



Avenue du Ciseau, 15  
1348 Louvain-la-Neuve

# Développement d'une application téléphone pour la 10ème Unité du Divin Sauveur en React Native

---

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en Informatique et Systèmes  
orientation Technologie de l'Informatique

## Aymeric PONJÉE



Rapporteur : Jonathan NOËL

**Année Académique 2021 – 2022**

## **Remerciements**

Je tiens tout d'abord à remercier les chefs d'unité pour avoir tenu le rôle de client pour le projet de mon TFE, ils m'ont apporté une grande aide sur plusieurs points tels que le design ou autres idées pour mon application.

Je remercie aussi tous les chefs de mon unité, la 10ème Divin Sauveur pour avoir donné leurs avis sur des détails importants et de m'avoir motivé avec leur enthousiasme quand ils ont appris le sujet de mon TFE.

Je remercie également mon rapporteur, monsieur Noël, pour son suivi et ses conseils, tout au long de mon TFE.

## Table des matières

<b>Remerciements.....</b>	<b>1</b>
<b>Introduction .....</b>	<b>4</b>
1. <b>Mise en contexte.....</b>	<b>4</b>
2. <b>Description du projet.....</b>	<b>5</b>
3. <b>Objectifs du projet.....</b>	<b>6</b>
4. <b>Contraintes.....</b>	<b>6</b>
<b>Présentation du contexte de travail.....</b>	<b>7</b>
1. <b>Technologies .....</b>	<b>7</b>
L'environnement informatique .....	7
L'environnement de développement .....	7
Node JS .....	7
React Native.....	7
Postman.....	7
Yarn.....	8
MongoDB.....	8
Trello .....	8
Gestion du temps.....	8
GitHub .....	8
2. <b>MERN Stack .....</b>	<b>9</b>
Front End.....	9
Back End.....	9
Base de données.....	10
Mon choix .....	10
<b>Méthodologie.....</b>	<b>11</b>
Exemple d'user story .....	11
<b>Étapes du projet.....</b>	<b>12</b>
1. <b>Réunion client .....</b>	<b>12</b>
2. <b>Analyse.....</b>	<b>12</b>
3. <b>Implémentation .....</b>	<b>12</b>
4. <b>Tests .....</b>	<b>12</b>
5. <b>Modification selon appréciation du client .....</b>	<b>13</b>
<b>Application.....</b>	<b>13</b>
1. <b>Structure de fichiers .....</b>	<b>13</b>
2. <b>Diagramme .....</b>	<b>14</b>
3. <b>Schéma application .....</b>	<b>15</b>
4. <b>Connexion/Inscription.....</b>	<b>15</b>
Front End.....	15
Back End.....	15
Design.....	15
5. <b>Calendrier .....</b>	<b>16</b>
Front End.....	16
Back End.....	16
Design.....	16

<b>6. Boutique.....</b>	<b>17</b>
Front End.....	17
Design.....	17
<b>7. Salon de conversation .....</b>	<b>17</b>
<b>Recommandations.....</b>	<b>18</b>
<b>Conclusion.....</b>	<b>19</b>
<b>1. Résumé .....</b>	<b>19</b>
<b>2. Problèmes rencontrés.....</b>	<b>19</b>
<b>3. Amélioration .....</b>	<b>20</b>
Connexion .....	20
Inscription.....	20
Calendrier.....	20
Boutique / Conversation .....	20
Amélioration générale .....	20
<b>4. Prolongement du travail.....</b>	<b>21</b>
<b>Code Source.....</b>	<b>22</b>
<b>1. Code source : GitHub.....</b>	<b>22</b>
<b>2. Trello .....</b>	<b>22</b>
<b>3. ClockiFy .....</b>	<b>22</b>
<b>Bibliographie.....</b>	<b>23</b>
<b>1. React Native .....</b>	<b>23</b>
<b>2. MangoDB .....</b>	<b>23</b>
<b>3. Stack MERN .....</b>	<b>23</b>
<b>4. Scrum .....</b>	<b>23</b>
<b>5. Yarn .....</b>	<b>23</b>
<b>6. Node.js .....</b>	<b>23</b>
<b>7. Express .....</b>	<b>23</b>
<b>8. Postman.....</b>	<b>23</b>
<b>Annexes.....</b>	<b>24</b>

## Introduction

Dans le but de finaliser mon bachelier au sein de l'**EPHEC**, j'ai réalisé un **TFE** pour mon unité scoute, la 10<sup>ème</sup> Divin Sauveur.

Durant ce projet, j'ai repris l'ensemble des connaissances et concepts accumulés lors de mes années d'apprentissage. En plus de tous les sujets abordés durant mes études en matière d'analyse, d'architecture de code et de gestion de projet, j'ai pu découvrir et pratiquer de nouveaux éléments dans le domaine de l'informatique tels que la découverte et la mise en place de nouveaux outils de travail, patterns, architecture et de Framework JavaScript.

### 1. Mise en contexte

Tout d'abord, j'aimerais mettre en contexte le domaine du scoutisme, car tout le monde ne connaît pas spécialement celui-ci, ni les termes utilisés.

Le scoutisme est le premier mouvement éducatif au monde. Il repose principalement sur l'apprentissage de plusieurs valeurs telles que la **solidarité**, **l'entraide**, le **respect** et bien d'autres valeurs. Dans ce mouvement, nous avons la fédération scoute, comportant une équipe de plusieurs membres nommés en tant que délégués d'un certain nombre d'unités présentes sur le territoire. Cette équipe est en charge de l'administratif, des changements de lois et des codes moraux du mouvement, de la transmission des informations aux unités, ...

Chaque unité dispose d'une équipe de minimum 3 membres, dont le but est de soutenir les animateurs et coordonne les actions éducatives entreprises au sein des différentes unités. Cette équipe a donc joué le rôle du client lors de mon TFE.

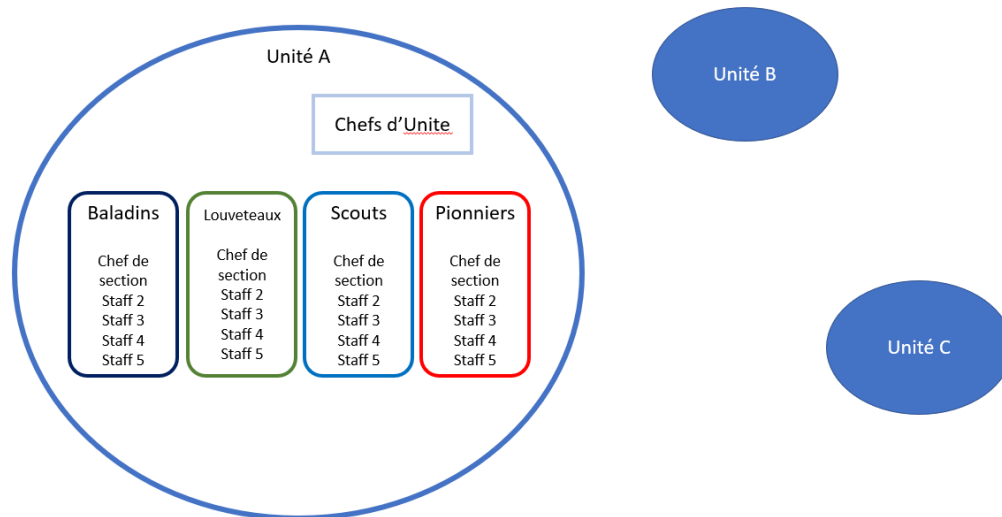
Pour bien situer le projet, nous nous ancrons dans notre unité, dans laquelle nous retrouvons des sections qui sont les groupes répartis par tranches d'âges des enfants/animés :

- Baladins (6-8 ans)
- Louveteaux (8-12 ans)
- Éclaireurs/Scouts (12-16 ans)
- Pionniers (16-18 ans)

Toutes les sections sont animées par des **bénévoles** qui sont appelés « chefs/staffs ». Dans chaque équipe de staffs nous retrouvons un chef de section qui est souvent le plus expérimenté du groupe. Lors de la réalisation de mon TFE, les chefs de mon unité ont pu donner leurs avis/idées.

Voici un petit schéma qui représente l'organisation que nous venons de présenter.

Cet ensemble forme donc la **fédération**.



## 2. Description du projet

Dans le cadre de mon TFE, j'ai décidé d'associer deux choses qui me tenaient à cœur, **l'informatique** et le **scoutisme**. C'est pour cela, que j'ai décidé de créer **une application mobile** pour mon unité, la 10ème Divin Sauveur.

En effet, depuis que je suis chef scout, nous avons toujours su faire avec les moyens que nous disposons, mais plus les années passent, plus notre unité grandit et plus les informations sont difficiles à trouver. Dans l'unité, nous faisons face à plusieurs problèmes. Il y a un important manque de communication, que ce soit avec les parents, les enfants ou même entre chefs, les endroits de camp sont de plus en plus difficiles à trouver, les gens cherchent énormément pour avoir des documents, photos ou autres. Plus l'unité grandit, et plus c'est dur à gérer.

C'est pour cela que j'ai décidé de tout regrouper en une seule application pour la réalisation de mon TFE ! La décision de faire une application mobile plutôt qu'un site, a été prise en unité. En effet, il est beaucoup plus facile de diffuser ou modifier les informations de dernière minute sur une application mobile plutôt que sur un site. Par exemple, prendre les présences sur place, envoyer un message à la suite d'un imprévu ou autre.

Cette application s'adressera dans un premier temps aux chefs actuels pour permettre une meilleure organisation au sein de l'équipe d'unité, puis, dans le futur, intégrera les parents/enfants.

### 3. Objectifs du projet

Le principal objectif de mon TFE, est de mettre en œuvre **une application mobile fonctionnelle**, contenant le plus de fonctionnalités utiles, dans un temps imparti de 4 mois.

Ce projet a pour but de regrouper toutes les applications utilisées par mon unité scout, telles que Slack, Messenger, Gmail, AtoutsCamp, Facebook, VotreCamp, ... en une seule application spécifique à l'unité. De plus, le design de l'application devra rester **sobre et facile d'utilisation** pour permettre à un maximum de personnes, jeunes comme âgées, d'utiliser l'application sans trop de soucis.

Malgré toutes les fonctionnalités à implémenter, j'ai décidé d'en sélectionner quelques-unes pour rester dans les temps imposés par l'EPHEC :

- Implémenter une page **d'inscription** et de **connexion**.
- Créer un **calendrier** affichant tous les événements futurs, avec une **liste de présence** par événement.
- Afficher une **liste d'articles** mis en vente par l'unité.
- Implémenter un **salon de conversation** pour diffuser les informations de dernière minute.

Avant d'entamer le projet, Il m'a fallu faire une analyse claire de la demande du client. La défense intermédiaire et les retours de mon rapporteur m'ont bien aidé à cerner les différents points du projet. De plus, j'ai réalisé un Trello avec plusieurs objectifs à réaliser, avec des deadlines pour m'assurer de la bonne compréhension de la demande.

### 4. Contraintes

Un projet informatique est souvent soumis à des contraintes. Cette rubrique énumérera toutes les contraintes soumises par le client et moi-même.

1. L'application développée est la propriété exclusive de l'unité du **Divin Sauveur**.
2. L'application doit rester en interne à l'unité.
3. Fournir une application dynamique adaptée à tout type d'écran mobile. L'affichage doit être sobre et facile d'utilisation pour une personne de tout âge.
4. Le code doit être écrit de sorte que l'on puisse facilement ajouter ou supprimer des éléments ou des produits dans le futur.

# Présentation du contexte de travail

## 1. Technologies

### L'environnement informatique

Ma machine personnelle a servi d'outil de développement pour ce projet. Je disposais d'un ordinateur portable Apple, ce qui m'a permis d'être assez flexible dans mon travail.

Concernant les logiciels utilisés pour le développement, Visual Studio Code était l'éditeur de texte utilisé pour mon projet. Ce logiciel populaire offre une grande panoplie de fonctionnalités et d'extensions pratiques pour le développement, une console de débogage, un terminal et plein d'autres avantages.

### L'environnement de développement

J'ai décidé d'utiliser Xcode, qui est l'environnement de développement Apple, permettant de développer des applications. Il m'a permis de créer l'application et, dans le futur, de publier celle-ci sur l'App Store.

### Node JS

Node.js est un moteur d'exécution JavaScript asynchrone piloté par événement. Il a été conçu par Facebook pour créer des applications réseau évolutives. Il présente une boucle d'événement comme une construction d'exécution plutôt que comme une bibliothèque.

### React Native

React Native est le langage de programmation que j'ai utilisé pour coder mon application mobile. Celui-ci est un Framework JavaScript développé par Facebook en 2015, et est totalement Open Source.

React Native permet de réaliser des applications natives. Grâce à cela, il permet la création d'un seul code source JavaScript pour construire deux applications, au lieu de les développer séparément. React Native crée donc une application pour iOS et une application pour Android.

C'est pour cela que j'ai choisi ce langage, car il offre un gain de temps assez important puisqu'il divise la charge du travail en deux.

### Postman

Postman est une application permettant de tester des API, créée en 2012 par Abhinav Asthana. Elle regroupe chaque test d'API dans une collection, permettant de mutualiser leurs URL et authentifications. Postman a servi d'outil pour tester toutes les routes créées dans mon application.



### Yarn

Yarn est un gestionnaire de dépendances **rapide**, **fiable** et **sécurisé**. Il utilise un algorithme déterministe pour les installations, Yarn est en mesure de garantir qu'une installation qui a fonctionné sur un système fonctionnera exactement de la même manière sur un autre. Ce gestionnaire de dépendances sera utilisé par la majorité de mon projet.

### MongoDB

Au niveau de la base de données, j'ai décidé d'utiliser MongoDB. Apparue au milieu des années 2000, MongoDB est une base de données NoSQL orientée document. Elle fournit des fonctionnalités de sécurité telles que l'authentification, le contrôle d'accès et le cryptage.

Le fait que j'ai choisi MongoDB pour la réalisation de mon TFE est qu'elle utilise des documents qui n'ont pas de schéma prédéfini avec des champs qui peuvent être ajoutés à volonté. Les modèles de données disponibles au sein de MongoDB permettent de représenter des relations hiérarchiques ou autres structures complexes plus facilement. Ce qui la rend très **flexible**.

Le monde du scoutisme est en continuel changement, ce qui me permet d'ajouter/changer/supprimer plus facilement des champs en cas de besoin.

### Trello

Trello a été utilisé pour la gestion des tâches de mon TFE. Celui-ci est un outil visuel qui permet de gérer n'importe quel type de projet.

Trello m'a beaucoup aidé lors de mon projet, il m'a permis de séparer mes tâches en plusieurs points, de fixer des objectifs avec des deadlines précises, d'avoir une idée plus concrète pour chaque étape de mon projet.

### Gestion du temps

Pour la gestion du temps, J'ai principalement utilisé Clockify et Google Calendar.

Clockify est une application gratuite de suivi de temps. Il m'a permis d'évaluer le temps donné sur une implémentation ou un problème réalisé lors mon TFE. Ce qui m'a permis, par la suite, de beaucoup mieux évaluer mon temps de travail.

Google Calendar, est un agenda dans lequel je note tous mes rendez-vous ou activités personnelles. Cette application m'a énormément servi d'un point de vue organisationnel, car il m'a permis de noter à l'avance quand j'allais travailler sur mon TFE et non l'inverse, lorsque j'avais un peu de temps sur ma journée.

### GitHub

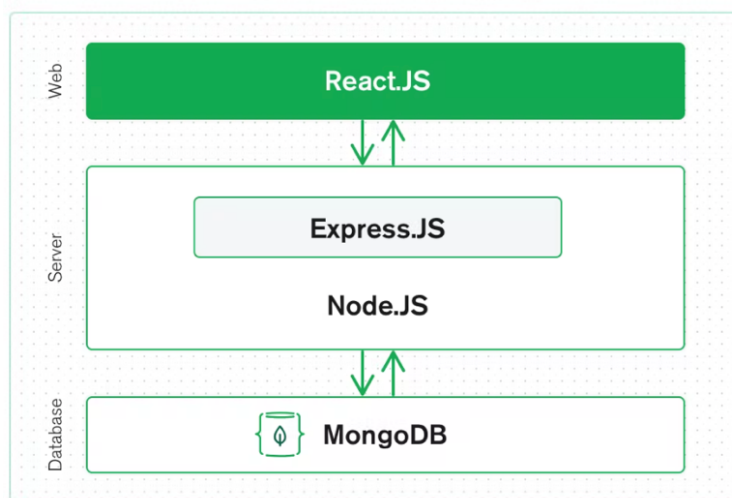
GitHub est une service web d'hébergement et de gestion de développement de logiciels. Celui-ci m'a permis de stocker mon code facilement. De plus, GitHub m'a permis un gain de temps à la suite de plusieurs crashes de l'application, me permettant de faire un backup rapidement.

## 2. MERN Stack

La pile **MERN** signifie **M**ongoDB, **E**xpress, **R**ead et **N**ode, d'après les quatre technologies clés qui composent la pile.

- MongoDB - Base de données de documents.
- Express(.js) - Framework Web Node.js.
- React(.js) - Framework JavaScript côté client.
- Node(.js) - Premier serveur Web JavaScript.

L'architecture MERN permet de construire facilement une architecture à trois niveaux (Front end, Back end, base de données) entièrement en JavaScript et JSON.



### Front End

React.js, dans le cas de mon TFE, React Native permet de créer des interfaces complexes grâce à des composants simples, et de les connecter aux données de votre serveur principal. Son point fort est la gestion d'interfaces dynamiques et d'être basé sur les données avec un minimum de code. C'est un excellent support pour tout ce qui est formulaires, gestion des erreurs, événements, listes, ...

### Back End

Le niveau serveur Express.js est exécuté à l'intérieur d'un serveur Node.js. Il dispose de modèles puissants pour le routage d'URL et gère les requêtes et les réponses HTTP.

En effectuant des requêtes HTTP ou des requêtes GET ou POST à partir de votre Front End, vous pouvez vous connecter aux fonctions de Express.js qui alimentent votre application. Ces fonctions, à leur tour, utilisent les pilotes Node.js de MongoDB, soit via des rappels, soit à l'aide de promesses, pour accéder et mettre à jour les données de votre base de données MongoDB.

### Base de données

MongoDB est tout aussi facile à utiliser que React Native, Express et Node. Les documents JSON créés dans votre Front End peuvent être envoyés au Back End, où ils peuvent être traités et stockés directement dans MongoDB.

### Mon choix

La stack MERN est une combinaison de données JSON qui circulent facilement d'avant en arrière, ce qui les rend relativement rapides à développer et plus ou moins simples à déboguer.

De plus, il ne suffit que d'apprendre un langage de programmation et la structure du document JSON pour comprendre tout le système, ce qui m'a permis un gain de temps énorme et une évolution rapide pour la réalisation de mon TFE.

## Méthodologie

Mon TFE consiste à créer une application mobile pour améliorer la communication au sein de mon unité. Lors de la réalisation de mon TFE, les chefs d'unité joueront le rôle de client. Les chefs des différentes sections donneront également leurs avis pour avoir un maximum d'idées. Les idées reçues seront centralisées et rediscutées avec le client pour savoir si elles sont importantes à implémenter ou non.

Durant le projet, j'ai décidé d'utiliser la méthodologie **Scrum**, qui, bien que je sois seul sur le projet, m'a permis de m'organiser correctement. **Scrum** est la **méthode agile** la plus utilisée dans les projets informatiques. Elle permet au client du projet d'être le principal pilote de l'équipe de développement.

L'avantage de cette méthodologie est d'aboutir rapidement à une fonctionnalité utilisable. Validés au fur et à mesure par le client, les sprints suivants permettent d'enrichir les étapes précédentes.

La méthode Scrum était idéale pour mon projet, car elle m'a permis de faire un sprint review toutes les deux réunions (plus ou moins tous les mois).

À la suite des diverses fonctionnalités à réaliser lors de ce travail, j'ai décidé d'utiliser l'outil Trello, ce qui m'a permis de bien cerner le projet étape par étape, en me fixant un ordre de priorité et des objectifs. Ces étapes sont appelées « user stories ».

### Exemple d'user story

En tant qu'utilisateur, j'aimerais une page calendrier afin que je puisse voir quand se déroule les réunions/événements durant l'année (2).

Condition pour terminer l'user story :

- Afficher la liste des réunions.
- Lier la base de données à l'application.
- Une fois l'évènement passé, il ne s'affiche plus.
- Un filtre a été mis en place.
- Design validé par le client.

Un screen de la page est mis pour pouvoir la montrer au client au cas où.

Un ordre de priorité a été établi avant que le projet commence avec un échéancier personnel pour ne pas s'éparpiller et savoir si les pages sont validées dans les temps ou non.

## Étapes du projet

Durant mon projet, j'ai suivi plusieurs étapes qui m'ont permis de mener à bien mon projet sans griller les étapes.

1. Rencontre avec le client pour pouvoir cerner les besoins en fonction du temps imparti.
2. Création d'une analyse du projet comprenant : Idées concrètes, design et schéma de base de données.
3. Implémentation du projet.
4. Réalisation de tests en fonction des scénarios.
5. Modifications en fonction des satisfactions du client.
6. Finalisation du projet.

Pour gérer toutes ces tâches, Trello m'a beaucoup aidé, car j'ai pu structurer le projet en mettant des deadlines, des objectifs, un suivi et bien d'autres choses.

### 1. Réunion client

Avec la fin du Covid-19, les activités scoutées ont pu reprendre normalement, ce qui m'a donné la possibilité de faire un feedback régulier à mes clients (plus ou moins tous les mois) et deux fois sur le quadrimestre une démo de l'application.

### 2. Analyse

Cette étape a été très importante pour moi car, quand j'ai présenté mon sujet, j'avais plein d'idées avec plein de fonctionnalités à implémenter. Cependant, c'était beaucoup trop ambitieux pour le temps imparti. L'analyse m'a permis de mieux évaluer la charge de travail.

De plus, elle m'a permis de réfléchir comment j'allais implémenter l'application avec des maquettes design, ma base de données avec son infrastructure et ma structure de fichiers préparés à l'avance.

### 3. Implémentation

Au niveau de l'implémentation, j'ai décidé de programmer page par page pour éviter de partir dans tous les sens. De plus, en fonctionnant comme ceci, je pouvais mieux évaluer le temps d'implémentation par la suite. Par exemple, si je prenais 20h pour la page connexion, je pouvais estimer à plus ou moins 20h le temps pour la page inscription.

### 4. Tests

Une fois la page implémentée, j'ai réalisé plusieurs tests pour pouvoir rendre l'application la plus optimale possible.

## 5. Modification selon appréciation du client

Une fois la page implémentée, nous avons fait une réunion avec le client pour déterminer si cette page correspondait à ses besoins. Sinon, je modifiais la page en fonction des remarques.

## Application

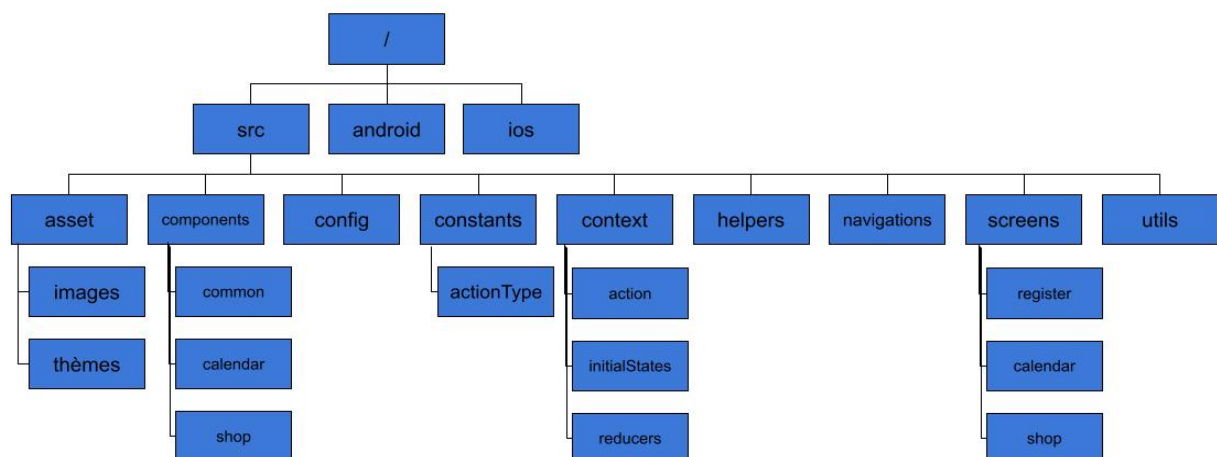
Le principal objectif du projet était d'implémenter **une application mobile** avec un maximum de fonctionnalités utilisables dans un temps imparti. L'application se verra évoluer même après la période du TFE terminé.

Dans un premier temps, nous allons découvrir la structure de fichiers et le diagramme utilisé pour l'application.

Par la suite, le schéma de l'application. Avec les différentes possibilités sur une page.

Pour finir, un descriptif de la page avec un rendu visuel.

### 1. Structure de fichiers



**Src** : Répertoire contenant l'ensemble du code source lié à l'application.

**Android** : Répertoire contenant l'ensemble des configurations liées à Android.

**Ios** : Répertoire contenant l'ensemble des configurations liées à IOS.

**Assets** : Contenant les images et les thèmes de couleurs.

**Components** : Contenant chaque fois les composants visibles dans l'application. Réparti par pages avec un répertoire « common » pour les composants utilisés sur plusieurs pages.

**Config** : Contenant l'environnement de production et développement.

**Constants** : Contenant toutes les constantes que j'utilise lors du projet.

**Context** : Contenant mes requêtes API.

- **Action** : Concernant toutes les actions sur un évènement, utilisateur ou animé.
- **InitialState** : Pour gérer l'état des profils utilisateurs.
- **Reducers** : Concernant toute la partie connexion et inscription.

**Helpers** : Contenant mon instance Axios.

**Navigation** : Contenant tout ce qui concerne la navigation entre les pages.

**Screens** : Contenant les processus/ les méthodes derrière un composant de la page.

**Utils** : Contenant diverses implémentations utiles.

## 2. [Diagramme](#)

User	Event	Anime
id : int name : string firstname : string mail : string totem : string password : string chef : boolean	id : int name : string address : string beginAt : date endAt : date section : array comment : string comment : string participant : array	id : int name : string firstname : string nationalNumber : int address : string mailParent : string numberParent : int section : array state : string

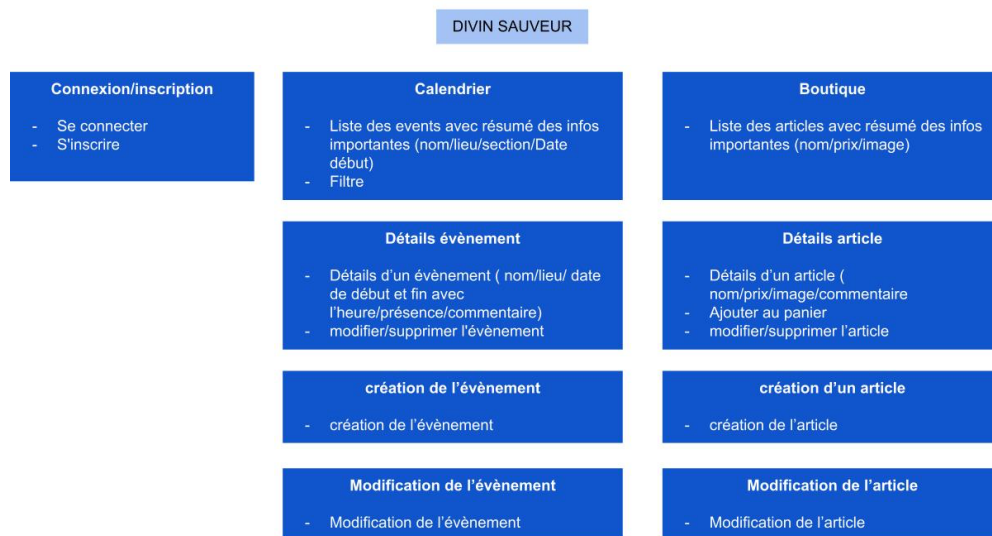
À ce jour, nous comptons 3 différents schémas Mongoose au sein de la base de données.

**User** : Ce modèle servira à la création d'un utilisateur, c'est grâce à ce modèle que nous pourrons gérer les différents droits administrateurs.

**Event** : Modèle qui servira à gérer les évènements sur l'application. Il contient les champs nécessaires afin de déterminer les différents évènements qu'on souhaite afficher.

**Anime** : C'est sur ce modèle que les différents animés seront gérés. Il contiendra tous les champs nécessaires qui permettront à l'administrateur de gérer celui-ci.

### 3. Schéma application



### 4. Connexion/Inscription

#### Front End

Grâce à ces pages, le client pourra s'inscrire et se connecter à l'application. La particularité de l'inscription est que le client ne devra pas introduire de mot de passe pour s'inscrire mais doit en utiliser un pour se connecter. En effet, une fois un client inscrit, il recevra un mot de passe dès que les identifiants seront vérifiés, cela permet de garder l'application en interne au cas où nous décidons de mettre l'application sur les plateformes.

Au niveau de l'inscription, un message d'erreur s'affiche au cas où un champ obligatoire n'est pas rempli. Un utilisateur ne peut pas avoir la même adresse mail qu'un autre utilisateur.

Une fois un utilisateur connecté, il se trouve sur la page d'accueil qui lui permettra de naviguer où il veut sur l'application via la page même ou la sidebar.

#### Back End

Une fois un utilisateur inscrit, il se voit attribuer un Token, qui permettrait à l'administrateur de savoir qui modifie/supprime/ajoute quelque chose dans l'application via le local storage. De plus, il permet une sécurité supplémentaire au cas où quelqu'un de mal intentionné connaissant le chemin d'accès, tenterait de faire une action quelconque via Postman.

Le Token permet de se reconnecter automatiquement à la suite d'un crash de l'application. Lorsqu'un utilisateur tente de s'identifier avec le Token reçu, s'il n'est pas expiré ou invalide, on envoie une réponse « ok » au Back End et l'utilisateur est redirigé directement vers l'écran d'accueil.

#### Design

« Voir figure 1 à 5 »



## 5. Calendrier

### Front End

Sur cette page, le client pourra retrouver la liste de tous les évènements qui se dérouleront sur le planning. En cliquant sur un évènement, le client retrouvera les détails de l'évènement avec une rubrique « présence » où il pourra préciser si l'enfant participe ou non à la réunion, au camp ou aux autres événements inscrits dans le calendrier. Les chefs pourront voir si les enfants participent ou non à la réunion et compléter au cas où l'enfant ou le parent n'a pas l'application, pour que le listing soit correct. Avec ce listing, les chefs pourront garder une trace des présences car c'était obligatoire à cause du COVID-19.

Étant conscient que les mesures prises lors du COVID-19 ont été supprimées, j'ai décidé de garder la liste de présences car, l'unité étant grandissante, cela permet aux chefs de voir si un enfant a été régulier ou non durant l'année.

### Back End

Sur cette page, les événements sont triés par ordre de début. Une fois que l'évènement prend fin, il ne s'affiche plus sur l'application. De plus, lors de la création d'un évènement, l'utilisateur ne peut pas sélectionner une date antérieure à maintenant. De même pour la date de fin, il ne pourra pas choisir une date de fin avant la date de début sélectionnée.

Les utilisateurs pourront également utiliser un filtre qui affichera seulement les événements avec les sections sélectionnées par celui-ci.

L'utilisateur a la possibilité de rajouter/supprimer/modifier un évènement. Au niveau de la modification, la page sera la même que la création d'un évènement mais avec les éléments déjà remplis pour éviter que l'utilisateur réintroduise toutes les données ou refasse d'autres erreurs.

Sur la liste de présences, l'utilisateur pourra rajouter/modifier/supprimer un animé s'il le souhaite.

Lors de la création/modification d'un évènement ou d'un animé, si un champ obligatoire n'a pas été rempli, un message d'erreur s'affichera.

### Design

« Voir figure 6 à 12 »

## 6. Boutique

### Front End

Cette page permet au client de payer/commander diverses choses, comme la cotisation annuelle, ou différents accessoires créés par l'unité.

### Design

« Voir figure 13 à 16 »

## 7. Salon de conversation

Sur cette page, le client pourra voir les messages importants, les infos de dernière minute ou les autres messages envoyés par les chefs.

## **Recommandations**

Après avoir réalisé ce TFE, je peux faire quelques recommandations par rapport à celui-ci.

Tout d'abord, avant de réaliser un projet pour un client, c'est de bien analyser les différentes demandes du client. Ensuite, il faudra choisir les produits qui conviendront le mieux au client.

Pour ce projet en informatique, il est conseillé de suivre la méthode Scrum qui va permettre d'avoir des réunions régulières pour être sûr que le projet correspond bien à ce que le client demande. La communication est le point clé de cette méthode. Le client doit pouvoir être écouté et conseillé vers des propositions qui conviennent à ses demandes.

Découper le projet en plusieurs parties permet de structurer l'implémentation de l'application. On pourra ainsi prévoir le temps qu'il faudra pour se consacrer à ce projet. En effet, lorsqu'un client vous demandera combien de temps cela prendra pour créer un projet de la sorte, vous pourrez répondre plus précisément grâce à cela.

Pour se documenter à propos d'un projet sur React Native, le mieux reste la documentation officielle, car c'est un langage qui change assez régulièrement, vu qu'il est récent. Il est possible aussi de trouver des réponses à des problèmes sur des forums tels que Stack Overflow, mais il faut bien vérifier la version de React Native au préalable.

## **Conclusion**

### **1. Résumé**

En conclusion, ce projet m'a énormément appris sur plusieurs aspects, que ce soit sur le plan technique, relationnel ou organisationnel. Car bien que j'aie été suivi par mon rapporteur, j'ai travaillé en très grande partie en autonomie. Les étapes du projet m'ont permis de prendre conscience à quel point ces étapes sont importantes pour éviter de recommencer plusieurs fois ou même de me perdre dans le projet.

Ce projet m'a permis d'évoluer en programmation. J'ai attrapé aussi beaucoup de reflexes de débogage, à savoir où me renseigner ou autres points des bonnes pratiques de la programmation.

Malgré le fait que mon projet ne soit pas totalement terminé, je suis quand même assez fier du travail fourni. Je compte travailler dessus par la suite, car c'est un projet qui me tient à cœur et qui, je pense, pourrait avoir beaucoup de potentiel à l'avenir dans tout ce qui est information ou facilités dans l'unité.

Pour conclure, le travail m'a appris à gérer un projet pour un client et m'a fait découvrir les différentes étapes pour mener à bien un projet informatique dans un cas concret.

### **2. Problèmes rencontrés**

J'ai rencontré beaucoup d'erreurs durant mon projet. J'ai eu quelques difficultés à implémenter la partie connexion/inscription. De plus, j'avais codé mon application de telle manière que je n'arrivais pas à avancer sur d'autres points, ce qui m'a fait beaucoup stagner sur cette partie.

J'ai mis beaucoup de temps sur la partie navigation, dont les versions changent assez rapidement. Du coup, j'ai eu plein d'erreurs, car la version que j'utilisais n'était pas la bonne.

J'ai perdu pas mal de temps sur la gestion des erreurs de mes formulaires. En effet, le fait d'utiliser plusieurs types de champs comme des input, checkbox, text area, ... m'a pris beaucoup de temps afin de gérer les erreurs correctement.

### 3. Amélioration

Dans un projet informatique, il y a toujours des améliorations possibles. Cependant, malgré la deadline passée, j'aimerais rajouter par la suite toutes sortes d'améliorations qui permettraient un meilleur confort pour l'utilisateur. Voici les améliorations que j'aimerais apporter à la suite de mon TFE.

#### Connexion

Au niveau de la connexion, rajouter un bouton pour voir le mot de passe et mettre un message d'erreur lorsque l'utilisateur se trompe quelque part.

#### Inscription

Au niveau de l'inscription, juste mettre un message d'erreur quand un utilisateur essaye de s'inscrire avec une adresse mail déjà enregistrée.

#### Calendrier

Comme amélioration, j'aimerais rajouter un filtre qui affiche les événements des années précédentes.

Sur la liste de présences, mettre la date de naissance de l'animé avec une rubrique commentaire qui permettra de noter les allergies ou autre dans les détails de l'enfant.

#### Boutique / Conversation

J'aimerais finir toute la partie Back End de la page boutique et implémenter toute la page chat en ligne avant d'installer l'application sur les gsm des staffs dans l'unité.

#### Amélioration générale

En amélioration générale, j'aimerais rajouter plusieurs fonctionnalités telles qu'un listing de camp, qui permettrait de retrouver tous les camps où les sections sont déjà parties avec le lieu, le thème utilisé, les avantages/inconvénients de l'endroit, ... Cette rubrique contiendrait aussi un listing de repas avec les différentes recettes et quantités.

Une page avec tous les documents nécessaires pour un chef ou un parent, comme les autorisations parentales, fiches médicales, ... car trop souvent les parents ou chefs ne donnent pas les bons documents.

Une galerie photos, avec les diaporamas de chaque section de chaque année.

Une page présentation avec un moyen de contact pour les chefs, pour que les parents puissent savoir qui anime leurs enfants et comment ils peuvent les contacter s'ils ont une question.

De plus, j'aimerais créer plusieurs droits administrateurs dans l'application pour pouvoir intégrer les parents/animés sur celle-ci.

#### 4. Prolongement du travail

Comme je l'ai expliqué dans la partie « améliorations », j'aimerais apporter pas mal de choses nouvelles dans l'application. Malgré le fait que mon TFE s'achève cette année, je resterai dans le monde du scoutisme pour pouvoir continuer à améliorer mon application.

## **Code Source**

### **1. Code source : GitHub**

- <https://github.com/AymericPonjee/TFE-10DS>

### **2. Trello**

- <https://trello.com/b/R6Uhyw5B/application-10ieme>

### **3. ClockiFy**

- <https://app.clockify.me/tracker>

## **Bibliographie**

### **1. React Native**

- <https://www.ideomatic.com/dictionnaire-digital/react-native/>
- <https://reactnative.dev/>

### **2. MongoDB**

- <https://datascientest.com/mongodb#:~:text=Apparue%20au%20milieu%20des%20ann%C3%A9es,des%20tableaux%20et%20des%20colonnes.>
- <https://mongoosejs.com/docs/>
- <https://axios-http.com/docs/intro>
- <https://kinsta.com/blog/mongodb-vs-postgresql/>
- <https://www.mongodb.com/compare/mongodb-postgresql>

### **3. Stack MERN**

- <https://www.mongodb.com/mern-stack>

### **4. Scrum**

- <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443834-scrum-guide-de-la-methode-agile-star/>

### **5. Yarn**

- <https://classic.yarnpkg.com/fr/>

### **6. Node.js**

- <https://nodejs.org/fr/about/>

### **7. Express**

- <https://expressjs.com/fr/>

### **8. Postman**

- <https://www.postman.com/>



## Annexes

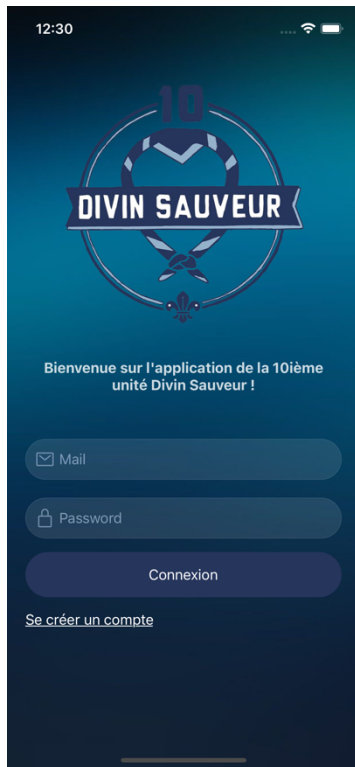


Figure 1: Connexion



Figure 2 : Erreurs inscription

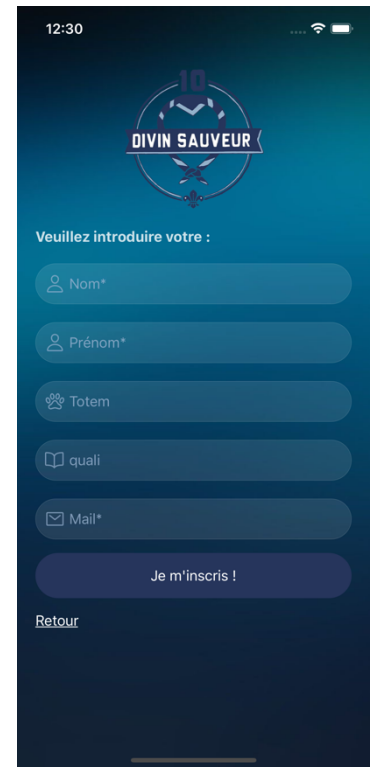


Figure 3 : Inscription

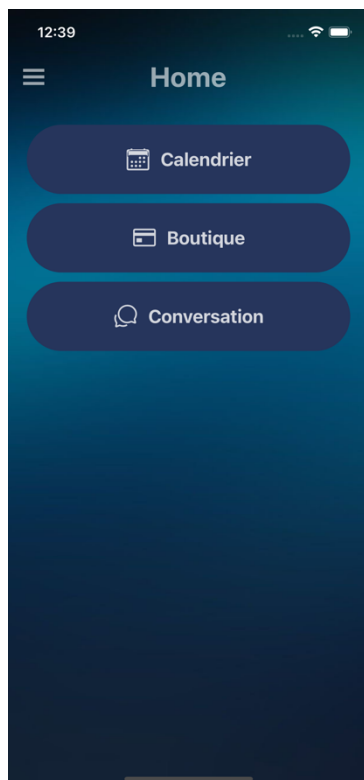


Figure 4 : Home

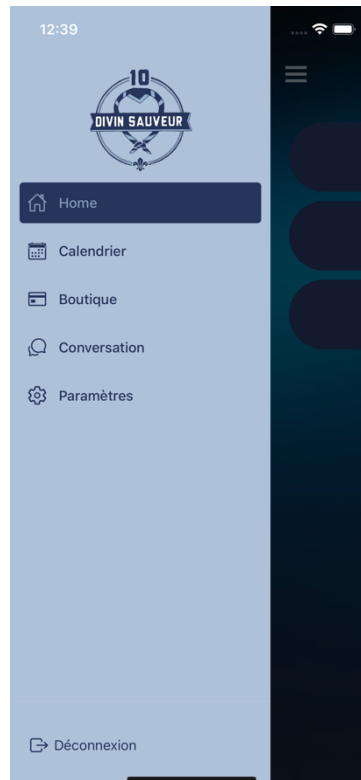


Figure 5 : Sidebar

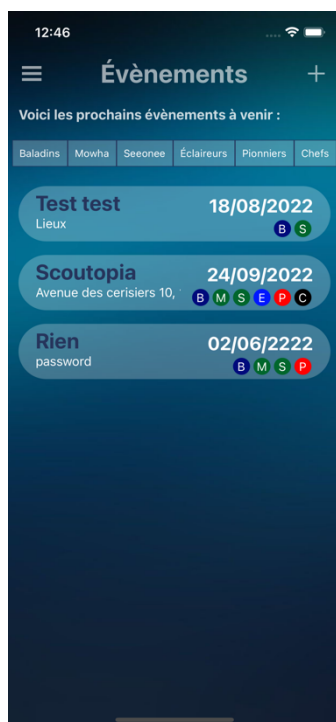


Figure 6 : Calendrier sans filtre

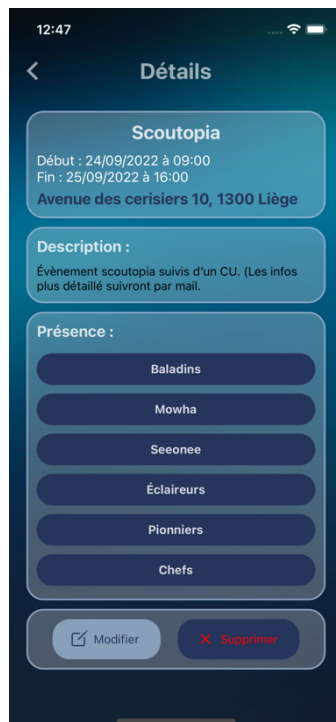


Figure 7 : Détails d'un événement

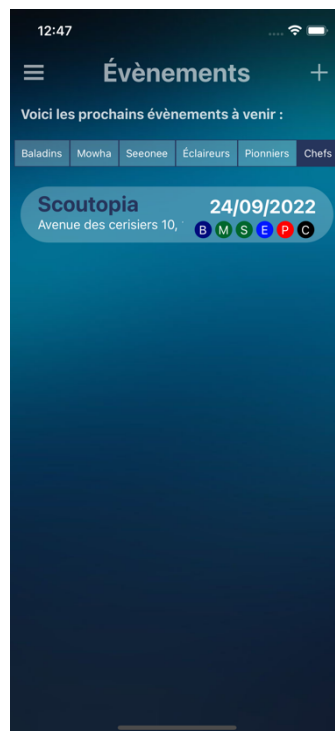


Figure 8 : Calendrier avec filtre

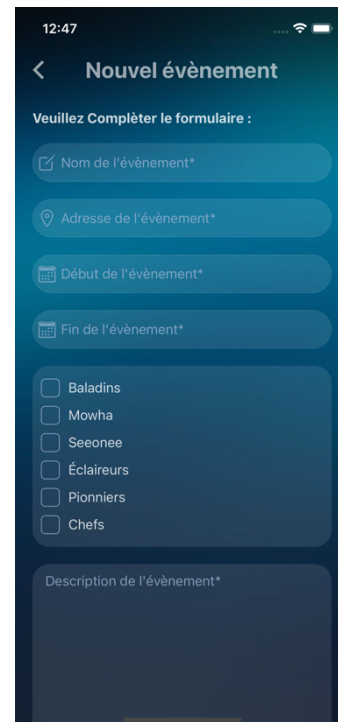


Figure 9 : Ajouter un événement



Figure 10 : Liste de présence

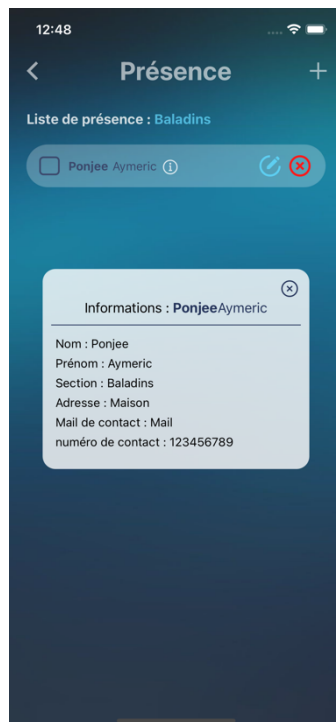


Figure 11: Détails d'un animé

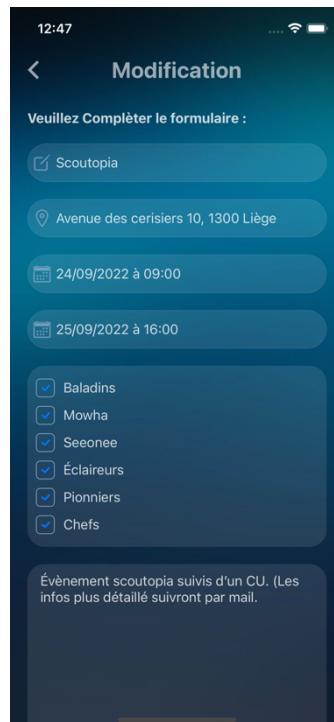


Figure 12 : Modification d'un événement

1:01

< **Nouvel article**

**Veillez Compléter le formulaire :**

Nom de l'article\*  
veillez entrer un nom pour l'objet

Prix de l'article\*  
veillez entrer un prix pour l'objet

Description de l'article\*  
veillez entrer une description pour l'objet

Image de l'article\*  
veillez upload une image pour l'objet

Ajouter l'objet à la boutique

Figure 13 : Erreur nouvel article

1:01

< **Nouvel article**

**Veillez Compléter le formulaire :**

Nom de l'article\*

Prix de l'article\*

Description de l'article\*

Image de l'article\*

Ajouter l'objet à la boutique

Figure 14 : Nouvel article

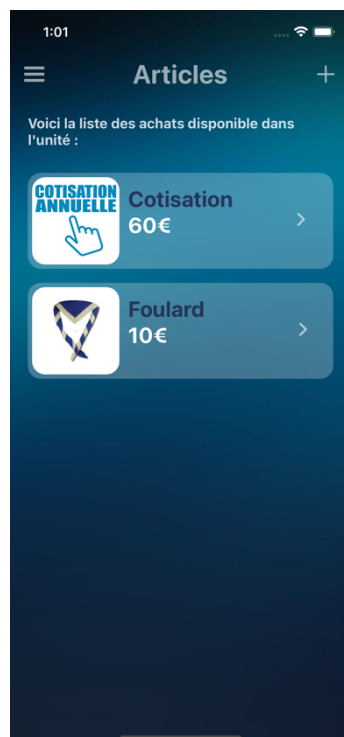


Figure 15 : Boutique



Figure 16 : Détails article