

KHELILI Hédi
BTS SIO 1^{er} année

Stage de Développeur mobile pour La Montagne



Table des matières

Présentation.....	3
Ressources informatiques utilisées.....	3
Création de la Page Menus.....	4
Création de la Page Typographies.....	6
Création de la Page Couleur.....	7
Création de la page boutons.....	9
Création de la Page Spacing.....	9
Création de la Page Changement de mode d'apparence.....	9
Conclusion.....	10

Présentation.

Le groupe Centre France est un acteur majeur de la presse en région depuis la création du journal La Montagne, en 1919. Il puise sa force dans un maillage fin du territoire, à travers une présence éditoriale sur 15 départements et 4 régions.

Avec l'avènement de l'ère numérique, Centre France a évolué pour répondre aux nouvelles exigences du paysage médiatique. Aujourd'hui, le groupe dispose d'une présence numérique substantielle, avec des versions en ligne de ses journaux et d'autres plateformes numérique dédiées à la vidéo, à l'audio et à la diffusion en direct.

Pour le premier stage j'ai intégré l'équipe des produits digitaux en charge du tournant numérique qu'opère le groupe. Cette équipe est responsable des divers supports digitaux du groupe. Ces supports représentent 915 000 visites quotidiennes sur l'ensemble des sites et applications et 2.3 millions de pages vues chaque jours.

Mon rôle consistait en la création d'une application mobile de test à destination de l'équipe UX dans le but d'avoir un aperçu concret de toutes les modifications esthétiques sur les applications mobiles du groupe. Cette application a été conçue via le framework React Native.

Mes tâches sont découpées en différents sprints correspondant chacun à une partie de l'application à concevoir en respectant un cahier des charges déterminé.

Ressources informatiques utilisées.

React Native

React Native fait référence à un cadre basé sur JavaScript utilisé pour créer des applications mobiles natives sur Android et la plate-forme iOS.

Il est basé sur React, la bibliothèque JavaScript de Facebook, pour créer des interfaces utilisateur pour les plateformes mobiles.

Un gros avantage et la raison pour laquelle La Montagne l'utilise est qu'un seul et même code écrit sous React Native sera à la fois utilisé pour Android et iOS. Les applications React Native sont créées en combinant JavaScript avec un balisage de type XML, c'est-à-dire JSX. La passerelle React Native est chargée d'effectuer le rendu natif des API en Java pour Android et Swift pour iOS.

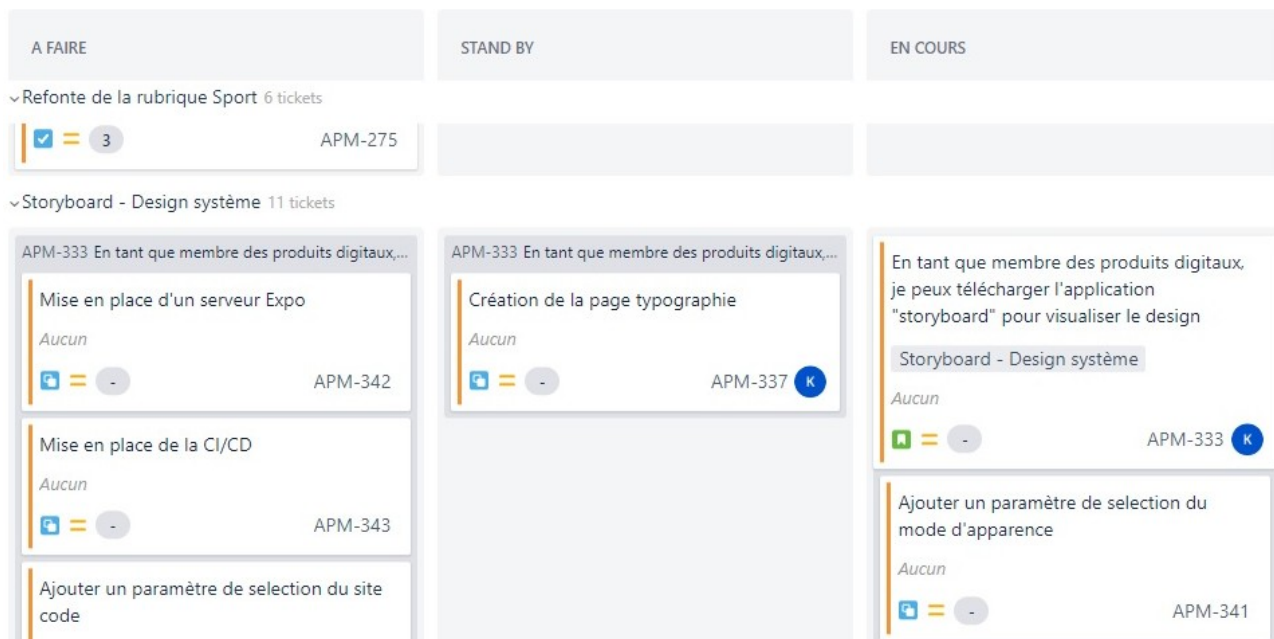
JIRA

Jira Software fournit des tableaux Scrum et Kanban prêts à l'emploi. Les tableaux constituent des centres de gestion des tâches, dans lesquels les tâches sont mappées à des workflows.

personnalisables. Ils assurent la transparence du travail d'équipe et offrent une visibilité sur l'état d'avancement de chaque tâche. Exemple d'une capture d'écran fait pendant mon stage :

ANDROID STUDIO

Android Studio est un environnement de développement pour développer des applications mobiles



Android. Je l'ai surtout utilisé pour son émulateur qui ma permit de lancer mon application.

GitLab

GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.

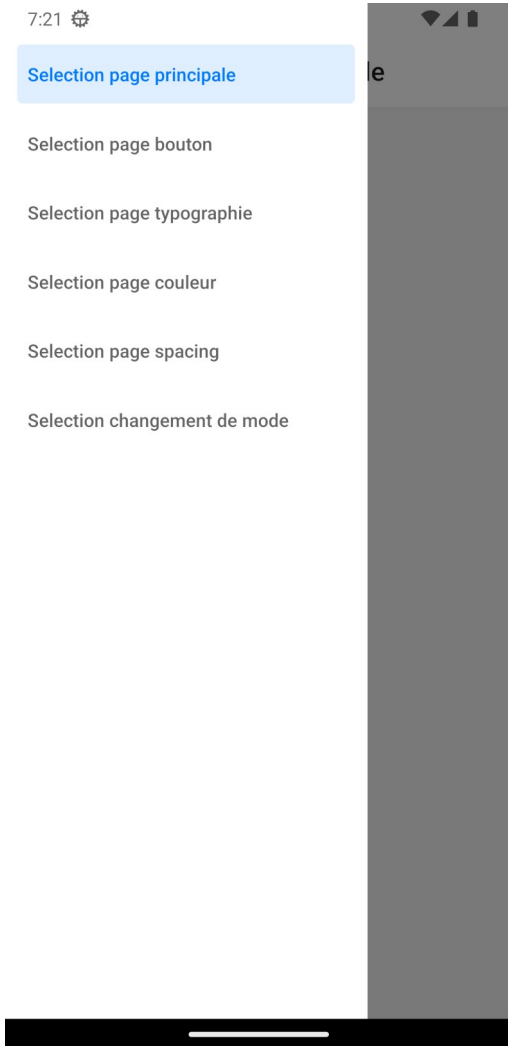
Création de la Page Menus

Dans un premier temps j'ai du préparer l'ordinateur qui m'a été confié en installant le nécessaire pour coder en React Native. La plateforme Node.js qui est un prérequis, l'éditeur de code studio code...

Une fois le pc près je me suis familiarisé avec le typeScript qui est le langage utilisé par React Native et j'ai suivi les tutoriels disponibles sur le site de React Native <https://reactnative.dev/>

Une fois près je me suis attaqué à mon premier ticket, la création du menu.

J'ai fait un menu dit Drawer Navigator, c'est un menu qui ouvre une fenêtre latérale lors d'un « clique » sur le haut a gauche de l'écran et qui propose les divers pages. je trouve les informations sur son fonctionnement <https://reactnavigation.org/docs/drawer-navigator/>
Chaque onglets du menu correspond à une page qui sera une tâche qui m'est attribué.



```
E: > notes > Rapport de stage > étapes > 01menu et typo fini > extrait de code > TS Drawer.tsx > ...
1  |/*
2  | Permet d'avoir un menu latéral qui redirige sur différentes pages.
3  | */
4  import * as React from 'react';
5  import {NavigationContainer} from '@react-navigation/native';
6  import {createDrawerNavigator} from '@react-navigation/drawer';
7
8
9  import PagePrincipale from './PagePrincipale';
10 import PageBouton from './PageBouton';
11 import PageTypographie from './PageTypographie';
12 import PageCouleurs from './PageCouleurs';
13 import PageSpacing from './Spacing';
14 import PageChangeMode from './ChangeMode';
15
16
17 const Drawer = createDrawerNavigator();
18
19 function MyDrawer() {
20   return (
21     <Drawer.Navigator>
22       <Drawer.Screen name="Selection page principale" component={PagePrincipale} />
23       <Drawer.Screen name="Selection page bouton" component={PageBouton} />
24       <Drawer.Screen name="Selection page typographie" component={PageTypographie} />
25       <Drawer.Screen name="Selection page couleur" component={PageCouleurs} />
26       <Drawer.Screen name="Selection page spacing" component={PageSpacing} />
27       <Drawer.Screen name="Selection changement de mode" component={PageChangeMode} />
28     </Drawer.Navigator>
29   );
30 }
31
32 const Menu = () => {
33   return (
34     <NavigationContainer>
35       <MyDrawer />
36     </NavigationContainer>
37   );
38 };
39 export default Menu;
```

Durant les différentes phases de ma programmation, c'est sur GitLab que mes codes étaient corrigés par les membres de l'équipe.

Overview 18 Commits 3 Pipelines 0 Changes 18 18 unresolved threads

packages/app-ui/example/src/App.tsx

Adrien Lefaire @adlefau1 started a thread on an old version of the diff 19 hours ago
Last updated by Hedi KHELILI 17 hours ago

packages/app-ui/example/src/App.tsx

1 + /**

Adrien Lefaire @adlefau1 · 19 hours ago Maintainer ✓ 😊 ⋮

Le bloc commentaire ne correspond pas au fichier. Généralement on ne l'utilise pas (à Centre France)

Hedi KHELILI changed this line in version 2 of the diff 17 hours ago · [Compare changes](#)

Reply...

Resolve thread

Adrien Lefaire @adlefau1 started a thread on an old version of the diff 19 hours ago
Last updated by Hedi KHELILI 17 hours ago

packages/app-ui/example/src/Drawer.tsx 0 → 100644

```

1 + import 'react-native-gesture-handler';
2 + import * as React from 'react';
3 + import {NavigationContainer} from '@react-navigation/native';
4 + import {createDrawerNavigator} from '@react-navigation/drawer';
5 +
6 +
7 + import PagePrincipale from './PagePrincipale';
8 + import PageBouton from './PageBouton';
9 + import PageTypographie from './PageTypographie';
10 + import PageCouleurs from './PageCouleurs';
11 + import Spacing from './Spacing';

```

Adrien Lefaire @adlefau1 · 19 hours ago Maintainer ✓ 😊 ⋮

Garde la même type de nommage pour toutes tes pages. Du coup, PageSpacing Et pour ChangeMode, je pense que tu voulais faire une page de paramètre ? Du coup, PageParamètre. Comme ça on pourra rajouter d'autre paramètre plus tard si on a besoin

Hedi KHELILI changed this line in version 2 of the diff 17 hours ago · [Compare changes](#)

Création de la Page Typographies

Pour réaliser cette page, on me donne un exemple de page de typographie que je dois copier.

Typographies - Doc - Storybook X

development.gitlab-pages.centrefrance.fr/4/storybook/doc/path=/docs/typographies-docs#typographies

Non sécurisé

Dev en cours de synchronisation

Typographies

- Docs
 - liste de typographies article
 - liste de typographies cards
- ATOMS
 - Icons
 - Logos
 - Badge
 - Checkbox
 - Combobox
 - Event
 - Feedback
 - FlagTagLogo
 - Input
 - Link
 - NavigationItem
 - Picticon
 - PremiumIcon
 - Radio
 - Select
 - Separator
 - Skeleton
 - StickyHeader
 - Switch
 - Tag
 - Textarea
- API
 - ScrollSnapper
- EXTERNALS
 - FacebookEmbed
 - InstagramEmbed
 - TwitterEmbed
- MOLECULES
 - Button
 - Citation
 - ConsentWrapper
 - Embed

red theme

Typographies

Articles

H1 - Titre de niveau 1

H2 - Titre de niveau 2

H3 - TITRE DE NIVEAU 3

H4 - Titre de niveau 4

H5 - Titre de niveau 5

H6 - TITRE DE NIVEAU 6

Paragraphe de niveau 1 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt morbi pulvinar leo, ipsum, in elit, massa tristique consectetur.

Paragraphe de niveau 1 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt morbi pulvinar leo, ipsum, in elit, massa tristique consectetur.

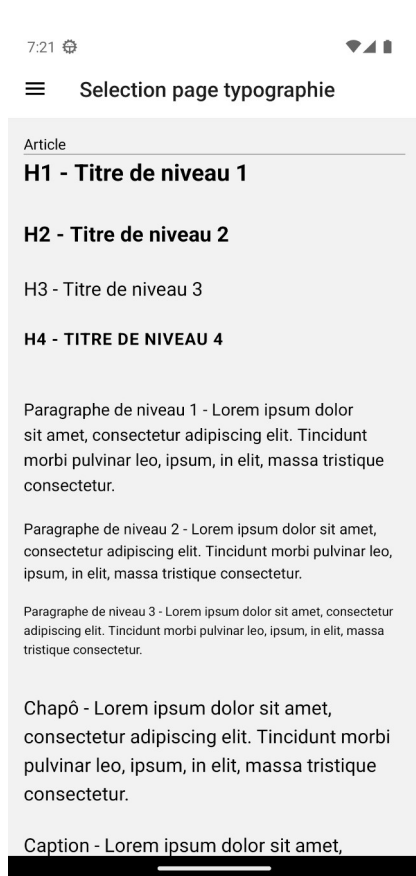
Paragraphe de niveau 2 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt morbi pulvinar leo, ipsum, in elit, massa tristique consectetur. Enim nibh hac sit malesuada sed. Feugiat justo pharetra natoque eleifend.

Paragraphe de niveau 2 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt morbi pulvinar leo, ipsum, in elit, massa tristique consectetur. Enim nibh hac sit malesuada sed.

Paragraphe de niveau 3 - Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt morbi pulvinar leo, ipsum, in elit, massa tristique consectetur. Enim nibh hac sit malesuada sed.

Acroche

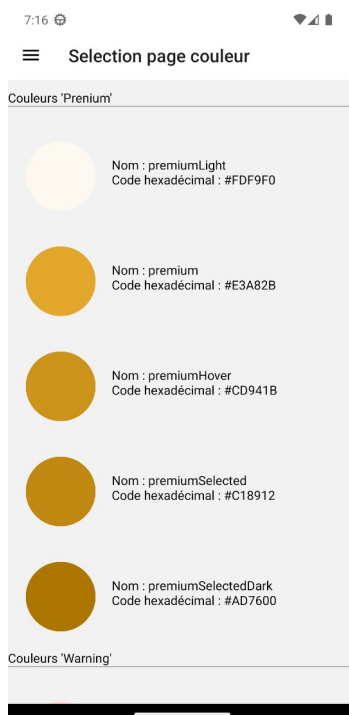
Le but est de refaire la même page mais avec les typographies spécifiques du journal.
Comme cette page est au final assez grande, j'ai du ajouter un scrolle pour naviguer.



```
TS PageTypographie.tsx 4 X
E: > notes > Rapport de stage > étapes > 01menu et typo fini > extrait de code > TS PageTypographie.tsx > default
1  /*
2  La page typographie qui permet de voir les différentes typo et couleurs.
3  */
4  import * as React from 'react';
5  import {View, Text} from 'react-native';
6  import {useTheme} from '@cf-mobile/app-ui';
7  import {ScrollView} from 'react-native-gesture-handler';
8
9  function PageTypo() {
10   const theme = useTheme();
11
12   return (
13     <ScrollView>
14       <View style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>
15         <Text style={theme.fonts.headline1}>H1 - Titre de niveau 1</Text>
16         <Text style={theme.fonts.headline2}>H2 - Titre de niveau 2</Text>
17         <Text style={theme.fonts.headline3}>H3 - Titre de niveau 3</Text>
18         <Text style={theme.fonts.headline4}>H4 - Titre de niveau 4 {"\n"}</Text>
19         <Text style={theme.fonts.chapo}>Chapô - Lorem ipsum dolor sit amet, conseq
20         <Text style={theme.fonts.body1}>Paragraphe de niveau 1 - Lorem ipsum dolor
21         <Text style={theme.fonts.body2}>Paragraphe de niveau 2 - Lorem ipsum dolor
22         <Text style={theme.fonts.body3}>Paragraphe de niveau 3 - Lorem ipsum dolor
23         <Text style={theme.fonts.quote}>quote - Lorem ipsum dolor sit amet, conseq
24         <Text style={theme.fonts.hook}>Acroche</Text>
25         <Text style={theme.fonts.date}>Date</Text>
26         <Text style={theme.fonts.caption}>Caption - Lorem ipsum dolor sit amet, conseq
27         <Text style={theme.fonts.signature}>Signature - Lorem ipsum dolor sit amet, c
28         <Text style={theme.fonts.link}>Liens article - Un vol Clermont-Paris contrain
29       </View>
30     </ScrollView>
31   );
32 }
33
34 export default PageTypo;
35
```

Création de la Page Couleur.

Les consignes sont relativement libre pour la conception de cette page. On me donne toutes les couleurs utilisées par le journal et je dois faire une page qui les composent avec leur noms. Donc après de nombreux essais, le plus simple et ludique me semble de faire des ronds de couleurs et d'écrire les informations les concernant sur le côté.



```

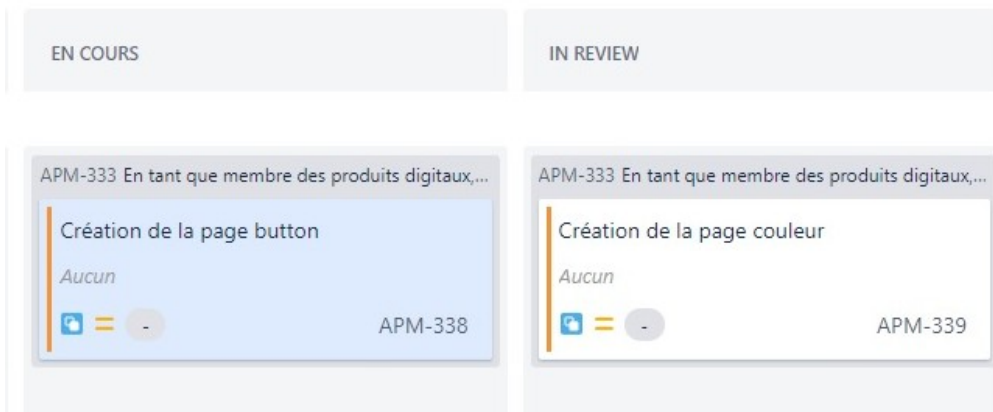
let rayon = 0.2;

function PageCouleur() {
  return (
    <ScrollView>
    <View style={{flex: 1}}>
      {
        Object.entries(AppColors).map((color, index) => {
          return (
            <View key={index} style={{flexDirection: 'row', padding: 20}}>
              <View style={{
                [
                  styles.cercle,
                  {backgroundColor: color[1]}
                ]
              }}>
            <Text>` \n Nom: ${color[0]} \n Code hexadécimal: ${color[1]} `</Text>
          </View>
        )
      }
    </View>
  </ScrollView>
);
}

const styles = StyleSheet.create({
  cercle: {
    borderRadius: Math.round(Dimensions.get('window').width + Dimensions.get('window').height)/2,
    width: Dimensions.get('window').width * rayon,
    height: Dimensions.get('window').width * rayon,
    marginRight: 20
  }
});

```

Une fois cela fait, je passe ce code en review pour être contrôlé et je commence un autre ticket, ma page de création des boutons.



Produits digitaux > app-mobile > Merge requests > !365

Affiche les couleurs avec leur noms et code hexa

Open Hedi KHELILI requested to merge [feat/APM-339](#) into [3.4.3-preview.1](#) just now

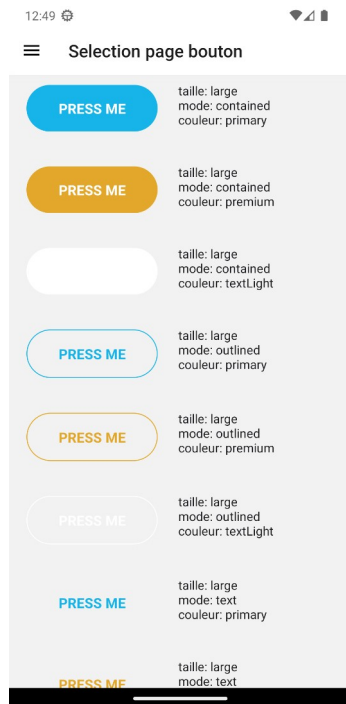
Overview **0** Commits **1** Pipelines **0** Changes **1**

Related to [APM-339](#)



Création de la page boutons

Ici il m'est demandé de faire une page qui contient les boutons utilisés par l'application mobile. Je dois reconnaître que cela à était assez fastidieux car un bouton contient de nombreux paramètres en fonction de l'effet voulu. Après avoir compris le fonctionnement d'un bouton, je les codes.



```
function PageBouton() {
  return (
    <ScrollView>
      <View style={{flex: 1, }}>
        <Text style={{textAlign:'center', padding:10}}>Boutons génériques</Text>
        {
          Object.entries(AppButton.Size).map((size ) => {
            return (
              Object.entries(AppButton.Mode).map((mode) => {
                return (
                  Object.entries(AppButton.Color).map((color,index1) => {
                    return (
                      <View key={index1} style={{flexDirection:'row',}}>
                        <View style={{padding:20}}>
                          <AppButton style={{width:150}} onPress={()=>{}}
                            size={size[1]}
                            mode={mode[1]}
                            color={color[1]}
                          >press me</AppButton>
                        </View>
                        <Text>{'\n taille: ${size[1]}\n mode: ${mode[1]}\n couleur: ${color[1]} `'}</Text>
                      </View>
                    );
                  });
                });
              });
            });
          });
        }
      </View>
    </ScrollView>
  );
}
```

Création de la Page Spacing

Cela peut sembler anodin mais rien n'est laissé au hasard dans un journal en ce qui concerne le format des pages. J'ai donc faire une page qui contient toutes les marges de l'application.

```
Object.entries(AppSpacing).map((space , index) => {
  return (
    <View key={index}>
      <Text style={{textDecorationLine:'underline', margin:10}}>Avec un espace: {space[0]}</Text>
      <Text style={{marginLeft:space[1]}}>Lorem ipsum dolor sit amet consectetur adipisicing elit. Explic
    </View>
  );
});
```

Création de la Page Changement de mode d'apparence

Il m'est demandé de faire une page qui contient un bouton qui permet de changer de mode dark mode /light mode et cela indépendamment du réglage du téléphone.

Cette page a été très compliqué à réalisé car il à fallut comprendre comment fonctionne les modes apparence, comment les appelé etc..

```
//comme ApplicationProvider, accède à applicationState.setThemeMode(ThemeMode.dark)
<View style={styles.container}>
  <Text>La page qui change de mode{"\n"}</Text>
  <AppButton color={AppButton.Color.Primary} onPress={()=>{applicationState.setThemeMode(ThemeMode.dark)}} >button dark MODE </AppButton><Text></Text>
  <AppButton color={AppButton.Color.Primary} onPress={()=>{applicationState.setThemeMode(ThemeMode.light)}}>button light MODE</AppButton><Text></Text>
  <AppButton color={AppButton.Color.Primary} onPress={()=>{applicationState.setThemeMode(ThemeMode.system)}}>button auto MODE</AppButton><Text></Text>
  <AppIconButton onPress={()=>{}}
    icon = {({color}) => <Entypo name="bug" size={40} color={color} />}
    disabled={false}
  ></AppIconButton><Text></Text>

  <Text>Thème {essai}</Text>
  <Switch
    onValueChange={(value) => {
      toggleSwitch();
      applicationState.setSiteCode(value ? "EV": "MT")
      value ? setEssai("l'EVEIL"): setEssai("La MONTAGNE")
    }}
    value={isEnabled}
  />
</View>
```

Conclusion

Comprendre comment fonctionne la partie développement logiciel d'une entreprise en prenant en compte les différents acteurs impliqués dans différents domaines (back office, front office, gestion de la publicité, expérience utilisateur...) a été extrêmement enrichissant d'un point de vue personnel et technique.

Apprendre sur le tas React Native et le TypeScript ne fut pas chose aisée. Bien que je n'ai pas réalisé l'ensemble des tâches attendu, je pense avoir laissé une bonne image de mon école et moi même.

Et pour finir je remercie l'équipe des produits digitaux d'avoir pris de leur temps pour m'accueillir dans leur entreprise.