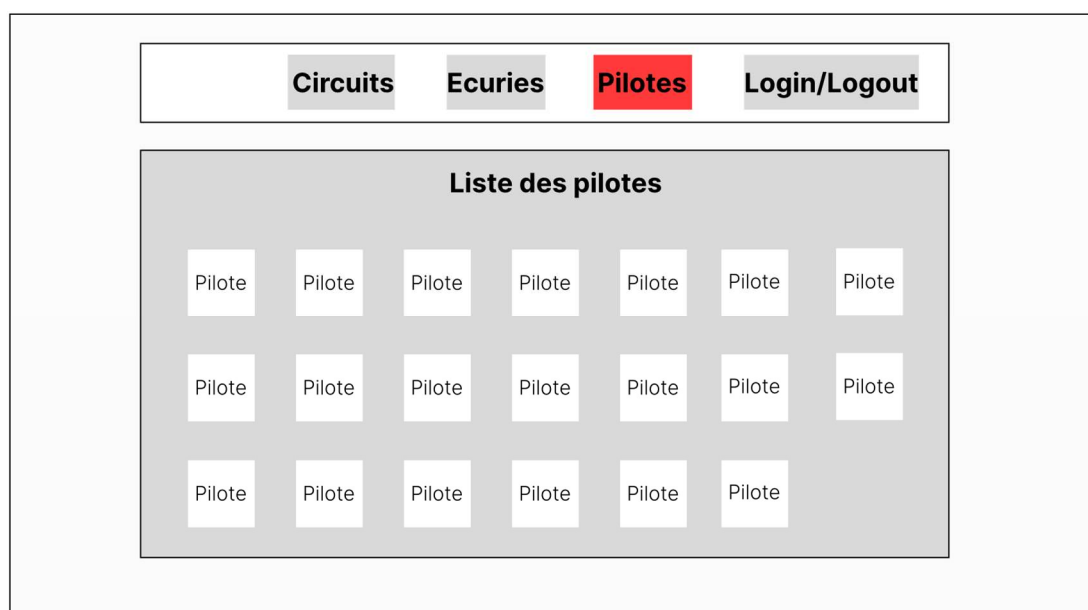
2.3.1.1 Page principale

Sur la page principale, on trouvera la barre de navigation en haut de notre écran qui contiendra les différents boutons pour ensuite naviguer à travers les différentes pages. Au centre de l'écran se trouvera la « zone de jeu ». C'est ici que se présentera le terrain de jeu pour notre utilisateur. Elle contiendra par exemple le déroulement de la course, ou bien le classement de la course une fois la course terminée.



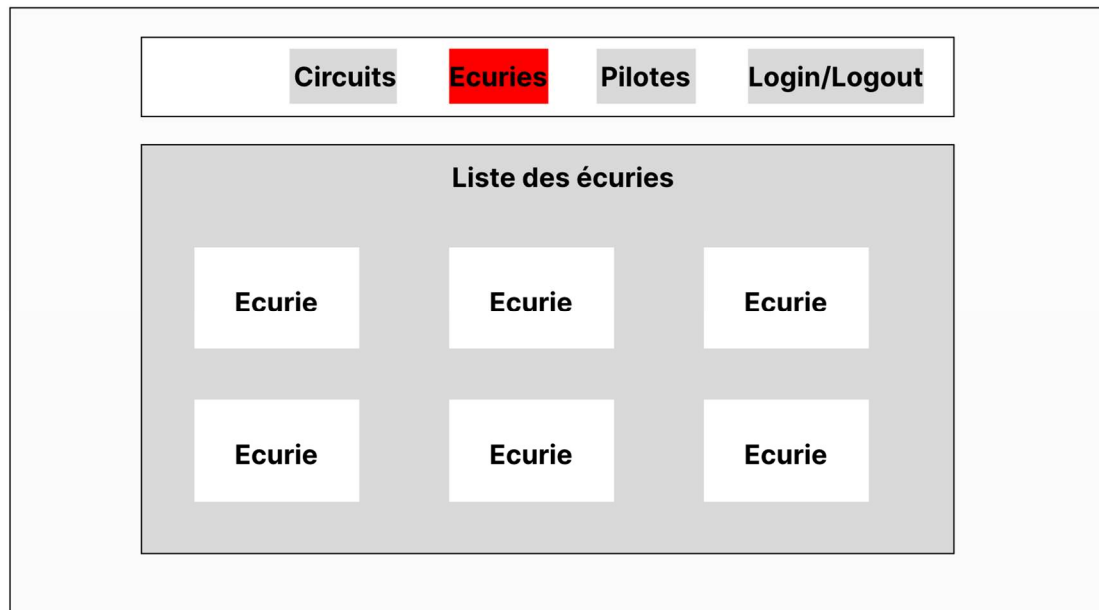
2.3.1.2 Liste des pilotes

Cette page contiendra, comme toutes les autres, la barre de navigation horizontale en haut de l'écran. La liste des pilotes y figurera. Un simple utilisateur pourra ne pourra que consulter ces informations. Seul un utilisateur ayant comme rôle « Administrateur » pourra ajouter, modifier ou supprimer un des pilotes.



2.3.1.3 Liste des écurie et des circuits

Ces pages sont relativement similaires à celle des pilotes dans leur fonctionnement. C'est-à-dire que seul un utilisateur de type « Administrateur » pourra y faire des modifications.

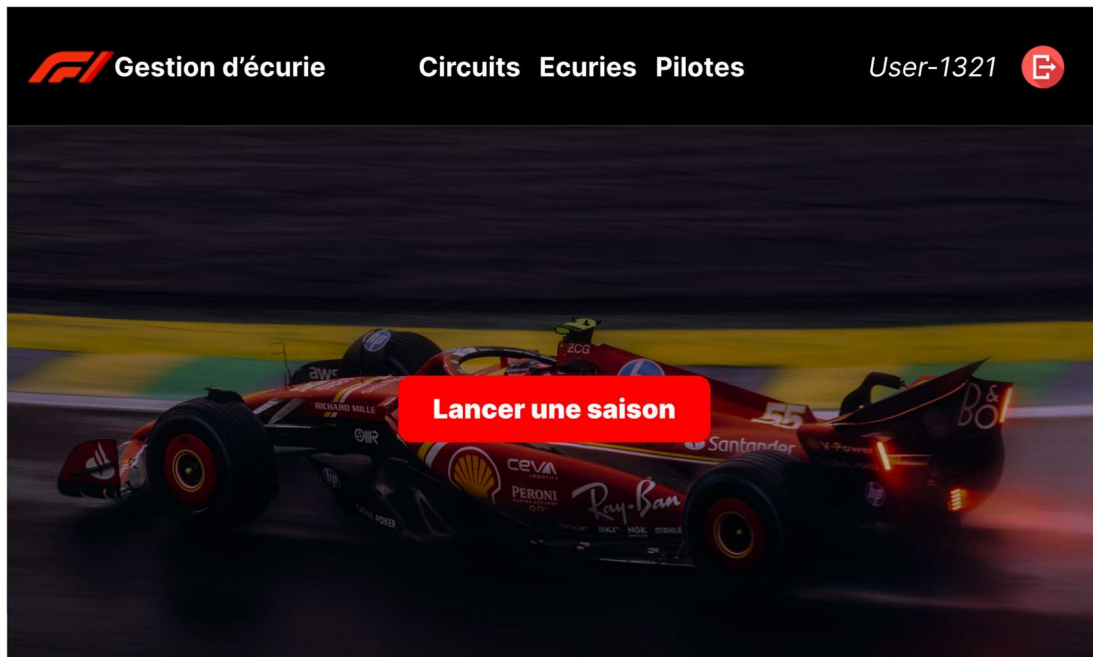


Etant donné que le zoning de la liste des circuits ressemble énormément à celui des écuries, j'ai trouvé plus pertinent de ne pas le mettre dans le rapport. Cependant il sera bien évidemment mis en annexe.

2.3.2 Maquettes

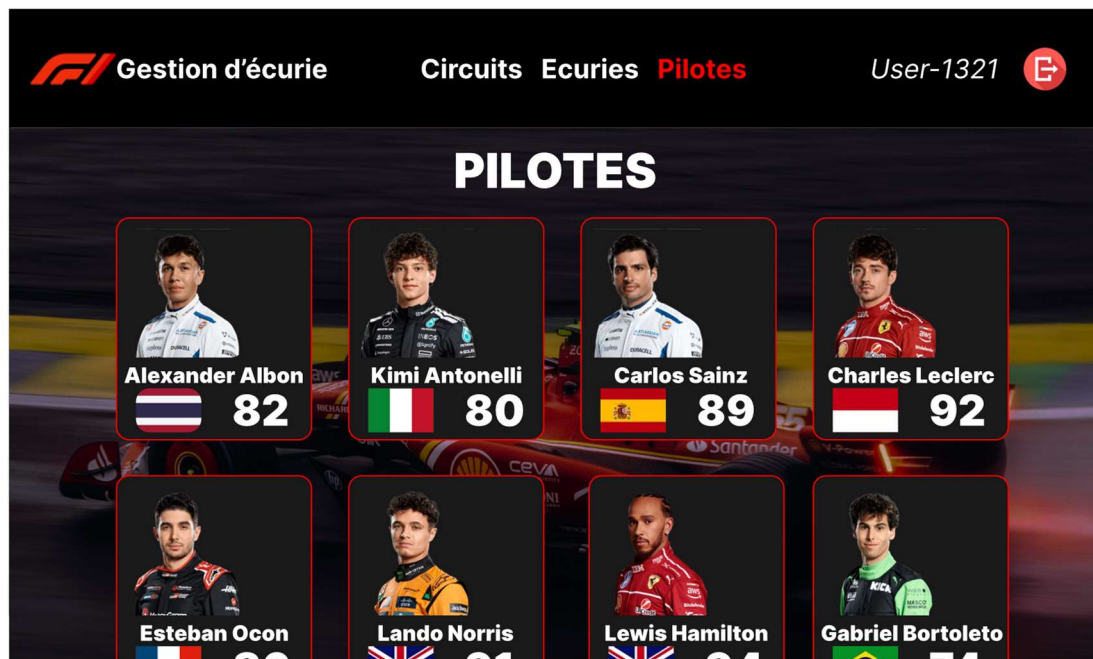
Les maquettes sont beaucoup plus complètes que les zonings et se rapproche plus du résultat final. Elles ont, comme pour les zonings, été faite sur Figma.

2.3.2.1 Page principale



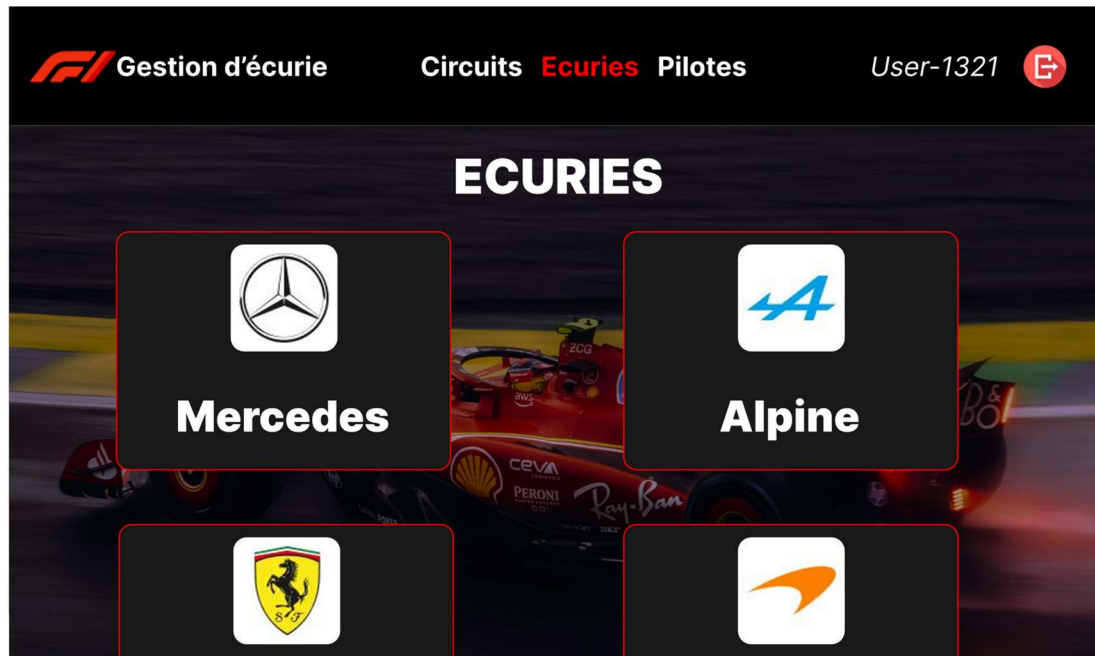
Sur la barre de navigation, la section ou était présent la zone « login/logout » dans le zoning contiendra le nom de l'utilisateur connecté ainsi qu'un bouton de déconnexion qui le reverra sur la page de connexion.

2.3.2.2 Liste des pilotes



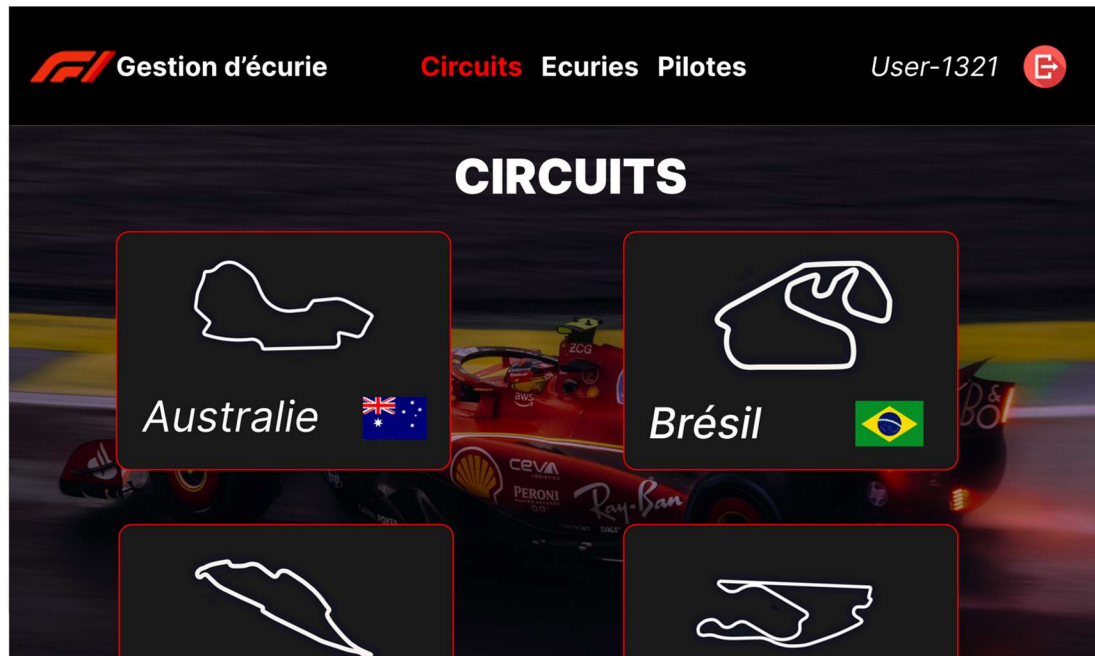
Les zones présentent sur la maquette se rapproche assez des zonings mais apportent plus de précision notamment au niveau de container des pilotes. Leurs nom sera affiché ainsi qu'un image, un rating et leur nationalité. Un administrateur aura une vue un peu différente de celle d'un simple utilisateur. Il verra affiché un bouton à côté de chaque pilote afin de pouvoir modifier ses informations. Un bouton redirigeant sur un formulaire de création de pilote sera aussi accessible uniquement aux administrateurs.

2.3.2.3 Liste des écuries



La liste des écuries reprends aussi les zonings avec la même fonctionnalité de connexion dans la barre de navigation. Un administrateur aura, de même, accès à un bouton pour modifier ou supprimer une écurie ainsi qu'un bouton qui redirige vers un formulaire de création d'écurie.

2.3.2.4 Liste de circuits



La liste des circuits reprend globalement l'affichage des écuries. Encore une fois, seul l'administrateur aura les droits de modification, de suppression et d'ajout de circuit tandis qu'un simple utilisateur n'aura que les droits de lecture.

2.3.3 Modèle de base de données

2.3.4 Analyse fonctionnelle

2.3.4.1 Connexion de l'utilisateur

L'utilisateur devra être identifié pour pouvoir profiter des fonctionnalités de l'application.

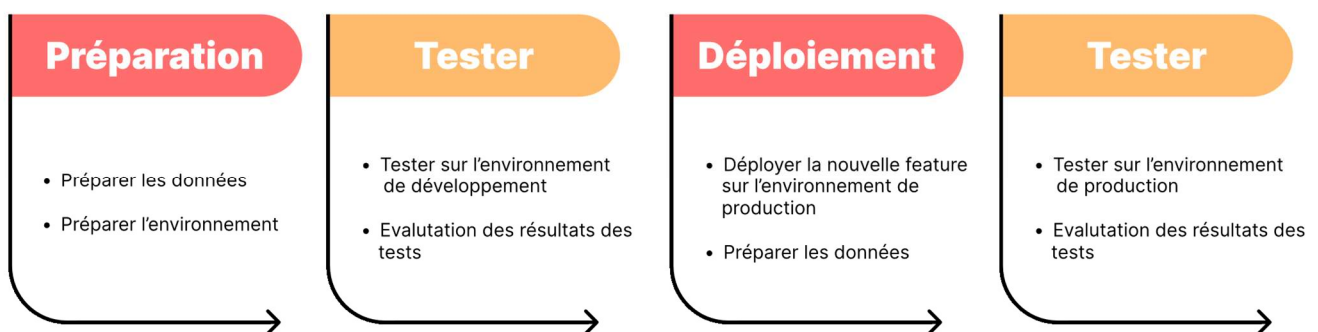
Fournir tous les document de conception:

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

2.4 Stratégie de test

Différents tests seront effectués durant ce projet. La stratégie de test adoptée est la suivante :



Lorsqu'une nouvelle fonctionnalité a fini d'être développée, nous allons d'abord être sûr que les données requises sont prêtes. Nous allons aussi nous assurer que notre environnement est aussi prêt (c'est-à-dire que notre environnement est démarré avec

la dernière fonctionnalité disponible). Je vais d'abord effectuer la liste des tests pour cette fonctionnalité sur l'environnement de développement. Puis les tests seront évalués selon les résultats attendus. Si tous les tests passent. La fonctionnalité sera déployée sur l'environnement de production. Si elle ne passe pas, une analyse des bugs sera faite afin de connaître ce qu'il faut corriger. Une fois que la feature est déployée sur l'environnement de production. Etant donné que cet environnement est différent de celui de développement, je vais réeffectuer la liste de test et réévaluer les résultats. Si tout est bon, la fonctionnalité sera déclarée comme « passée ». Sinon il faudra analyser l'erreur. Il faudra tout de même faire attention à ce qu'une nouvelle feature n'en casse pas une autre. Refaire rapidement les tests des autres features présentes sur le site sera une chose à faire.

Je suis conscient qu'effectuer des tests sur l'environnement de production n'est pas l'idéal. Il serait préférable d'avoir un environnement de test non accessible aux utilisateurs et similaire à l'environnement de production dédié uniquement aux différents tests.

Décrire la stratégie globale de test:

- *types de tests et ordre dans lequel ils seront effectués.*
- *les moyens à mettre en œuvre.*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?).*
- *les testeurs extérieurs éventuels.*

2.5 Risques techniques

- *risques techniques (complexité, manque de compétences, ...).*

Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).

2.6 Planification

Révision de la planification initiale du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

3 Réalisation

3.1 Point de design spécifique

3.2 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

3.3 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.4 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

3.5 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique