

**Guerin Dewi SN2**

## **Rapport de stage**

**BTS SIO : SLAM**

**14 janvier/ 28 février**

Feldrise

18 Rue d'Isly, 35000 Rennes

Tuteur : Victor Denis

EPSI

Rue Fernand Robert, 35000 Rennes

2024/2025

1. Introduction.....	3
2. Présentation des missions. ....	3
a. FlashVote.....	3
b. WebiSport. ....	5
3. Démarche suivie pour effectuer ces missions.....	6
a. Première mission .....	6
b. Deuxième mission .....	16
4. Evaluations des réalisations et des compétences mobilisées. ....	17
5. Conclusion .....	19

## 1. Introduction

Pour commencer, j'ai effectué mon stage dans l'entreprise Feldrise, située dans le centre de Rennes, c'est une entreprise qui propose de l'accompagnement opérationnel, des conseils et des stratégies, ainsi que du développement d'application mobile et web.

Durant ce stage, deux missions m'ont été attribuées, la première avait pour but de développer un réseau social basé sur des sondages sous forme d'application mobile en utilisant Flutterflow, un outil de développement d'application mobile en No-Code. L'application avait besoin de plusieurs fonctionnalités telles que la création de sondages, l'ajout d'amis, la possibilité d'effectuer des recherches de profil, etc...

Ma deuxième mission était composée de plusieurs tâches différentes, l'entreprise travaille sur une application nommée Webisport, qui permet à des clubs de sport de gérer plus facilement leur organisation, en offrant la possibilité de gérer les convocations pour les matchs, les plannings d'entraînements, les différents événements. Les diverses tâches étaient la résolution de bugs, de l'optimisation ou bien des corrections visuelles. Cette deuxième mission avait pour but d'aider l'entreprise à effectuer des tâches chronophages afin d'avancer plus rapidement sur des points importants de ce projet. Ces deux missions ont été réparties sur 7 semaines de stage, allant du 14 janvier au 28 février.

Je vais dans un premier temps, présenter les missions plus en détails. Dans un second temps je présenterai les démarches suivies pour effectuer les missions, puis la réalisation des missions et enfin l'évaluation des réalisations et des compétences mobilisées.

## 2. Présentation des missions.

### a. FlashVote.

Lors de mon premier jour, une première mission m'a été confiée : celle de développer un réseau social basé sur des sondages, que les utilisateurs peuvent créer et y répondre, en utilisant Flutterflow pour la conception de l'application et FireBase pour la gestion des données. Avant de rentrer dans les détails de la mission, je vais présenter Flutterflow et FireBase.

Tout d'abord, Flutterflow est un outil de conception d'application simplifié en No-Code. Le but est de permettre à des débutants et des personnes n'ayant pas ou peu de compétences en code de tout de même avoir la possibilité de développer des applications. La conception se fait via un système de "widgets", ces "widgets" sont des blocs ayant différentes fonctions : un widget 'text' permet l'affichage de texte, un widget 'image', l'affichage d'une image, ceux-ci peuvent aller du champ de texte à remplir aux boutons. Le placement des widgets est très simple, il suffit de faire glisser sur la page le widget que l'on souhaite voir sur notre page, ils peuvent être rangés en colonne ou en ligne, et s'imbriquer les uns dans les autres. Bien qu'il y ait un grand nombre de fonctionnalités disponibles, il est possible créer ses propres fonctions en les codants soi-même en Dart, un langage de programmation fait pour les applications, offrant ainsi un plus large champ de possibilités. De plus, il est possible d'exporter son projet Flutterflow, sous un projet Flutter, et donc le passer en projet sous forme de code. Pour la gestion de données, FlutterFlow propose d'utiliser FireBase. C'est un outil d'aide au développement d'application web, Android, iOS et Unity, celui-ci propose un hébergement cloud et du NoSQL pour héberger des bases de données (le NoSQL est un modèle de stockage de données basé sur des tables indépendantes les unes des autres, au contraire du modèle relationnel). Dans le cadre de cette mission, Firebase aura servi principalement au stockage de données. De plus, FireBase dispose également d'un système d'authentification, utilisable au sein de notre projet Flutterflow, pour faciliter les gestions des comptes (connexion, déconnexion, inscription, etc..). L'utilisation de FireBase, plutôt qu'un autre outil de stockage, est justifiée par la présence d'un onglet dédié à FireBase dans FlutterFlow, permettant une configuration simple et une liaison entre l'application et la base de données qui est automatique.

La mission était donc, à l'aide de ces deux outils, de créer un réseau social basé sur des sondages, du nom de FlashVote. L'application devait avoir différentes fonctionnalités :

- La création de sondage, avec la possibilité de choisir le type de sondage (choix multiples et réponses uniques ou multiples, vrai/faux, oui/non, barre de pourcentage, étoiles, etc.).
- La possibilité de répondre aux sondages.
- Un système d'inscription et de connexion, mais aussi un accès 'anonyme', cet accès ne permet pas d'accéder à toutes les fonctionnalités de l'application.
- La possibilité d'ajouter des amis.
- La possibilité de voir ses statistiques (nombre de sondages auxquels on a répondu, le nombre de sondages créés, nos amis, etc.).

L'accès en mode anonyme n'autorise que la possibilité de répondre aux sondages, la création de sondages, l'ajout d'amis ou voir ses statistiques n'est pas disponible.

L'application devait être coupée en quatre grandes parties : la page d'accueil, sur laquelle s'affichaient les sondages créés par nos amis, une page de navigation, avec les sondages d'autres utilisateurs mis en public et une barre de recherches de profil. Une page de création de sondage, et notre page de profil.

## **b. WebiSport.**

La deuxième qui m'a été confiée était donc composée de plusieurs tâches différentes. Feldrise travaille sur une application mobile qui a pour but de proposer aux clubs de sport une solution de gestion de club, cette application dispose donc de plusieurs fonctionnalités :

- Système d'authentification, permettant d'identifier les membres d'un club.
- Gestion des matchs et des convocations, comprenant la création et l'édition des matchs, la gestion des joueurs convoqués pour les matchs et un suivi des présences/absences.
- Gestion des événements avec leur organisation, choisir s'ils seront répétés dans le temps ou non, la possibilité de publier des médias.

- Un système de communication avec une messagerie interne, un système de commentaires et des notifications.

Cette application étant un très gros projet, des bugs, plus ou moins gros, font leur apparition au cours du développement, j'avais donc pour missions d'en régler un maximum :

- Titre trop gros
- Fautes d'orthographe à divers endroits
- Cache non configuré pour certaines pages

Certains bugs devaient être corrigés au plus vite car ils pouvaient entraver l'utilisation de l'application ou gêner l'utilisateur.

## 3. Démarche suivie pour effectuer ces missions.

### a. FlashVote

Pour la première mission, toutes les technologies à utiliser étaient déjà choisies, je devais donc réfléchir à l'approche, la construction de l'application, les pages nécessaires, les tables qui allaient composer la base de données.

Pour pouvoir construire l'application efficacement j'ai établi l'ordre dans lequel j'allais procéder au développement : d'abord noter toutes les fonctionnalités. À partir de ça, j'ai pu réfléchir aux différentes tables de données, puis dresser la liste des pages nécessaires, et enfin passer au développement. La liste des fonctionnalités étant déjà faite, j'ai commencé à réfléchir aux données nécessaires à enregistrer, le nombre de tables, le contenu et le type des données stockées.

Tout d'abord, lorsque l'on crée notre projet FlutterFlow et que l'on décide de créer une base pour le projet, une table 'User' sera par défaut créée. De plus, FireBase implémente dans le projet FlutterFlow des fonctions de connexion, inscription et déconnexion, ce qui facilite grandement le développement et la gestion des profils. La table 'User' étant par défaut existante, on va pouvoir ajouter des champs nécessaires à nos besoins.

Users

Schema

Manage Content























Field Name	Data Type	Actions
email	String	 
photo_url	Image Path	 
created_time	DateTime	 
bio	String	 
friend	List < String >	 
notificationOn	Boolean	 
password	String	 
display_name	String	 
uid	String	 
phone_number	String	 
 New field name...	Data Type	<div> <div></div> <div>Is List</div> <div><input type="checkbox"/></div> </div> 

Table 'User' stockant les informations concernant les utilisateurs inscrit.

Dans cette table, nous pouvons voir les informations de base de la table 'User', tel que le mail, le pseudo, la date de création, le mot de passe et l'id. Comme je créais un réseau social, certains champs supplémentaires étaient nécessaires. J'ai donc rajouté un champ pour une photo de profil, une bio (qui s'affiche sur le profil), un booléen pour savoir si les notifications sont activées et un champ 'friend' typé 'list' permettant de stocker tous les id des amis que l'utilisateur ajouteras.

Cette table permet de gérer l'authentification d'un utilisateur mais également les informations de son profil et les liens qu'il a avec les autres utilisateurs.

Les sondages étant le cœur de FlashVote une table sondage était donc primordiale.

polls

Schema

Manage Content





















Field Name	Data Type	Actions
question	String	 
options	List < String >	 
category	String	 
creatorId	String	 
public	String	 
createdAt	DateTime	 
totalVotes	Integer	 
type	String	 
pollId	String	 
 New field name...	Data Type	<div> <div>Is List</div> <div><input type="checkbox"/></div> </div> 

Table 'polls', stockant les sondages.

Un sondage est composé tout d'abord d'une question et associé à un type (choix multiples avec réponses unique ou multiple, pourcentage, vrai/faux, etc.), le champ type permet donc d'identifier de quel type de sondage il s'agit, et ainsi permet d'afficher le format de réponses correspondant.

Bien que pour les sondages vrai/faux, oui/non n'ont pas d'options de réponses personnalisées, les sondages à choix multiples ont des choix qui sont définis par les utilisateurs, expliquant la présence du champ 'options'. Ce champ est typé 'list', car au moment de la création du sondage, on va enregistrer toutes les options créées par l'utilisateur dans une seule liste. Lorsqu'un utilisateur arrivera sur la page pour répondre à un sondage ayant des options personnalisées, il y aura juste à parcourir cette liste, et pour chaque objet dans cette liste, on le fait apparaître en tant qu'option de réponse, ce fonctionnement permet de créer des sondages très diversifiés et plus intéressants.

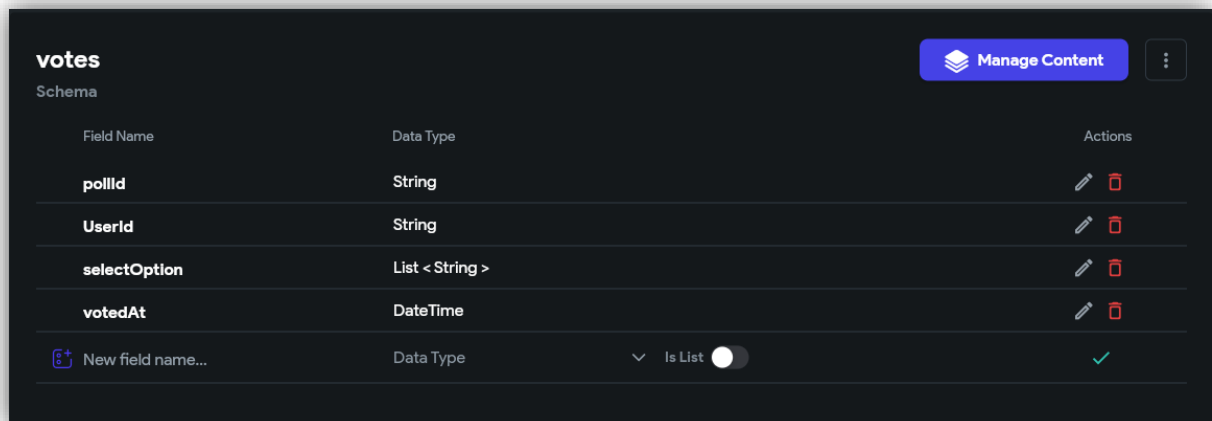
Un champ 'category' est également présent, permettant d'identifier quel type de question sera affiché, si c'est une question politique, social, informatique, etc.

Un champ 'creatorId' permet d'identifier l'auteur du sondage, pour gérer son affichage pour ses amis, qui l'auront en page d'accueil.



Enfin, un dernier champ, a utilité statistique est le champ 'totalVotes', comptant le nombre de fois où le sondage a été répondu.

Le stockage des sondages se faisant dans cette table, le stockage des votes, quant à eux, sont fait sur une autre table.













Field Name	Data Type	Actions
pollId	String	 
UserId	String	 
selectOption	List <String >	 
votedAt	DateTime	 
 New field name...	Data Type	 Is List <input type="checkbox"/>

Table 'votes' stockant toutes les réponses de sondages qu'il y a eu.

Cette table ne possède pas d'identifiant unique à cette table comme les autres, en effet, un vote est identifié par deux informations, un utilisateur et un sondage.

Lorsqu'un vote est envoyé, cette table va recueillir l'id de l'utilisateur qui l'a envoyé, et l'id du sondage qui a été répondu. On stocke également la ou les option(s) choisie(s) par l'utilisateur (le champ est sous forme de liste dans le cas où le sondage est à choix multiple et réponse multiples, on stocke toutes les options sélectionnées), et on garde également la date et l'heure où le vote a eu lieu.

Il reste deux autres tables, 'comments' et 'notifications'












comments			
Schema			
Field Name	Data Type	Actions	
pollId	String		
userId	String		
content	String		
createdAt	DateTime		
 New field name...	Data Type	<div>  <div>Is List <input type="checkbox"/></div> </div>	

Table 'comments' stockant les commentaires laissés sous un sondage

La table 'comments' va permettre de stocker les commentaires laissés sous un sondage par une personne. On va enregistrer l'utilisateur qui a créé le commentaire et le sondage sous lequel il a été déposé, pour l'afficher dans la section commentaire du sondage en question. On stocke également son contenu et sa date de création.














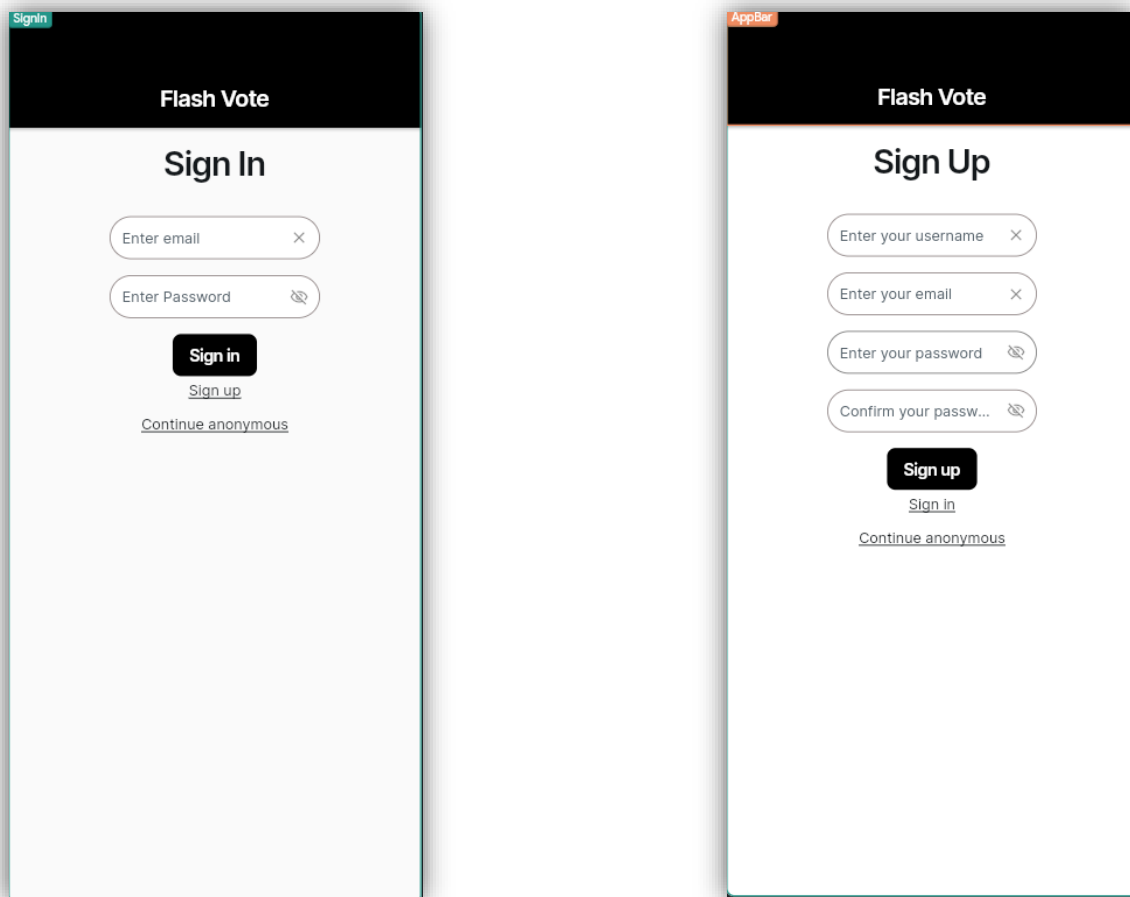
notifications			
Schema			
Field Name	Data Type	Actions	
userId	String		
type	String		
message	String		
createdAt	DateTime		
read	Boolean		
 New field name...	Data Type	<div>  <div>Is List <input type="checkbox"/></div> </div>	

Table 'notifications' stockant les notifications envoyées aux utilisateurs

Dans cette table, on stocke le type de notifications, si c'est quelqu'un qui nous ajoute en amis, une réponse à un de nos sondages. On stocke le message affiché, l'id de l'utilisateur qui l'a reçu et si la notification a été lue.

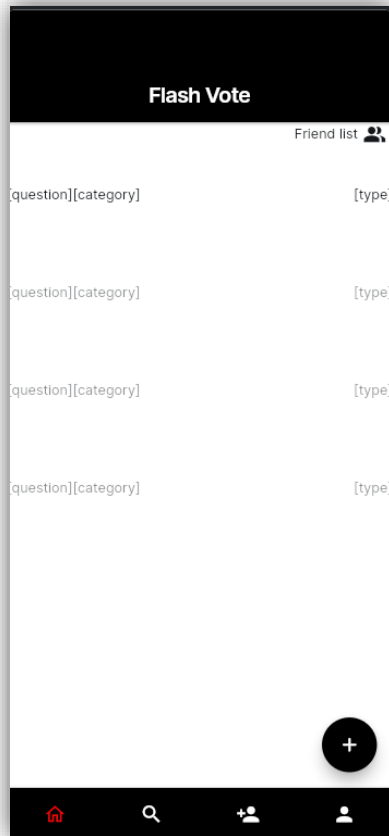
Une fois que toutes les tables ont été créées comme il faut, j'ai pu commencer à dresser la liste des pages et leur fonctionnalité. Tout d'abord, lorsque l'on arrive sur l'application, on est envoyé sur la page de connexion.



Page connexion et de la page inscription.

Par défaut, au premier lancement de l'application, nous sommes redirigés vers la page 'connexion', sur celle-ci on peut se connecter, si nous n'avons pas de compte, on peut aller sur la page d'inscription.

Les pages sont des pages d'authentification basique, cependant, il y a la possibilité de parcourir l'application en anonyme, la seule option est de répondre au sondage, toutes les autres fonctionnalités telles que l'ajout d'amis, la visualisation de son profil et la création de sondage sont impossibles. Une fois la connexion, ou le passage en mode anonyme d'effectué, nous somme redirigés vers la page d'accueil.



Page 'home'

Sur cette page, nous avons accès à la totalité des fonctions.

Pour commencer on peut aller sur notre liste d'amis en cliquant sur 'Friend list' en haut à droite. Depuis cette liste d'amis, on peut visualiser leur profil.

Sur le reste de la page s'affichent les sondages créés par nos amis, on voit la question (qui remplace [question]), la catégorie (politique, social, etc.) et le type de sondage (étoiles, pourcentage, oui/non, etc.), permettant à l'utilisateur de répondre ou non au sondage, en fonction de s'il le trouve intéressant ou non. Lorsque l'on clique sur un sondage, on est redirigé vers la page où l'on peut y répondre.

Au-dessus de l'app bar, un bouton flottant avec un '+' est présent, celui-ci permet de créer et poster un nouveau sondage.

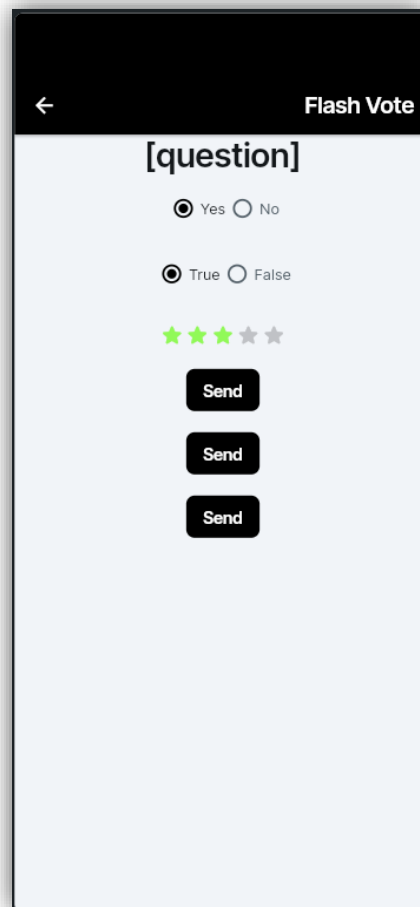
Sur l'app bar, on peut voir quatre boutons, un premier permettant d'aller à la page d'accueil, le deuxième permet l'accès à la page 'explorer', qui affiche des sondages mis en public par des utilisateurs, on peut donc voir les sondages privés ou publics de nos-

Le troisième bouton affiche une liste d'utilisateurs, que l'on peut ajouter ou non en tant qu'amis, et enfin, le dernier bouton permet l'accès à son profil.

Page de création d'un sondage.

Par défaut, on peut voir plusieurs widget ‘textfield’ à remplir, ceux-ci disparaissent si le type de de sondage est différent de choix multiple. Lorsque c’est un choix multiple, on a la possibilité d’ajouter ou de supprimer une option de réponse, le minimum étant deux.

Un dernier menu déroulant se situe en bas de la page, permettant au créateur du sondage de choisir si le sondage est public ou privé. S'il est privé, il apparaîtra seulement sur la page "home" de ses amis, s'il est public, il apparaîtra sur la page home de ses amis, mais il pourra aussi apparaître sur la page 'explorer' d'autres utilisateurs. Pour répondre à un sondage, nous sommes redirigés vers une page dédiée, l'affichage d'un sondage sur cette page va dépendre donc du type de sondage.

The image shows a mobile application interface for a 'Flash Vote' poll. At the top, there is a black header bar with a white back arrow on the left and the text 'Flash Vote' on the right. Below the header, the question '[question]' is displayed in a bold, black font. Underneath the question, there are two sets of radio buttons: the first set has 'Yes' and 'No' options, and the second set has 'True' and 'False' options. Below these options is a five-star rating system, with the first three stars highlighted in green and the last two in grey. At the bottom of the screen, there are three black buttons with the word 'Send' in white text, stacked vertically.

Page de réponse à un sondage.

Lorsque l'on décide de répondre à un sondage, nous sommes redirigés sur cette page. Sur cette image, étant la vue de développement, on voit plusieurs boutons d'envoi et de moyen de répondre. Au chargement de cette page, l'application va regarder le type du sondage choisi et afficher seulement le moyen de répondre y correspondant et son bouton d'envoi associé. Lorsque l'envoi est confirmé, les informations de vote sont enregistrées dans la table 'votes', et on incrémente d'un le nombre total de vote pour ce

sondage dans la table 'poll'. C'est la seule fonctionnalité qu'un utilisateur anonyme peut utiliser, avec la page explorer.

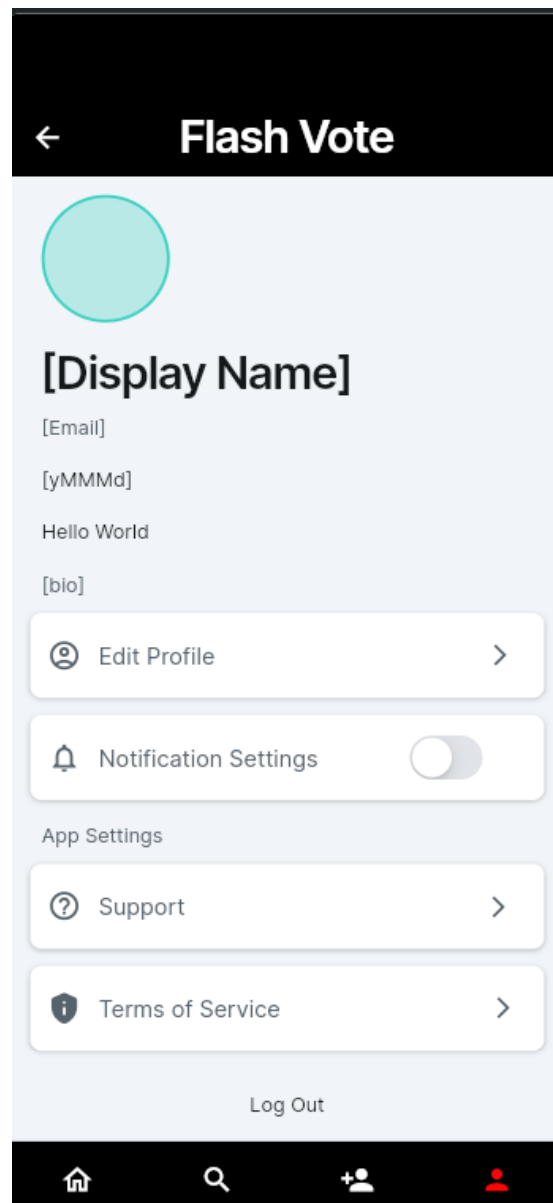


Image de la page profil.

Cette page n'est accessible que si l'on est connecté, tenter d'y accéder en anonyme renvoie automatiquement sur la page de connexion. Elle permet à un utilisateur de visualiser et de gérer les informations de son profil, on peut y voir sa photo de profil, son pseudo, son adresse mail et la date de création (visible seulement par l'utilisateur) et la bio. L'utilisateur peut, s'il le souhaite, modifier les informations de son profil, activer ou désactiver les notifications ou bien se déconnecter. Dans la partie 'edit profil', il a la

possibilité, en plus de pouvoir modifier les informations de son profil, de supprimer son compte avec toutes les données liées à ce compte, telles que les sondages qu'il a créés.

Cette première mission n'a pas été terminée, le but était de m'initier au NoSQL, avec FireBase et de pouvoir découvrir des outils de No-code et pouvoir comparer les limites avec du code basique. Mon tuteur voulait également un retour sur FlutterFlow et ce que j'en avais pensé. Bien que le No-code soit très pratique, accessible et permette de faire des applications efficacement et facilement, on peut vite être limité par les fonctionnalités. De plus, j'ai pu constater quelques bugs, surtout visuels. Par exemple, lorsque j'affichais l'application en test, les couleurs ne correspondaient pas à ce que j'avais configuré et empêchaient une utilisation confortable.

Cependant, il est possible d'exporter son projet, cette fois en code, et donc de continuer ce que l'on avait commencé en codant, et ne plus être restreint par le no-code.

### a. Webisport

J'ai travaillé 3 à 4 semaines sur la première mission, le reste du stage, on m'a confié une autre mission qui était de corriger des bugs sur l'application Webisport. L'application étant déjà disponible sur les stores tels que PlayStore et AppStore, étant donc très avancée, j'ai d'abord dû comprendre l'organisation des fichiers, le fonctionnement, le code et toute la structure de l'application.

Les premiers jours étaient compliqués, des tâches très simples m'ont été attribuées, comme régler des problèmes de style à divers endroits, pour que je puisse prendre en main le code, et m'habituer à la structure.

L'application est codée en Flutter, et les données sont stockées sur une base de données Wordpress, et récupérées via des routes API, que l'on appelle dans le projet. Autres que des problèmes de styles, j'ai également pu faire des tâches d'optimisation. Afin d'éviter, à chaque rafraichissement de page, de faire une nouvelle requête pour récupérer des données et donc d'augmenter les temps de chargement, celles-ci sont mises en cache pendant une durée prédéfinie, permettant ainsi un chargement plus rapide. Or, le cache n'était pas configuré sur toutes les pages, j'ai donc dû ajouter du cache sur les pages où il n'y en avait pas, en comparaison avec ce que j'avais pu faire en cours en Flutter, la difficulté de la tâche était bien plus élevée. De plus, lorsque certaines



modifications de données étaient faites, le cache restait inchangé et la donnée affichée n'était donc pas des données à jour, il fallait donc, dans le cas où certaines données sont modifiées par un utilisateur, supprimer le cache, afin de pouvoir recharger la page avec les nouvelles données, et les mettre en cache.

En dehors de ces deux tâches principales, mon tuteur m'a expliqué et montré diverses choses, comme le processus de publication d'une application sur les deux stores, AppStore et PlayStore, montrant ainsi les différences propres à chaque store. Il m'a également montré différentes manières d'organiser les fichiers de son projet Flutter, chaque manière ayant ses avantages et inconvénients. Par exemple, il y a la structure par fonctionnalité, où chaque fonctionnalité de l'application aura son propre dossier, ou bien la structure par type de fichier où les fichiers seront séparés en fonction de leurs rôles. Le premier est très utile pour les gros projets, au contraire du deuxième, qui est plus utile pour les petits projets. Il m'a également expliqué comment bien gérer un projet, allant de l'approche à avoir lorsque l'on commence un projet.

- Brainstorming avec les clients.
- Rédaction d'un cahier des charges.
  - Liste des fonctionnalités.
  - UI attendu
  - UX attendu

## 4. Evaluations des réalisations et des compétences mobilisées.

La première mission que j'ai effectuée n'apportait pas directement de valeur à l'entreprise, elle avait pour principal objectif de me familiariser avec un outil de création d'applications et de m'initier au NoSQL. Le but était également que je donne un avis sur cet outil à l'entreprise. En revanche, les tâches réalisées dans le cadre de la deuxième mission étaient utiles à l'entreprise, puisqu'elles ont permis de corriger des bugs et d'apporter des améliorations techniques à l'application et ainsi leur faire

gagner du temps, et approfondir mes compétences. Ainsi, la première partie de mon stage était essentiellement axée sur mon apprentissage et le développement de mes compétences, tandis que la seconde alliait à la fois un apport concret pour l'entreprise et un renforcement de mes compétences techniques. Durant ce stage, j'ai pu tout d'abord approfondir mes connaissances en Flutter. Durant l'année, nous avons eu des cours sur le développement d'application mobile. Cela restait cependant les bases : développer une application simple, sans backend et fonctionnalités compliquées, essentiellement de la mise en page et comprendre comment le langage fonctionne. Avoir travaillé sur une application aussi aboutie que Webisport m'a apporté de nouvelles connaissances sur ce langage, et cela m'a aussi permis de comprendre comment optimiser une application. J'ai également pu découvrir l'organisation d'un projet de cette envergure, comment les tâches sont attribuées, comment elles sont vérifiées puis implémentées à la version disponible de l'application. La deuxième mission m'a permis de m'améliorer sur la compréhension et le développement d'applications en Flutter. Je suis arrivé au stage avec seulement de très simples bases dans ce langage, cependant, travailler sur ce projet m'a permis de me familiariser à ce langage, d'abord dans la compréhension du langage, puis dans le développement. À force de lire le code et d'essayer de le comprendre, j'ai fini par assimiler sa logique et sa structure globales. Puis j'ai commencé à travailler sur des tâches simplistes telles que la correction de problèmes de style, puis j'ai pu passer à des tâches plus complexes et finir par travailler sur de l'optimisation de l'application. J'ai pu constater une amélioration de mes compétences par rapport à mon arrivée dans l'entreprise. Autre que des compétences en développement, j'ai pu voir l'organisation et la gestion de projet, allant du listage des tâches à leur attribution aux membres de l'équipe.

Pour m'améliorer en tant que Concepteur-Intégrateur DevOps, il me manque tout d'abord de l'expérience dans le métier, qui viendra au fur et à mesure du temps. Il me manque également des connaissances dans les bonnes pratiques du développement, telle que la gestion de projet.

## 5. Conclusion

Lors de mon stage, j'ai été accueilli dans les bureaux de Feldrise, qui se situe dans un espace de coworking aux Trois Soleils. L'espace est un très bon endroit pour travailler. De plus, mon maître de stage était très disponible dans les moments où j'avais des questions, et s'il n'était pas présent, je pouvais me tourner vers son alternant, qui m'a beaucoup aidé aussi.

J'ai trouvé ce stage très enrichissant.

Au niveau professionnel : ça m'a permis de voir et comprendre comment une entreprise peut fonctionner (échanges entre client et prestataire, travail d'équipe, brainstorming), étant donné que l'année dernière, j'ai effectué mon stage seul (en distanciel).

Et au niveau technique : ce stage m'a permis de découvrir de nouveaux outils, tels que FlutterFlow et FireBase et de nouvelles technologies, comme le NoSQL, mais j'ai également pu approfondir et améliorer mes compétences en Flutter.

Un point plus théorique que j'ai trouvé très intéressant est la gestion de projet au sein de cette entreprise (Outil de gestion de projet : Notion, avec une roadmap, objectifs à atteindre).

Pour mon prochain stage, je souhaiterais m'orienter plutôt vers une entreprise axée sur la sécurité. Étant donné que je souhaite travailler dans ce domaine, je trouve donc pertinent de faire un stage dans ce milieu. Cela pourrait me permettre de comprendre comment la sécurité est abordée et gérée dans le milieu professionnel, ainsi qu'améliorer mes compétences.