

Présentation du projet de fin de formation pour l'obtention
du certificat de qualification

Intitulé

Développement d'une application mobile « FIXUP »

Spécialité : Développement WEB & Mobile

Réalisé par : Mohammed Iliass AFFANI

Encadré par : Abdellatif SOKLABI



Date : 21 Juin 2022

Dédicace

Je dédie ce modeste travail :

A mes chers parents qui ont tant donné.

Pour leur immense soutien, leur grand amour, leurs sacrifices et leurs prières.

A mes chers frères et sœurs.

Vous aviez toujours cru en moi, et c'est dans votre présence que j'ai puisé la volonté de continuer.

A toute ma famille.

A tous mes chers amis.

A tous ceux qui m'ont aidé de près ou de loin.

Remerciement

Je tiens tout d'abord à remercier Mon encadrant et enseignant Abdellatif SOKLABI pour son soutien qui m'a porté durant cette période de formation ainsi que toute personne j'ai eu l'opportunité de rencontrer à SOLICODE TANGER et qui nous ont accueilli avec beaucoup d'amabilité, chacun a su rendre notre formation plus agréable et instructif.

Sommaire

Dédicace	2
Remerciement	2
I. Introduction.....	4
II. Problématique.....	4
III. Gestion de projet	5
1. Diagramme GANTT	5
2. DESIGN THINKING.....	5
IV. Réalisation	7
1. Analyse fonctionnelle :	7
2. Outils et technologies utilisées.....	7
3. L'interface graphique.....	8
Conclusion générale	10
Annexes	11

Tables des figures

Figure 1: Diagramme GANTT du projet	5
Figure 2 : Les étapes de la méthode DESIGN THINKING	5
Figure 3 : Exemple d'une Carte d'empathie	6
Figure 4 : Diagramme des cas d'utilisation	7
Figure 5 : Diagramme de classe de l'application FIXUP	7
Figure 6 : Page d'accueil	8
Figure 7 : schéma descriptif pour créer un compte sur FIXUP.....	9
Figure 8 : Envoi d'une demande d'intervention.....	9

I. Introduction

Le **CRM**, Customer Relationship Management en Anglais, ou **Gestion de la Relation Client (GRC)** en français, est un concept de La gestion de la relation client, ou gestion des relations avec la clientèle est l'ensemble des outils et techniques destinés à tenir compte des souhaits et des attentes des clients et des prospects, afin de les satisfaire et de les fidéliser en leur offrant ou proposant des services.

Notre projet s'articule autour de cette gestion de relation client qui se traduit automatiquement par la gestion des services vis-à-vis le client.

Le concept de notre projet repose sur l'idée de gagner le plus de temps possible pour faciliter la vie quotidienne des gens à l'aide d'une application facile à utiliser pour donner une expérience interface-client unique.

II. Problématique

De nos jours et avec les occupations des gens, il est devenu difficile de trouver le temps pour chercher un technicien, l'appeler et voir selon sa disponibilité est ce qu'il peut assurer une intervention de maintenance à domicile ou non, surtout si le client est en situation difficile et la panne est très et elle exige une intervention instantanée.



L'Offre : Notre service offre une prestation à domicile permettant le client de demander en ligne une intervention de maintenance selon son besoin. Cette prestation est assurée par des techniciens et des spécialistes pouvant résoudre à tout moment les pannes qui peuvent nuire au quotidien des gens.

L'application **FIXUP** est conçue pour répondre à ce besoin permettant l'utilisateur à demander de multiples types d'intervention : Electricité, menuiserie, climatisation, plomberie etc, en ligne et en un seul clic.

La valeur ajoutée : L'application permet à l'utilisateur d'envoyer une description bien détailler avec une photo. A ce moment-là l'intervenant apportera avec lui tout le matériel nécessaire pour résoudre le problème rapidement sans avoir besoin de se déplacer une deuxième fois.

III. Gestion de projet

1. Diagramme GANTT

Pour mieux visualiser les diverses phases de ce projet, un diagramme GANTT a été réalisé qui représente graphiquement la planification des tâches au cours du temps imparti, comme le montre la figure ci-après :

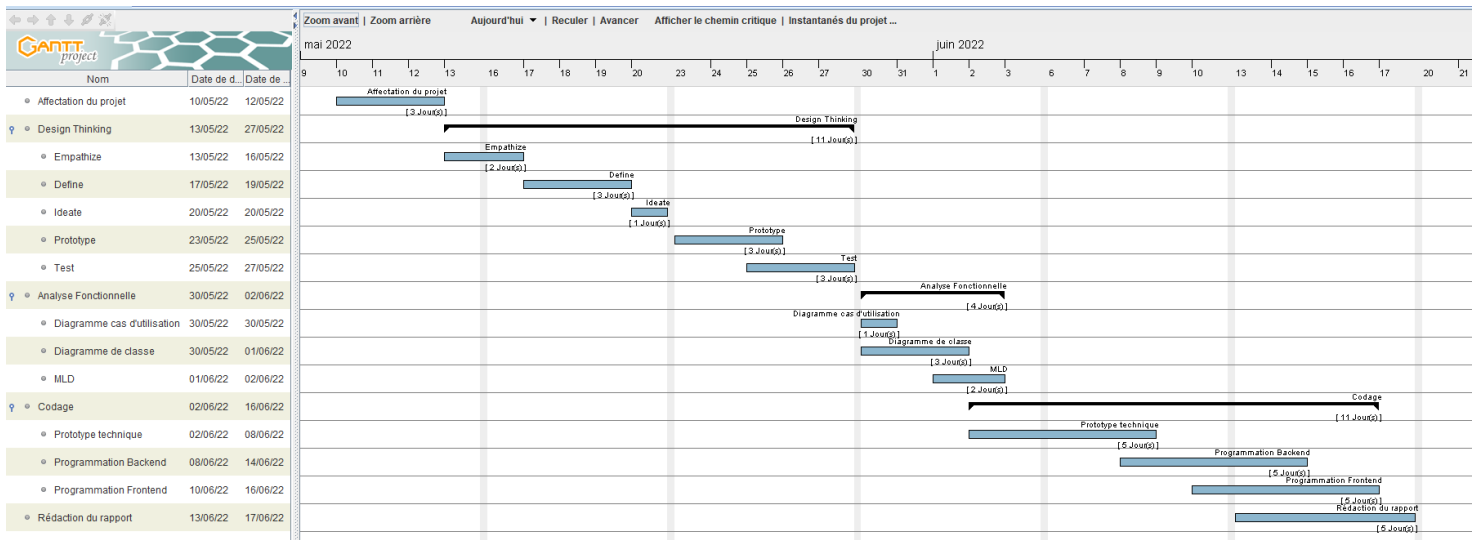


Figure 1: Diagramme GANTT du projet

2. DESIGN THINKING

Méthode du DESIGN THINKING : est une approche de l'innovation et une méthode ou un process de conception globale, centré sur l'utilisateur en vue de réaliser des services ou produits innovants.

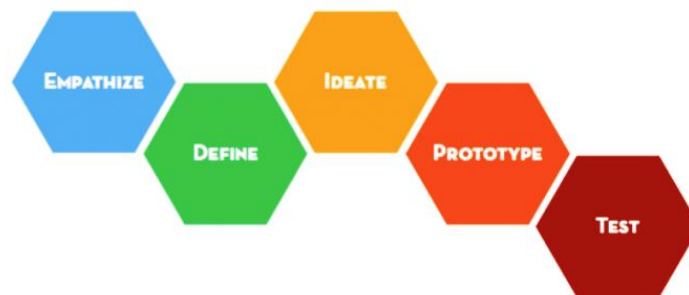


Figure 2 : Les étapes de la méthode DESIGN THINKING

1) Empathie

La première étape pour but de se mettre à la place du consommateur et d'essayer de comprendre ses besoins. Il faut donc les rencontrer, les interroger pour connaître leur mode

de vie, leurs pensées, leurs sentiments et leur avis en utilisant une carte d'empathie. L'objectif est de se mettre à leur place pour comprendre ce dont ils ont besoin (voir Figure 3).

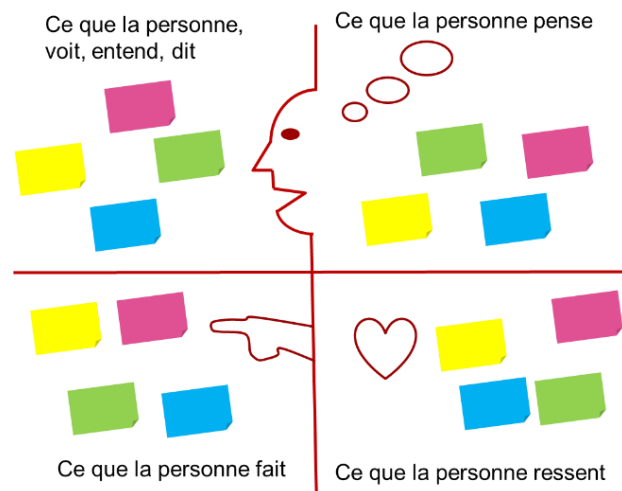


Figure 3 : Exemple d'une Carte d'empathie

2) Définir

Il s'agit de **comprendre la problématique** sur laquelle on travaille. C'est le point de départ. Il est essentiel car on sait bien qu'un problème mal posé conduit rarement à une solution satisfaisante. Pour bien comprendre le besoin d'un projet, il faut mettre le doigt sur le cœur de la problématique à résoudre.

3) Idéation

C'est l'étape la plus importante du process qui consiste à générer des idées (brainstorming) et de les tester auprès de l'utilisateur final en guise d'optimisation du produit final (l'apprentissage par l'erreur).

4) Prototype

L'objectif de cette quatrième phase est de croiser les idées, de les affiner, de commencer à élaborer les premiers brouillons du projet. Durant cette étape, vous pouvez, ou non, envoyer vos premiers schémas aux clients et leur demander leur avis.

5) Test

La dernière phase a pour objectif de tester le projet, d'affiner les solutions proposées grâce au retour des utilisateurs. Le test est le meilleur moyen de savoir si son projet tient la route, si ses idées sont réalisables. Le feedback des utilisateurs vous permettra d'avoir une nouvelle vision de votre projet et de réagir en conséquence. Le but de cette étape est simple : tester vos prototypes, écouter les retours des utilisateurs, modifier, redéfinir et revoir les priorités.

IV. Réalisation

1. Analyse fonctionnelle :

- Diagramme des cas d'utilisation :

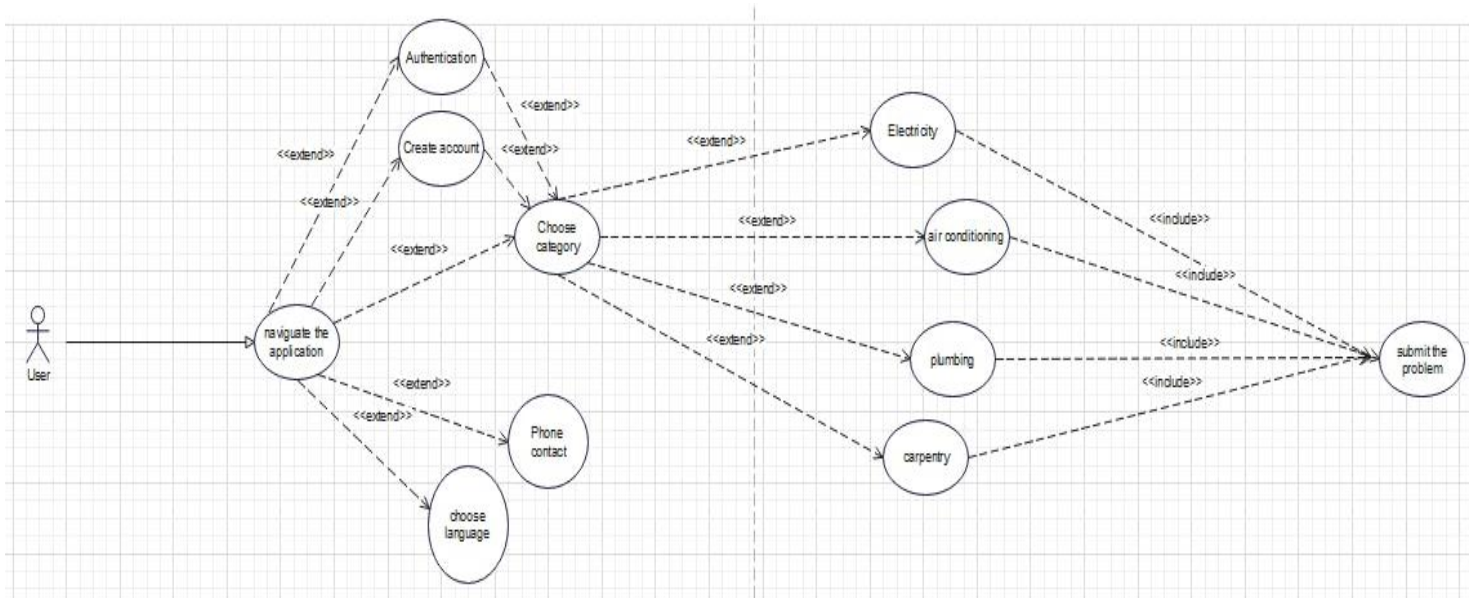


Figure 4 : Diagramme de classe de l'application FIXUP

- Diagramme de classe :

L'application repose sur deux entités comme suit :

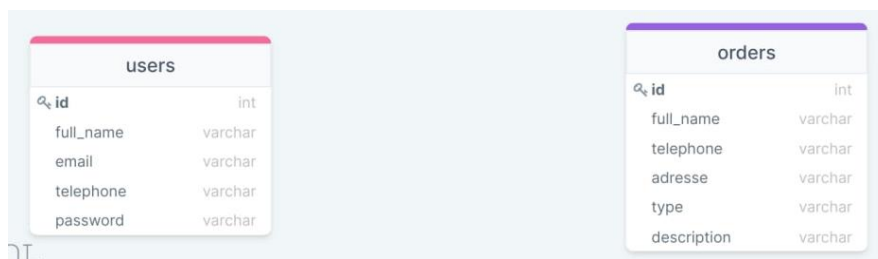


Figure 5 : Diagramme des cas d'utilisation

2. Outils et technologies utilisées



Visual Studio Code

: est un éditeur de code extensible développé par Microsoft pour Windows, Linux et mac OS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.



git : est logiciel VCS (Version Control System) local qui permet aux professionnels de la data et aux développeurs de sauvegarder l'historique de modifications de leurs projets. C'est un logiciel de gestion de version pour gérer les sources de son noyau open source. Il s'agit ainsi d'un logiciel de versioning qui permet de conserver un historique des modifications.



React Native : est un framework Javascript d'applications mobiles open source, utilisé pour développer des applications pour Android, iOS et UWP.



Firebase : est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.



Adobe
XD

: est un outil de design et prototypage d'interface et de conception de sites web ou d'applications mobile.

3. L'interface graphique

Lors de l'ouverture de l'application, l'utilisateur entre dans une page d'accueil qui se présente comme suit :



Figure 6 : Page d'accueil

Dans un premier temps, l'utilisateur peut directement appeler le service maintenance à l'aide du **bouton d'appel en bas**.

S'il voudrait envoyer une demande en ligne par message écrit, il doit créer un compte (voir Figure 7) et puis remplir une page, selon la nature de la panne (Voir figure 8), avec toutes les informations demandées et une photo du problème si besoin.



Figure 7 : schéma descriptif pour créer un compte sur FIXUP

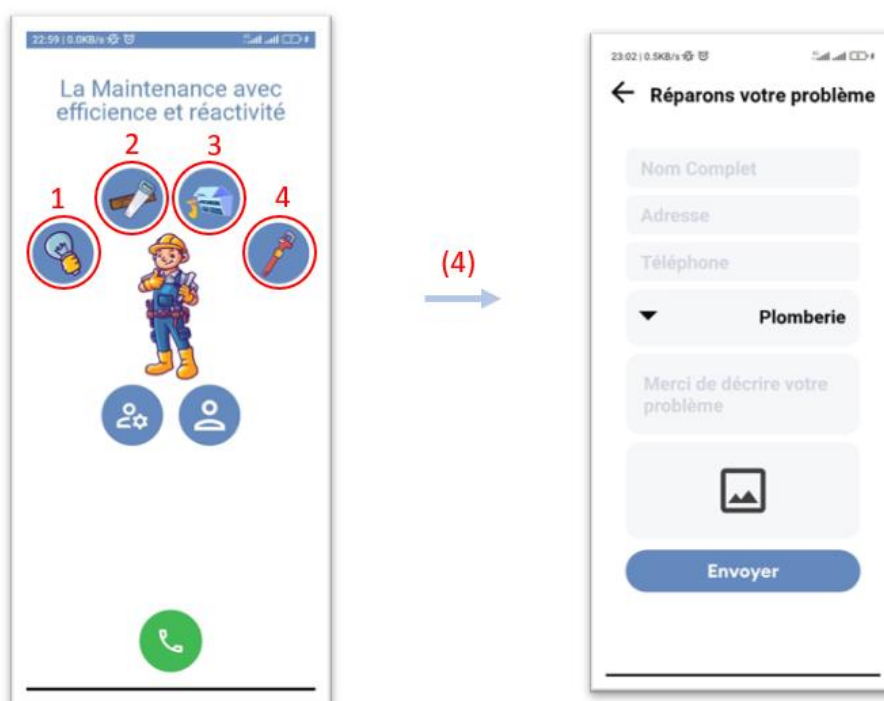


Figure 8 : Envoi d'une demande d'intervention

Conclusion générale

En guise de conclusion, la recherche d'un technicien compétent est devenue de plus en plus complexe, d'où le gaspillage du temps lors de la recherche et lors des interventions de maintenance chez le client.

La réalisation de l'application FIXUP facilitera la tâche non seulement pour le client mais aussi pour l'intervenant en lui permettant de connaître au préalable le matériel nécessaire et réduire le temps de résolution du problème.

Grâce à cette expérience, j'ai pris conscience de ma capacité de gestion de temps et de projet, j'ai développé durant cette formation des compétences techniques en développement web et mobile tout en bénéficiant du retour d'expérience de plusieurs personnes expérimentées lors des événements et les workshops qu'on a pu organisés.

Annexes

Extraits du code

```

EXPL... LoginScreen.js
src > screens > LoginScreen.js
1 import {
2   StyleSheet,
3   Text,
4   TouchableWithoutFeedback,
5   View,
6   Keyboard,
7 } from 'react-native';
8 import React, {useState} from 'react';
9 import Header from '../components/Header';
10 import BtnTwo from '../components/BtnTwo';
11 import {Link} from '@react-navigation/native';
12 import {Font} from '../constants';
13 import Traduction from '../translate';
14 import {useSelector, useDispatch} from 'react-redux';
15 import {Formik} from 'formik';
16 import FormInput from '../components/FormInput';
17 import * as Yup from 'yup';
18 import auth from '@react-native-firebase/auth';
19 import GeneralStorage from '../Store/Storage/GeneralStorage';
20 import GeneralAction from '../Store/Actions/GeneralAction';
21
22 const LoginScreen = ({navigation}) => {
23   const [lang] = useSelector(state => state.Language);
24
25   const validation = Yup.object({
26     email: Yup.string().required('Required').email('Email Format is Incorrect'),
27     password: Yup.string()
28       .required('Required')
29       .min(6, 'Password Must Contain At Least 6 Characters,');
30   });
31   const dispatch = useDispatch();
32   const [isArabic, setIsArabic] = useState(lang);
33
34   return (
35     <View style={styles.container}>
36       <Header navigation={navigation}>{Traduction[isArabic].title}</Header>
37       <TouchableWithoutFeedback onPress={() => Keyboard.dismiss()}>
38         <View style={styles.wrapper}>
39           <Formik

```

```

EXPL... HomeScreen.js
src > screens > HomeScreen.js
1 import {
2   Image,
3   SafeAreaView,
4   ScrollView,
5   StatusBar,
6   StyleSheet,
7   Text,
8   View,
9 } from 'react-native';
10 import React, {useState} from 'react';
11 import {Displayer} from '../utils';
12 import {Colors, Waves, Images, Fonts} from '../constants';
13 import {Box, Btn} from '../components';
14 import Traduction from '../translate';
15 import GeneralStorage from '../Store/Storage/GeneralStorage';
16 import {useDispatch, useSelector} from 'react-redux';
17 import GeneralAction from '../Store/Actions/GeneralAction';
18
19 const {setWidth, setHeight} = Displayer;
20 const logoSize = 35;
21
22 const HomeScreen = () => {
23   const dispatch = useDispatch();
24
25   const toServicesScreen = () => {
26     GeneralStorage.setFirstTimeUse().then(() => {
27       dispatch(GeneralAction.setIsFirstTimeUse());
28     });
29   };
30   const [lang] = useSelector(state => state.Language);
31

```

```

EXPL...  src > navigation > index.js > ...
1  import React, {useEffect, useState} from 'react';
2  import {NavigationContainer} from '@react-navigation/native';
3  import {createNativeStackNavigator} from '@react-navigation/native-stack';
4  import {useSelector} from 'react-redux';
5  import {
6    HomeScreen,
7    SplashScreen,
8    ServicesScreen,
9    FormScreen,
10   SendScreen,
11   LogInScreen,
12   SingUpScreen,
13   SettingScreen,
14   UserInfo,
15   Camera,
16 } from '../screens';
17 // import GeneralAction from '../Store/Actions/GeneralAction';
18
19 const Stack = createNativeStackNavigator();
20 export default function index() {
21   const [isLoading, setIsLoading] = useState(true);
22   useEffect(() => {
23     setTimeout(() => {
24       setIsLoading(false);
25     }, 5000);
26     return;
27   }, []);
28
29   const {isFirstTimeUse, uid} = useSelector(state => state.Language);
30   console.log(uid);
31   return (
32     <NavigationContainer>
33     <Stack.Navigator screenOptions={{headerShown: false}}>
34       <Stack.Screen name="Splash" component={SplashScreen} />
35     </Stack.Navigator>
36   );
37   // if (isFirstTimeUse) {
38   //   <Stack.Screen name="Home" component={HomeScreen} />
39   // }

```

```

EXPL...  src > translate > fr.js > default > Tel
1  export default {
2    // Home
3    homescreen: 'La Maintenance avec efficience et réactivité',
4
5    Plmo: 'Plomberie',
6    Elec: 'Électricité',
7    Manui: 'Menuiserie',
8    clima: 'Climatisation',
9
10   // AUTH
11
12   title2: 'Création de compte',
13   title3: 'Connexion',
14   FullName: 'Nom Complet',
15   Tel: 'Téléphone',
16   Setting: 'Changer de langue',
17   Save: 'Enregistrer',
18   BTM1: 'Créer un compte',
19   Email: 'Adresse Mail',
20   Camera: 'Appareil photo',
21   galerie: 'galerie',
22   imageProb: 'Image de problème',
23   chosePic: 'Choisissez une photo',
24
25   BTM2: 'Connexion',
26   btn4: 'Suivant',
27   LOGOUT: 'Déconnecter',
28
29   // Prop
30   title1: 'Réparons votre problème',
31 }

```