



المدرسة العليا
للتكنولوجيا - الصويرة
L'ÉCOLE SUPÉRIEURE DE
TECHNOLOGIE – ESSAQUIRA

*Filière : Génie Informatique
Option : Génie logiciel et réseaux*

RAPPORT DE PROJET DE FIN D'ETUDE

**Application mobile de gestion d'absence en utilisant
le QR code**

Réalisée par : LAQDOUR Abdelghani
YACOUBI Abdelhakime

Encadré par : Mme. ZIAD Lamia
Mr. GOUNANE Said
Mr. KARAMI Fahd

Année universitaire 2019-2020

Remerciements

Nous tenons à exprimer toute nos reconnaissances à notre encadrent Madame ZIAD Lamia, nous la remercions de nous avoir encadré, orienté, aidé et conseillé.

Nous voudrions exprimer nos reconnaissances envers Monsieur GOUNANE Said et Monsieur KARAMI Fahd qui nous apporté leur support moral et intellectuel tout au long de nos démarche.

Nous adressons nos sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leur parole, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et répondre à nos questions durant nos recherches.

Nous remercions également les membres de jury d'avoir accepté de juger notre travail.

Enfin, nous remercions tous nos ami(e)s et collègues, pour leur sincère amitié et confiance, et qui nous doit notre reconnaissance et nos attachements.

Sommaire

Remerciement.....	02
Liste des Figures	04
Introduction	06
CHAPITRE I : Contexte générale du projet.....	07
I Problématique et objectif.....	08
II Planning du projet.....	09
CHAPITRE II : Analyse et Conception du projet.....	10
I Benchmarking.....	11
II Le cahier de charge.....	13
1. Phase de démarrage.....	13
2. Phase de lancement.....	13
III La conception du projet.....	14
1. Diagramme de cas d'utilisation.....	14
2. Diagramme de séquence.....	14
3. Diagramme de classe.....	15
CHAPITRE III : Environnement et outils de travail.....	16
I Framework.....	17
II Les langages de programmation.....	20
III Les outils de travail.....	21
CHAPITRE IV : La réalisation du projet.....	22
I Les étapes de la réalisation.....	23
II L'application.....	32
Conclusion.....	52
Webographie.....	53

Listes des figures

Figure 1 : Diagramme de GANTT :.....	09
Figure 2 : L'application 'EMSI' :... ..	11
Figure 3 : L'application 'Student information system'. :... ..	12
Figure 4 : Diagramme de cas d'utilisation :	14
Figure 5 : Diagramme de séquence. :.....	14
Figure 6 : Diagramme de classe :.....	15
Figure 7 : La fonction connectPresed(). :... ..	23
Figure 8 : La fonction setId(). :.....	23
Figure 9 : La fonction httpLogin :	24
Figure 10 : La fonction setEtd(). :.....	24
Figure 11 : La requête get pour l'authentification :	25
Figure 12 : La fonction ajouter(). :.....	26
Figure 13 : La fonction scan(). :.....	27
Figure 14 : La fonction checkAbsence(). :... ..	28
Figure 15 : La fonction de vérification de la présence :	29
Figure 16 : La fonction changeState() :.....	29
Figure 17 : La fonction de traduction :.....	30
Figure 18 : La fonction themes().	31
Figure 19 : La fonction ChangeThemes().	31
Figure 20 : Splash Screen :... ..	32
Figure 21 : Page authentification :.....	33
Figure 22 : Page paramètre (avant se connecter) :.....	34

Figure 23 : Page pour changer la langue de l'application :.....	35
Figure 24 : Le changement de mode de l'application au mode nuit. :.....	36
Figure 25 : Le Tiroir de l'application (drawer) :... ..	37
Figure 26 : La page assistance :... ..	38
Figure 27 : La page FAQ. :.....	39
Figure 28 : Le message des informations incorrectes. :.....	40
Figure 29 : Le message de saisir les données nécessaires. :.....	41
Figure 30 : La Page d'accueil :... ..	42
Figure 31 : La page guidance :.....	43
Figure 32 : La page paramétrer (après se connecter) :.....	44
Figure 33 : La page profil. :.....	45
Figure 34 : Demande d'autorisation pour utiliser la caméra :.....	46
Figure 35 : Scanner de QR Code :.....	46
Figure 36 : L'autorisation de caméra refuse. :.....	47
Figure 37 : Message si vous avez sortie avant de scanner le QR code :.....	48
Figure 38 : Message si vous avez scannez le QR avec succès :.....	49
Figure 39 : Message si vous avez scannez un QR Code pas correcte :.....	50
Figure 40 : Le message si vous avez scannez un ancien QR code :.....	51

Introduction

Naturellement, la plupart des écoles au Maroc rencontré beaucoup des problèmes, à cause d'emploi de la méthode manuelle de la gestion d'absence comme l'utilisation des papiers ou des logiciels (Excel, Word...etc.), qui permette de gérer l'absence d'une façon mal structuré et très lent, au moment de la séance. Et le perde de temps est le principal problème dans ce cas-là.

Et dans notre école ESTE ne dispose d'aucun logiciel informatique ou application mobile permettant la gestion des absences. Le service accompli ses tâches de gestion d'une manière presque manuelle en utilisant des logiciels de bureautique. Ceci rend leurs tâches fastidieuses et difficiles, et le travail compliqué.

C'est pour cela nous avons proposé l'idée de la création d'une application mobile de gestion d'absence avec l'utilisation de QR code, qui va générer par une autre application web réalisé par deux développeur, pour gérer l'absence d'une façon professionnelle. Et plus rapide. Et donnée une valeur ajoutée à notre école. Grâce à l'utilisation de la technologie.

Dans ce rapport, on va expliquer l'étape de la réalisation de cette l'application en quatre chapitres :

- Le chapitre I : Contexte générale du projet.
- Le chapitre II : Analyse et conception du projet.
- Le chapitre III : Environnement et outils de travail.
- Le chapitre IV : La réalisation du projet.

Chapitre I :

Contexte générale du projet

I. Problématique et objectif

• Problématique

La gestion des absences c'est une tâche principale pour les profs et aussi l'administration, mais cette dernière il exige plus de temps avec la méthode classique, ce qui conduit à ne pas le faire.

Et Parmi les problèmes qui nous avons vu au sien de notre école:

- **Problème de temps** à cause de le grand nombre d'étudiants.
- **Problème d'organisation.**
- **Problème de la sécurité** : n'importe quelle personne peut accéder aux informations et les modifier.
- **Problème de gestion de l'information** : il n'y a pas de base de données pour stocker les absences de chaque séance ce qui peut mener à une perte d'information.

• Objectif

Afin de remédier aux problèmes présentés dans la section précédente, nous avons proposé la création d'une application qui permettre de :

- Faciliter la tâche.
- Gagner le temps.
- Diminuer les risques d'erreur.
- Gérer l'absence d'une façon professionnelle et plus rapide.
- Garantir la sécurité (Authentification).

II. Planning du projet

La planification du projet est l'activité qui consiste à déterminer et ordonnancer les tâches du projet, à estimer leurs charges et à déterminer les profils nécessaires à leur réalisation.

Le diagramme de Gantt présenté ci-dessous jouait le rôle d'un fil conducteur tout au long du projet. Il nous a permis de maîtriser la gestion du temps alloué au projet.

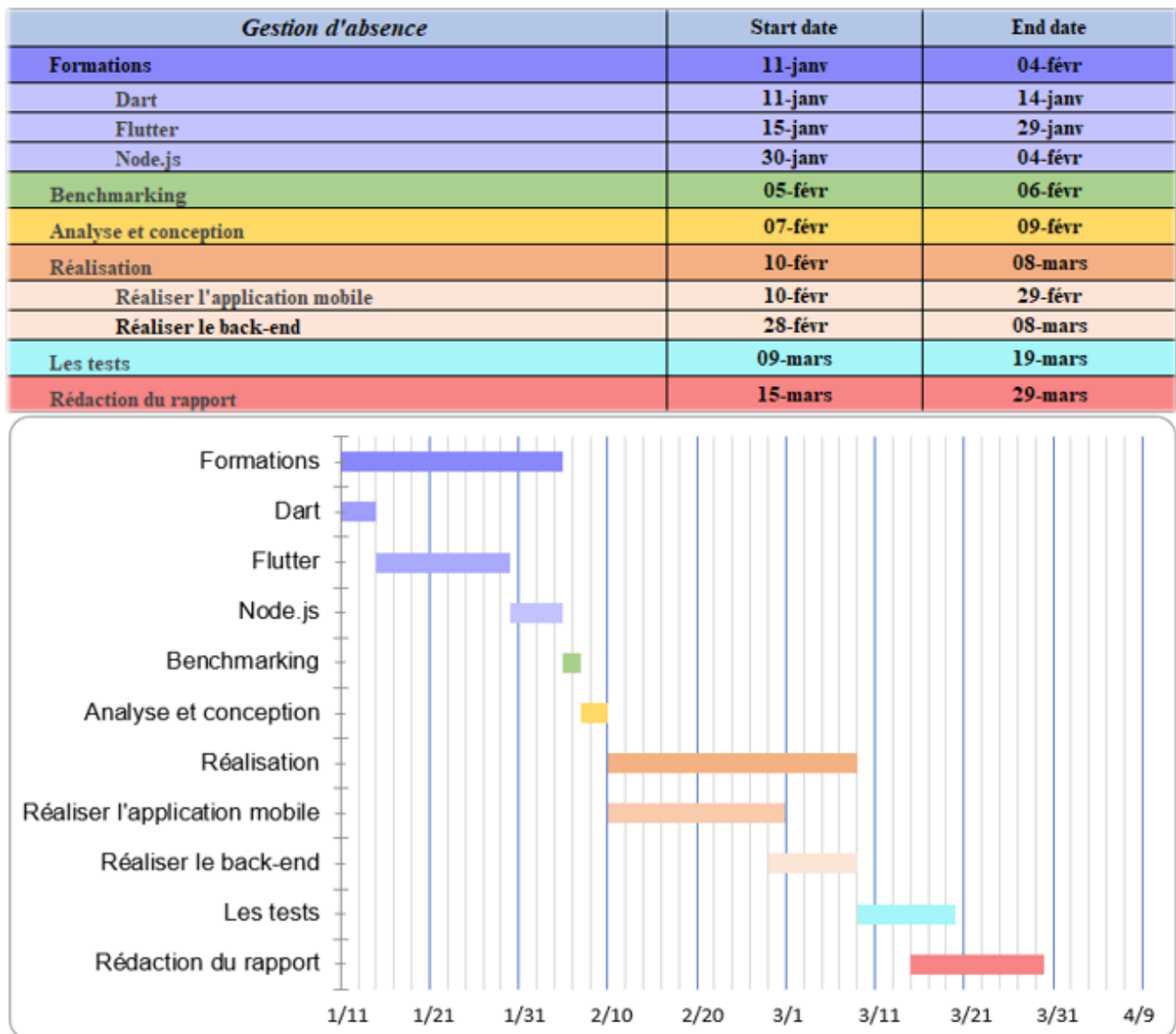


Figure 1 : Diagramme de GANTT.

Chapitre II :

Analyse et conception du projet

I. Beanchmarking

Après nous recherche sur internet, nous avons trouvé beaucoup des applications qui font la même fonctionnalité que notre application. Parmi ces applications on va prendre l'exemple de deux applications :

1. Application EMSI

C'est une application développée par les jeunes de l'école marocaine des sciences d'ingénieur.



Figure 2 : l'application EMSI

❖ Les points faibles de l'application :

Cette application utilisé le **barcode** pour gérer l'absence, qui permette de stocker des informations limitées. Par rapport notre application utilisé le QR code comme une moyen qui permette de stocker une grande quantité des informations. Comme ça on va utiliser n'importe quel type d'information.

Cette application développée pour les appareils **Android**. Ici on va voir un problème si un étudiant il a un appareil **IOS**. Par rapport notre application développer avec l'utilisation de Framework Flutter, comme une Framework multiplateforme (un seul code va marcher sur les deux plateformes IOS & Android).

2. Application Student information System

C'est une application développée par les jeunes d'une école étrangère.



Figure 3 : l'application Student information system.

❖ Les points faibles de l'application

- Cette application aussi développée pour les appareils Android.
- Et aussi utiliser le barcode pour la gestion d'absence.



II. Le cahier de charge

1. Phase de démarrage

La création d'une application mobile utilisée par les étudiants pour enregistrer l'absence d'une façon professionnelle et plus rapide et Pour gagner le temps.

L'application utilisé le QR code qui permette de stocker une grande quantité des informations (de la séance).

2. Phase de lancement

Après les remarques des étudiants de l'école, nous n'avons conclus que notre application a des points faibles, qu'on a essayé de dépasser à l'aide d'une autre application web. Responsable sur la génération de le QR code. Parmi les points faibles.

Si l'étudiant ne possède pas de smartphone ou possède un téléphone mais ne parvient pas à scanner le QR code à distance ici on a un problème.

Mais grâce à l'application web réaliser par un autre binôme on va gérer l'absence des étudiants qui rencontrer ces problèmes-là.

III. La conception du projet

1. Diagramme de cas d'utilisation

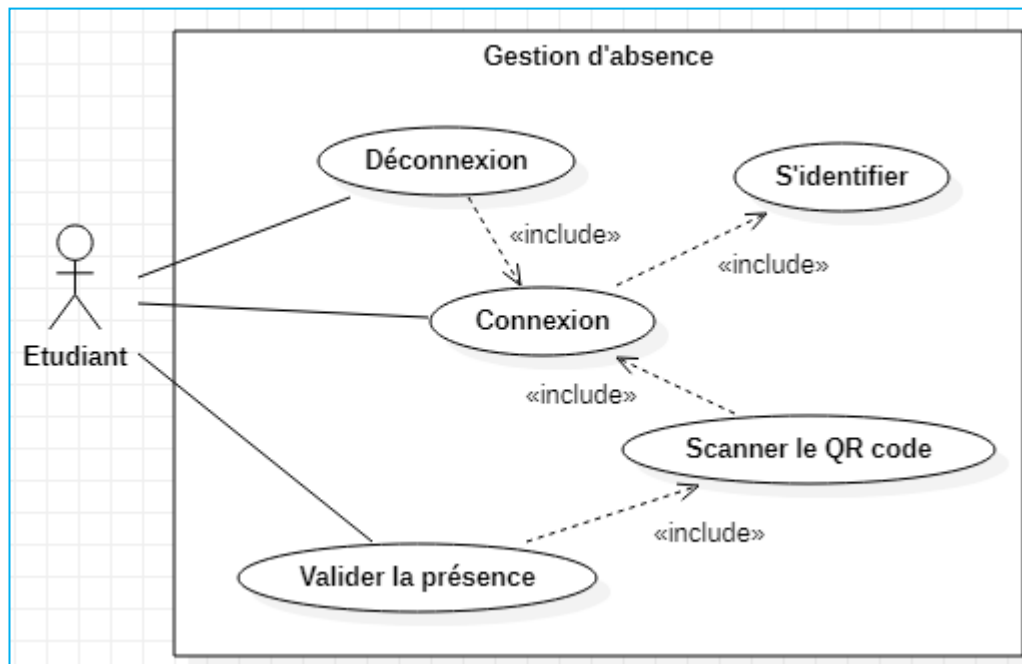


Figure 4 : diagramme de cas d'utilisation.

2. Diagramme de séquence

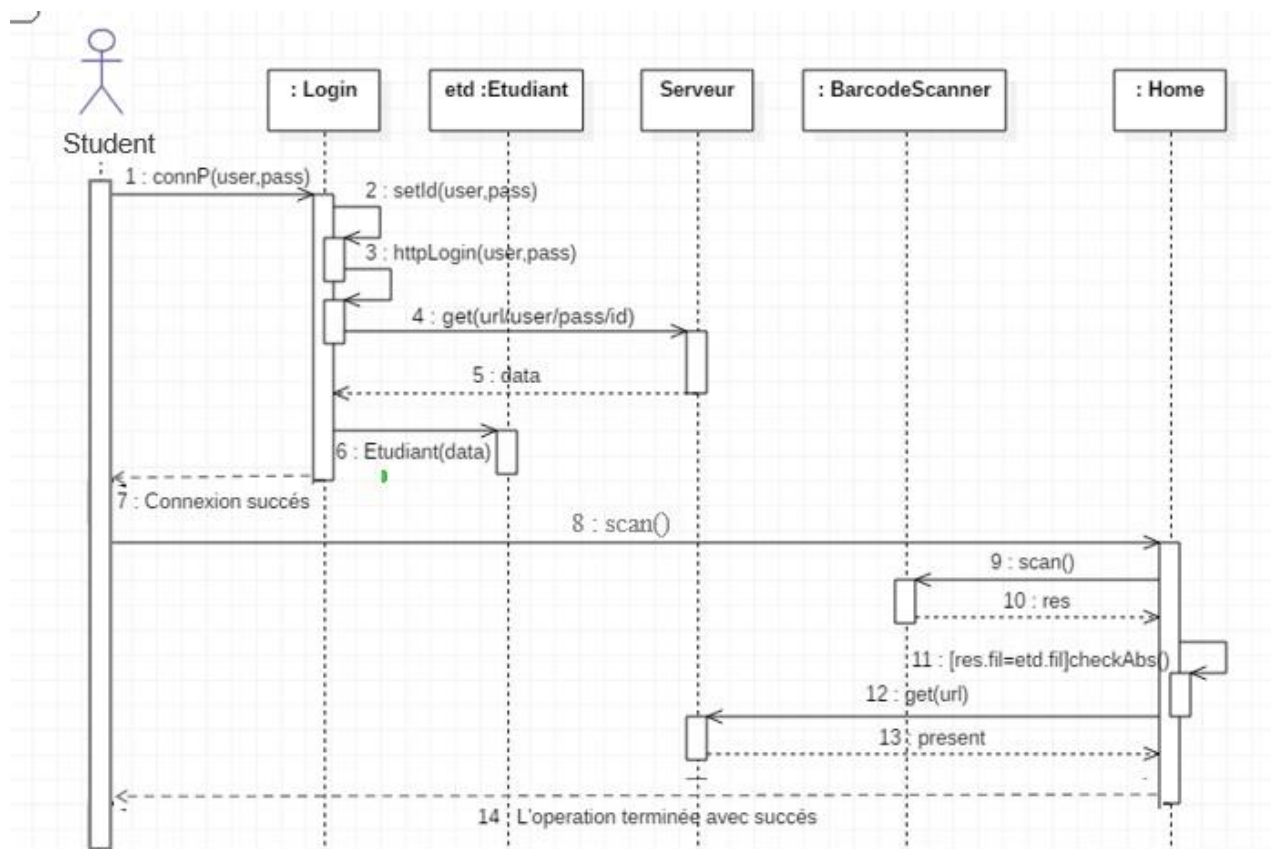


Figure 5 : diagramme de séquence.

3. Diagramme classe

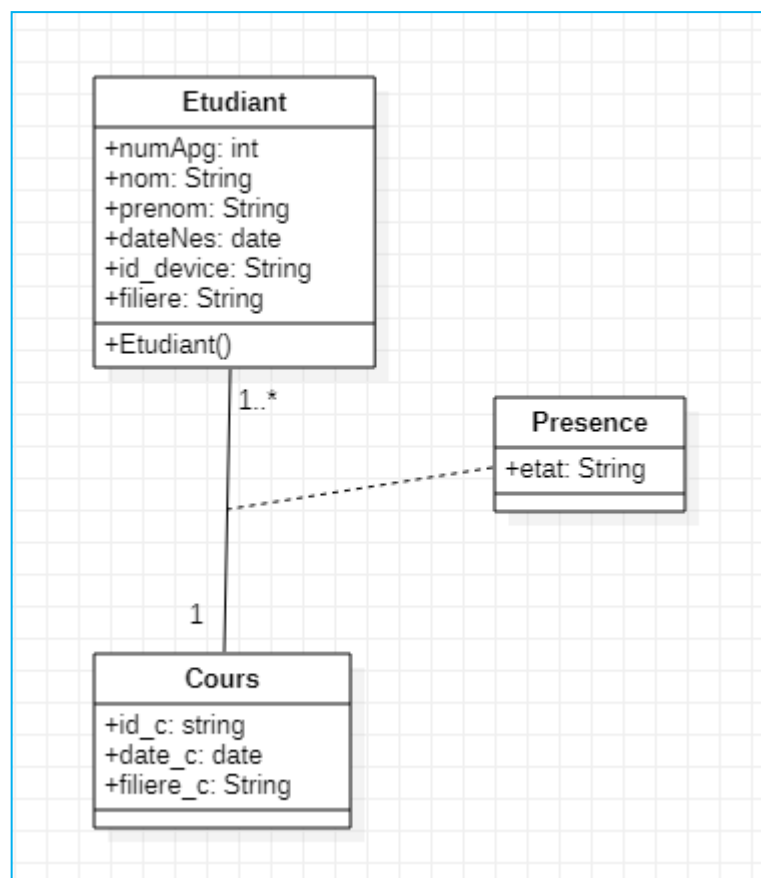


Figure 6 : diagramme de classe

Chapitre III :

Environnement et outils de travail

I. Framework

1. Flutter

Flutter c'est un Framework basé sur langage Dart (on va le voir ci-dessous) gratuit et open source de développement natif multiplateforme créé par Google. Flutter prend en charge le mobile (Android & iOS), le desktop (Windows, Mac OS, Linux, etc), les périphériques intégrés (Raspberry Pi, Google Home Hub etc.) Et le web.



Flutter est encore jeune mais il a déjà commencé à faire ses preuves. Le Framework a déjà séduit certains grands noms de la Tech : Alibaba, le géant du e-commerce chinois l'a utilisé pour son application Xianyu. Les applications comme Reflectly, Hamilton Musical et bien d'autres utilisent Flutter. Google l'utilise également dans ses produits (Google Ads et Google Greentea).

➤ Les bibliothèques utilisées

1. Barcode_Scan

L'accès à la caméra et la numérisation sont ces fonctionnalités natives dont la plupart des applications hautement évoluées ont besoin. Les Barcode et les QR codes sont les codes officiels de reconnaissance des produits et sont également utilisés dans divers aspects industriels. Ainsi, travailler avec des Barcodes et des QR code peut être très pratique pour la gestion des données au moyen de code uniques.

Le Framework Flutter nous aide à travailler avec les codes car il a un package appelé barcode_scan.

Et pour l'installation vous devez simplement ajoutez l'autorisation de caméra à votre AndroidManifest.Xml pour Android.

```
<uses-permission android:name="android.permission.CAMERA" />
```

Et pour IOS vous devez ajouter la description d'utilisation de la caméra à votre info.plist

```
<key>NSCameraUsageDescription</key>  
<key>UISupportedInterfaceOrientations</key>
```

Finalement ajouter ce package à votre projet en ajoutant la bibliothèque à votre fichier pubspec.yaml.

```
dependencies:  
  flutter:  
    sdk: flutter  
  barcode_scan: ^1.0.0
```

Après, vous devez taper la commande **flutter pub get** dans votre terminal ou faire juste **ctrl + S**

2. Device_info

Cette bibliothèque nous permet de obtenir toutes les informations actuelles comme **Board, Brand, Host, Device Id, Model, Manufacture...**etc. sur l'appareil à partir de l'application Flutter.

Et pour l'installation vous devez ajouter ce package à votre projet en ajoutant la bibliothèque à votre fichier pubspec.yaml.

```
dependencies:  
  flutter:  
    sdk: flutter  
  barcode_scan: ^1.0.0  
  device_info: ^0.4.2+1
```

Après, vous devez taper la commande **flutter pub get** dans votre terminal ou faire **ctrl + S**.

3. http

Ce package contient un ensemble de fonctions et de classes de haut niveau qui facilitent la consommation des ressources http. Il est indépendant de la plate-forme et peut être utilisé à la fois sur la ligne de commande et le navigateur.

Et ce package contient cinq fonctions principales :

- [GET](#) : est utilisé pour obtenir des ressources.
- [PUT](#) : est destiné à remplacer une ressource ou à la créer si elle n'existe pas.
- [POST](#) : est utilisé pour créer une nouvelle ressource.
- [PATCH](#) : est destiné à modifier une ressource existante.
- [DELETE](#) : est bien sûr pour supprimer des ressources.

Concernant l'installation est comme les autres bibliothèques.

4. video_player

Est une bibliothèque flutter pour IOS et Android nous permet de visualiser des vidéos, et pour l'installation est tout simplement comme les autres bibliothèques.

2. NodeJS

Node.js sert à faire du JavaScript server Side, et peut être utilisé dans des applications de bases de données, la plus populaire étant MongoDB, Ce n'est ni un Framework ni un server, Node.js est souvent confondu avec ce dernier, car c'est sa base :



Créer des applications en temps réel, ou le server à la possibilité de transmettre de l'information au client il est comparé à Python, Ruby, Java ou encore PHP.

Node.js à de nombreux avantages :

- Système de paquet intégré (NPM).
- Modèle non bloquant.
- Performance du moteur V8, logiciel libre (licence MIT).

II. Les langages de programmation

1. UML

Le langage UML (unified Modeling language) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.



L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet.

2. Dart

Dart est un langage de programmation structuré, open source et évolutif qui fonctionne à l'aide d'une machine virtuelle, le projet de Google à travers le Dart et de faciliter le développement web, de combler les carences du JavaScript et d'offrir de meilleure performance, surtout pour les gros projets.



Le Dart est destiné à fonctionner sur les navigateurs web modernes ainsi que sur les serveurs et il embarque également un convertisseur JavaScript, Il y a plusieurs manières d'exécuter du code Dart soit en utilisant un navigateur qui supporte directement ce code, soit en compilant. En plus du navigateur web et de la conversion JavaScript le code peut être exécuté en ligne de commande, hébergé dans une machine virtuelle ce qui permet au client et au serveur d'avoir les applications écrites dans le même langage.

3. Json

JSON (JavaScript Object Notation) est un format d'échange de données en texte lisible. Il est utilisé pour représenter des structures de données et des objets simples dans un code qui repose sur un navigateur Web.



III. Les outils de travail

1. Visual studio Code

Visual studio code est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multiplateforme, open source et gratuit, supportant une dizaine de langages.



2. StarUML

StarUML est un logiciel de modélisation UML, qui a été cédé comme open source par son éditeur, à la fin de son exploitation commercial (qui visiblement continue), sous une licence modifiée de GNU GPL.



3. Android Studio

Android Studio est l'environnement de développement intégré (IDE) officiel du système d'exploitation Android de Google, construit sur le logiciel JetBrains IntelliJ IDEA et conçu spécifiquement pour le développement Android. Il est disponible en téléchargement sur les systèmes d'exploitation Windows, MacOS et Linux. Il remplace les outils de développement Eclipse Android (ADT) en tant qu'IDE principal pour le développement d'applications natives Android.



4. Git

Git est un système de contrôle de version distribué pour suivre les modifications du code source pendant le développement du logiciel. Il est conçu pour coordonner le travail entre les programmeurs, mais il peut être utilisé pour suivre les modifications dans n'importe quel ensemble de fichiers. Ses objectifs incluent la vitesse, l'intégrité des données et la prise en charge des workflows distribués et non linéaires.



- **GitHub**

Est un outil gratuit pour héberger notre code, qui utilise Git, Git est un gestionnaire de versions des fichiers qui permet d'enregistrer les modifications faites sur le même fichier. C'est le numéro 1 mondial et il héberge plus d'une dizaine de millions de repositories !



Chapitre IV : La réalisation

I. Les étapes de la réalisation

1. L'authentification

➤ Partie flutter

❖ La fonction connectPresed()

Lorsque de la connexion on va appeler la fonction connectPresed qui vérifie les informations s'ils ne sont pas vides, directement on va appeler la fonction setId().

```
void connectPresed(String userid,String passid,context){
    if(userid.isEmpty || passid.isEmpty){
        emptyInfoDialog(context);
        user.clear();
        passe.clear();
    }else{
        setState(() {
            startMoving = true;
        });
        setId(userid, passid);
    }
}
```

Figure 7 : la fonction connectPresed().

❖ La fonction setId()

La fonction setId() va chercher sur l'id device de téléphone, ça dépend leur plateforme(soit IOS ou Android). Et après il va appeler la fonction httpLogin() avec trois argument (le nom d'utilisateur, le mot de passe et l'id device).

```
void setId(String user, String passe) async{
    DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
    if (Theme.of(context).platform == TargetPlatform.iOS) {
        /*-----IOS-----*/
        IosDeviceInfo ios = await deviceInfo.iosInfo;
        setState(){ id_dev = ios.identifierForVendor.toString(); });
    } else {
        /*-----Android-----*/
        AndroidDeviceInfo android = await deviceInfo.androidInfo;
        setState(){id_dev = android.androidId.toString();});
    }
    httpLogin(user,passe,id_dev);
}
```

Figure 8 : la fonction setId().

❖ La fonction httpLogin()

La fonction httpLogin permet de envoyer une requête http avec la méthode get pour obtenir les informations d'étudiant mais avant cela on va voir :

- Si la réponse est égale à 'true' on va appeler la fonction setEtd et compléter l'authentification.
- Si est égale à 'id_x' donc l'étudiant est en train d'utiliser un autre téléphone.
- Si est égale à 'false' il signifie que les informations d'authentification sont incorrectes.

```
httpLogin(String userid,String passid, String idDevice)async{
    var newurl =url+"user/"+userid+"/"+passid+"/"+idDevice;
    try{
        Response res = await get(newurl);
        if(res.statusCode == 200){
            var data = json.decode(res.body);
            if(data["rep"] == 'true'){
                setEtd(data);
                Navigator.push(// enter to Home
                    context,
                    new MaterialPageRoute(builder: (context) => new Home())
                );
            }else if (data["rep"] == 'id_x'){
                setState((){ startMoving = false; });
                BodyHomestate.showDialog(context,pas_egale,AlertType.warning);
            }else{
                setState((){ startMoving = false; });
                incorrectInfoDialog(context);
            }
        }else{
            BodyHomestate.showDialog(context,err_server,AlertType.error);
        }
    }on SocketException catch(_){
        setState((){ startMoving = false; });
        noConnectionDialog(context);
    }
}
```

Figure 9 : la fonction httpLogin.

❖ La fonction setEtd()

La fonction setEtd() affecter les informations de la requête get à l'objet « etd » de la classe Etudiant.

```
void setEtd(var data){
    setState() {
        etd = Etudiant(
            data["info"][0]["nom"],
            data["info"][0]["prenom"],
            data["info"][0]["numApg"],
            data["info"][0]["Date_naissance"],
            data["info"][0]["filiere"],
            data["info"][0]["id_device"]
        );
        startMoving = false;
    });
}
```

Figure 10 : la fonction setEtd().

➤ Partie Node.js

On a créé une route « /user/:user/:pass/:id » à l'aide de la méthode GET, qui va faire l'authentification avec le numéro d'apogée, la date de naissance et l'id device qui sont passés par url.

Premièrement nous avons établi la connexion avec la base de données **Etudiant**. Et après sélectionner cet étudiant à la collection étudiantes s'il existe, et vérifier leur id device.

Si son id device dans la base de données est vide on va appeler la fonction **ajouter**, qui permet d'ajouter l'id device passé par url dans la base de données, Et on va envoyer la réponse 'true' plus toutes les informations d'étudiant se format de Json. Sinon on va comparer id device dans la base de données avec notre id device passé par url, si nous avons une déference entre les deux, on va transmettre la réponse 'id_x'. Sinon on va transmettre la réponse 'true' plus toutes les informations d'étudiant concerné.

Finalement on va envoyer la réponse 'false', si les informations d'authentification sont incorrectes.

```
app.get('/user/:user/:pass/:id',(req,res)=>{
  var user = req.params.user;
  var pass = req.params.pass;
  var id = req.params.id;
  var quiry = {numApg: user , Date_naissance: pass};
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("Etudiant");
    dbo.collection("etudiants").find(quiry).toArray(function(err, result) {
      if (err) throw err;
      if(result.length > 0){
        if(result[0]['id_device'] == ""){
          ajouter(user, id,dbo);
          res.json({rep:'true',info:result});
        }else{
          if(id == result[0]['id_device']){
            res.json({rep:'true',info:result});
          }else{
            res.json({rep:'id_x'});
          }
        }
      }
      }else{
        res.json({rep:'false'});
      }
      db.close();
    });
  });
});
```

Figure 11 : la requête get pour l'authentification.

❖ La fonction ajouter()

```
function ajouter(num , id_dev,dbo){  
  var quiry = {numApg : num};  
  var setvalue = {$set : {id_device : id_dev}};  
  dbo.collection("etudiants").updateOne(quiry, setvalue, function(err){  
    if(err) throw err;  
  });  
}
```

Figure 12 : La fonction ajouter().

2. Le scan du QR Code

➤ Parties flutter

❖ La fonction scan()

Pour scanner un QR code on va utiliser la fonction scan qui va scanner et lire les informations du QR code, premièrement on va comparer la filière d'étudiant et du QR code si sont les mêmes, on va appeler la fonction checkAbsence.

Dans cette fonction on à traiter les exceptions suivantes :

- **PlatformException** : Cette exception va être déclenché si on à pas autorisé l'Access au camera.
- **FormatException** : Exception levée lorsqu'une chaîne ou d'autres données n'ont pas un format attendu et ne peuvent pas être analysées ou traitées.

```
void scan() async {
  try{
    resScan = await BarcodeScanner.scan();
    var data = json.decode(resScan);
    var date = data["date_sc"];
    var filiere = data["filiere"];
    var id_sc = data["id_sc"];
    if(filiere == etd.filiere){
      var urlAbs = url+"present/"+date+"/"+filiere+"/"+id_sc+"/"+etd.numApg;
      print("url :"+urlAbs);
      checkAbsence(urlAbs,context);
    }else{
      showDialog(context, noFil ,AlertType.error);
    }
  }on PlatformException catch(ex){
    if(ex.code== BarcodeScanner.CameraAccessDenied){
      showDialog(context, denie_camera ,AlertType.error);
    }else{
      showDialog(context, "$ex" ,AlertType.error);
    }
  }on FormatException {
    if(resScan.isNotEmpty){
      showDialog(context, scane_autre_qr ,AlertType.warning);
    }else{
      showDialog(context, back_befor_scane ,AlertType.warning);
    }
  }catch(ex){
    showDialog(context, "$ex" ,AlertType.error);
  }
}
```

Figure 13 : la fonction scan().

❖ La fonction `checkAbsence()`

La fonction `checkAbsence` permet de transmettre les informations au fichier Node.js pour les traiter et renvoyer une réponse soit 'present' ou 'absence'.

```
checkAbsence(urlAbs,context)async{
  Response result;
  try{
    result = await get(urlAbs);
    if(result.statusCode == 200) {
      var data = json.decode(result.body);
      if(data["rep"]=="present"){
        showDialog(context, present_ok ,AlertType.success);
      }else{
        showDialog(context, present_no ,AlertType.warning);
      }
    }else{
      BodyHomestate.showDialog(context,err_server,AlertType.error);
    }
  }on SocketException catch(_){
    showDialog(context, noConn ,AlertType.info);
  }
}
```

Figure 14 : la fonction `checkAbsence()`.

➤ Parties Node.js

Cette une fonction permette de confirmer la presence des étudiants en utilisant la date de la séance, le filiere, l'id de la séance et le numéro d'apogée de l'étudiant qui sont passés par URL.

Après la connexion avec la base de données, on va vérifier l'état d'étudiant concerné. Si l'état 'present' on va envoyer la réponse 'present' ce format de Json. Sinon on va exécuter la fonction `changeState()`, qui permet de remplacer l'état absence par present. Et on va envoyer la réponse 'present' se format de Json a l'application flutter.

Et l'envoi de la réponse 'false' il signifie soit la séance n'existe pas dans la base de données (à cause de scanné une QR code incorrect) ou existe mais l'id de séance a été modifier (id va modifier chaque trois second).

```

app.get('/present/:date_sc/:filiere/:id_sc/:numApg',(req,res)=>{
  var date_sc = req.params.date_sc;
  var id_sc = req.params.id_sc;
  var filiere = req.params.filiere;
  var numApg = req.params.numApg;
  var quiry = {"date":date_sc,"filiere":filiere,"qr":id_sc,"etudiants.numApg":numApg};
  var setvalue = {$set:{"etudiants.$.etat":"present"}};
  var fields = {fields: {_id:0,"etudiants.numApg":1,"etudiants.$.etat":1}};
  MongoClient.connect(url, function(err, db) {
    if (err) throw err;
    var dbo = db.db("Etudiant");
    dbo.collection("cours").find(quiry, fields).toArray(function(err, result) {
      if (err) throw err;
      if(result.length>0){
        if(result[0]["etudiants"][0]["etat"] == "present"){
          res.json({rep:"present"});
        }else{
          changeState(quiry,setvalue,dbo);
          res.json({rep:"present"});
        }
      }else{
        res.json({rep:"false"}); //problem in the QRcode
      }
      db.close();
    });
  });
});
});

```

Figure 15 : la fonction de vérification de la présence.

❖ La fonction changeState()

```

function changeState(quiry,setvalue,dbo){
  dbo.collection("cours").updateOne(quiry,setvalue, function(err, result) {
    if (err) throw err;
    console.log(result.result.nModified + " present");
  });
  dbo.collection("cours").update(quiry,{ $inc:{nb_scans:1}}, function(err, result) {
    if (err) throw err;
    console.log(result.result.nModified + " increment");
  });
}

```

Figure 16 : la fonction changeState().

3. La fonction de traduction

Cette fonction permet de traduire tous les textes en Arabe, Français et Anglais dans notre application pour une meilleure compréhension.

```
void converte(String value){
    if(value == 'Francais'){
        setState(() { /*pour le Francais*/
            drawerParametre = "Paramètres";
            drawerExit = "Fermer";
            present_ok="L'opération terminée avec succès.";
            /*...*/
        });
    }else if(value == "العربية"){
        setState(() { /*pour l'Arabe*/
            drawerParametre = "الإعدادات";
            drawerExit = "أغلق";
            present_ok="تمت العملية بنجاح";
            /*...*/
        });
    }else{
        setState(() { //pour la page en English
            drawerParametre = "Settings";
            drawerExit = "Close";
            present_ok="the operation is well done.";
            /*...*/
        });
    }
}
```

Figure 17 : la fonction de traduction.

4. Le changement de thème

Dans notre projet on a travaillé sur deux thèmes différents, un mode standard (mode light) et un mode nuit pour un bon confort aux yeux, et une meilleure utilisation.

Et nous avons ici la fonction **themes()** qui permettent d'appeler la fonction **ChangeThemes()** qui va changer le thème de notre application.

❖ La fonction themes()

```
void themes(bool value){
    if(activier == false){
        setState(() {
            ThemesBody().ChangeThemes("dark");
        });
    }else{
        if(activier == true){
            setState(() {
                ThemesBody().ChangeThemes("light");
            });
        }
    }
}
```

Figure 18 : la fonction themes().

❖ La fonction ChangeThemes()

```
void ChangeThemes(var theme){
    if(theme == "dark"){
        activier = true;
        themes = ThemeData.dark();
        bgAppBarColor = Colors.grey[600];
        cId_pass = Colors.white60;
        chint = Colors.white54;
        /*....*/
    }else{
        activier = false;
        themes = ThemeData.light();
        bgAppBarColor = Colors.indigo;
        bgloginColor1 = Colors.blue[300];
        /*....*/
    }
}
```

Figure 19 : la fonction ChangeThemes().

II. L'application

1. Splash Screen

Lorsqu'un étudiant se connecte à notre application la première interface qui s'affiche c'est **Splash screen**. Qui contient le logo de notre application plus le bienvenu.

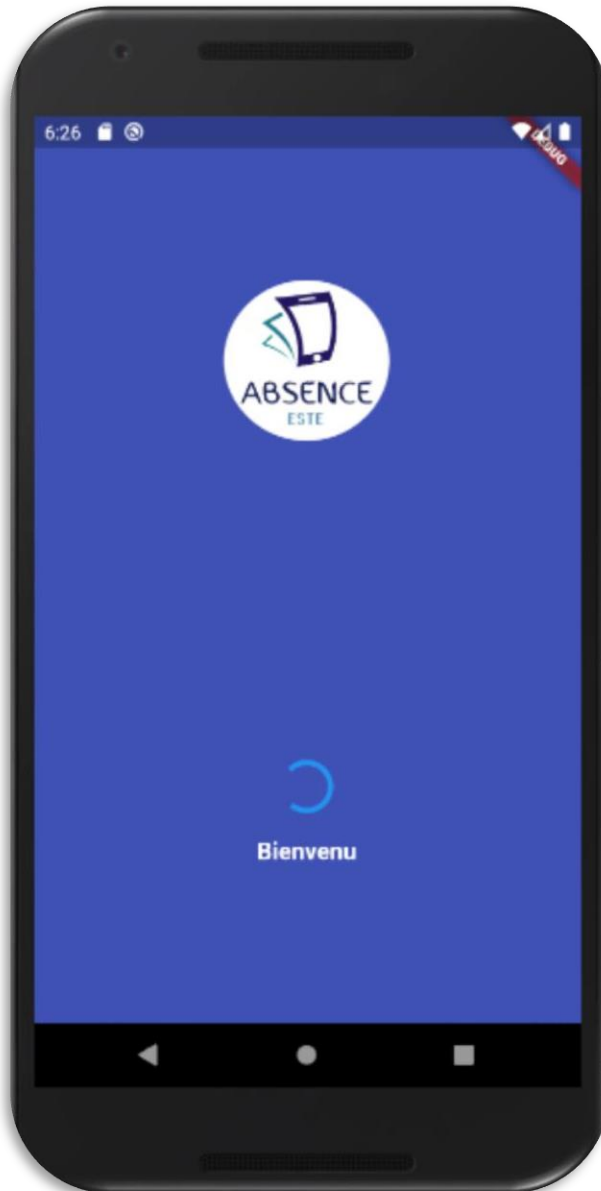


Figure 20 : première page

2. Page d'authentification

Dans cette interface, nous avons deux champs pour saisir le nom d'utilisateur et le mot de passe pour accès à l'application. Plus une icône Paramètre pour aller a les paramètre de l'application.

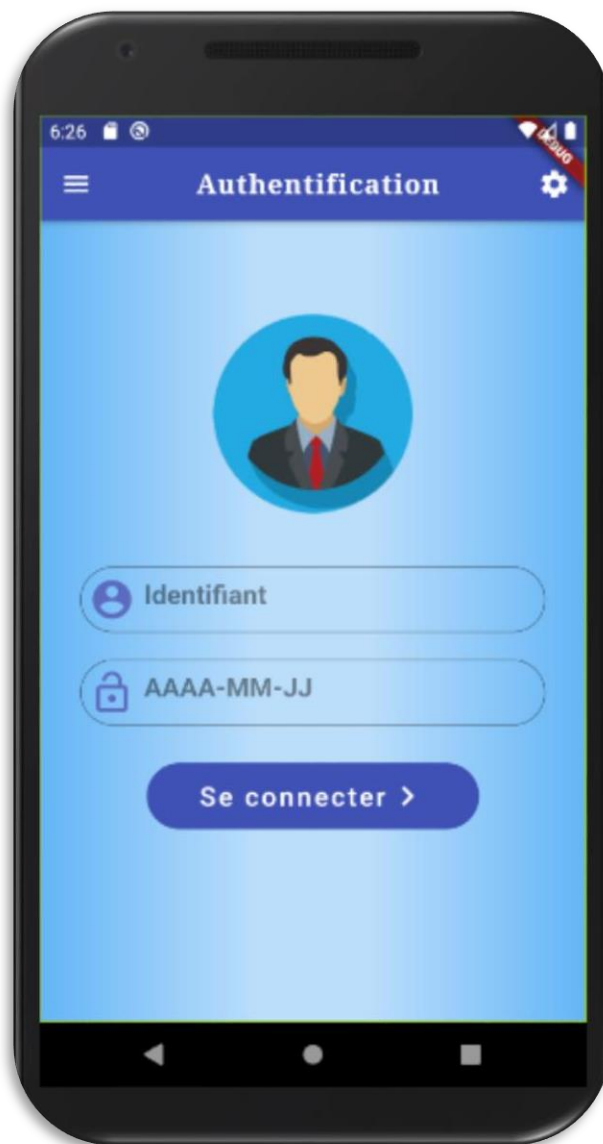


Figure 21 : page authentication

➤ Page de paramètre

C'est une interface qui contient un bouton pour changer la langue et autre pour changer le mode de l'application (mode light et mode nuit).

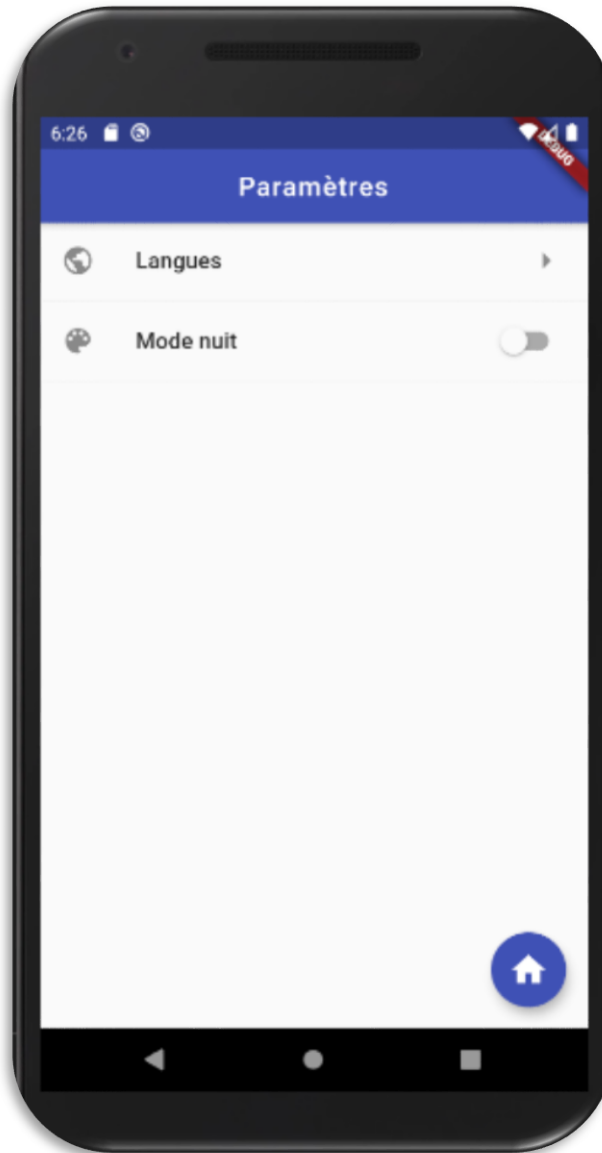


Figure 22 : page paramètre (avant se connecter)

➤ Page Langues

Si On clique sur le bouton langue on va voir une interface qui contient une liste de trois langues : Arabe, Français et Anglais.

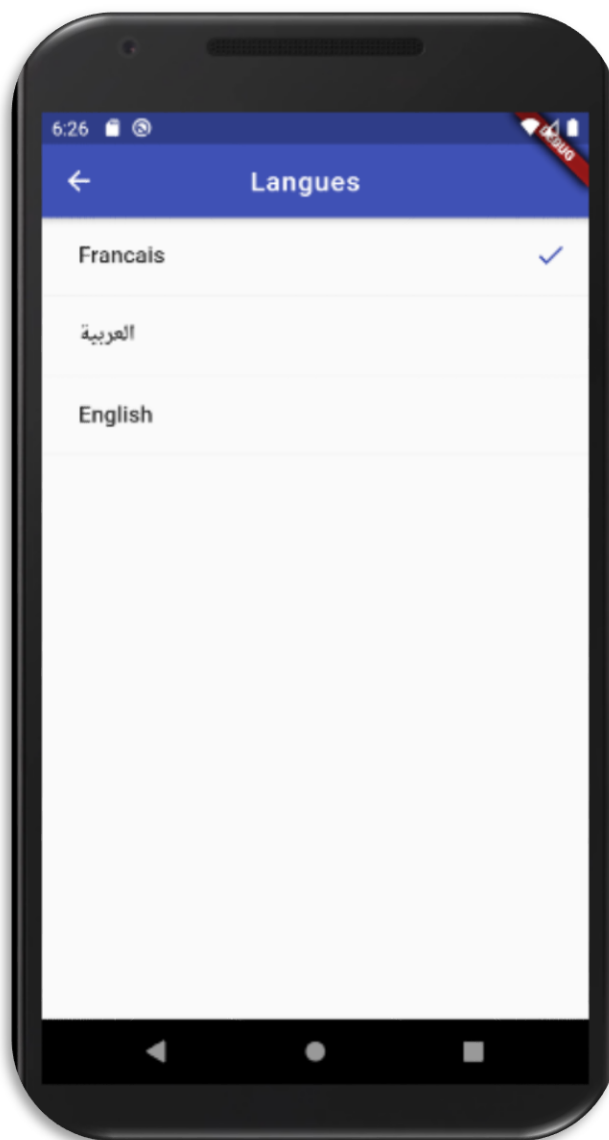


Figure 23 : page pour changer la langue de l'application.

➤ Activer le Mode nuit

Si nous avons activé le mode nuit, on va changer le mode de l'application.

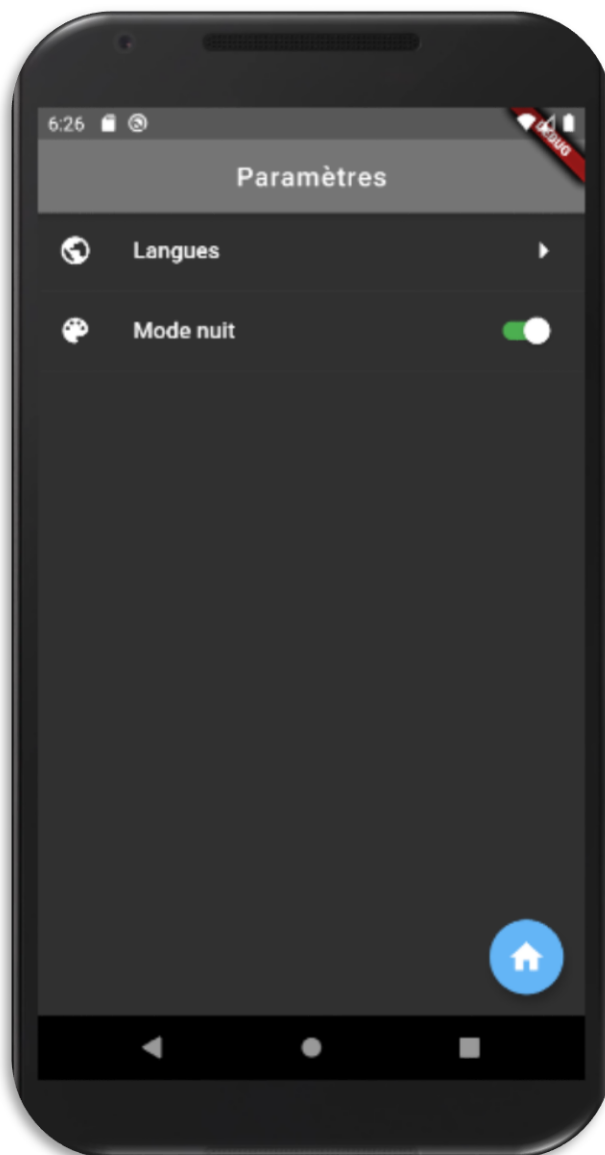


Figure 24 : le changement de mode de l'application au mode nuit.

➤ Drawer de page Authentification

Nous avons voir une Liste qui contient 4 bouton. Le premier pour accès à la page d'accueil et à la page d'assistance plus la page FAQ et un bouton pour fermer le drawer.

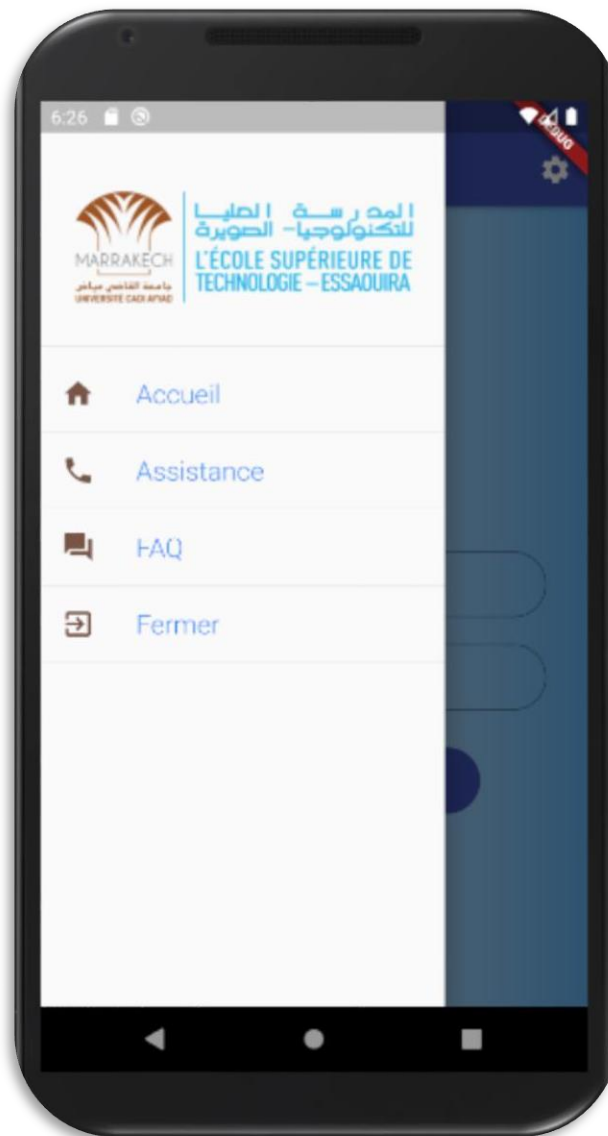


Figure 25 : tiroir de l'application (drawer)

➤ Page Assistance

Ici nous avons une interface qui contient les informations générales de l'école, plus de chaque filière, comme le nom de responsables, le nom de filières et le Gmail plus le numéro de téléphone

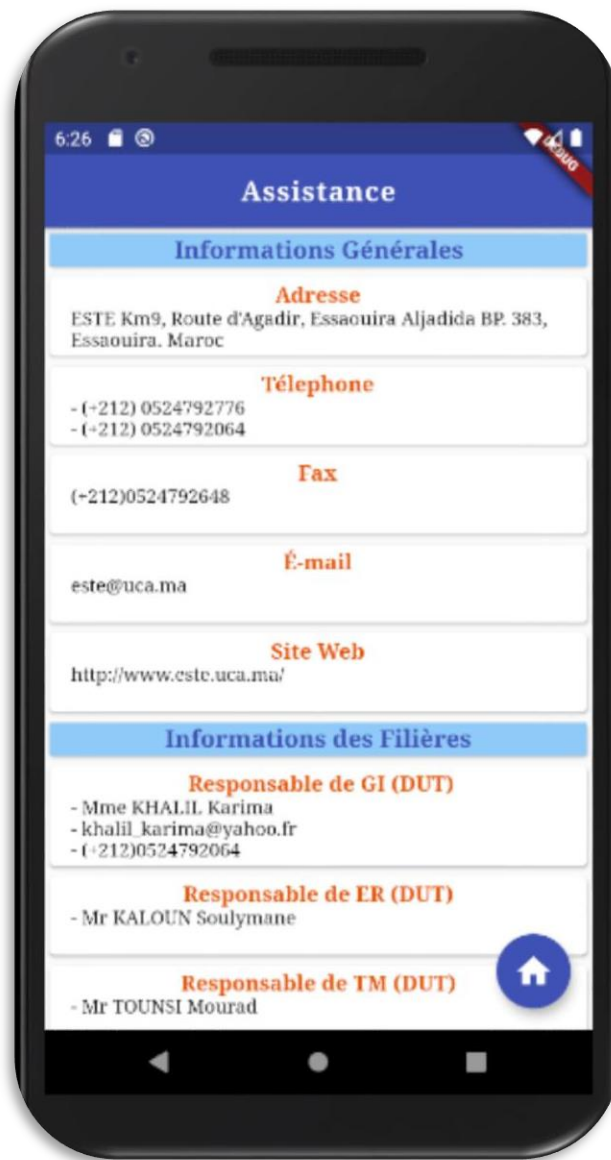


Figure 26 : la page assistance

➤ Page FAQ

On va voir une introduction sur l'application plus les questions fréquemment posées et leur réponse.

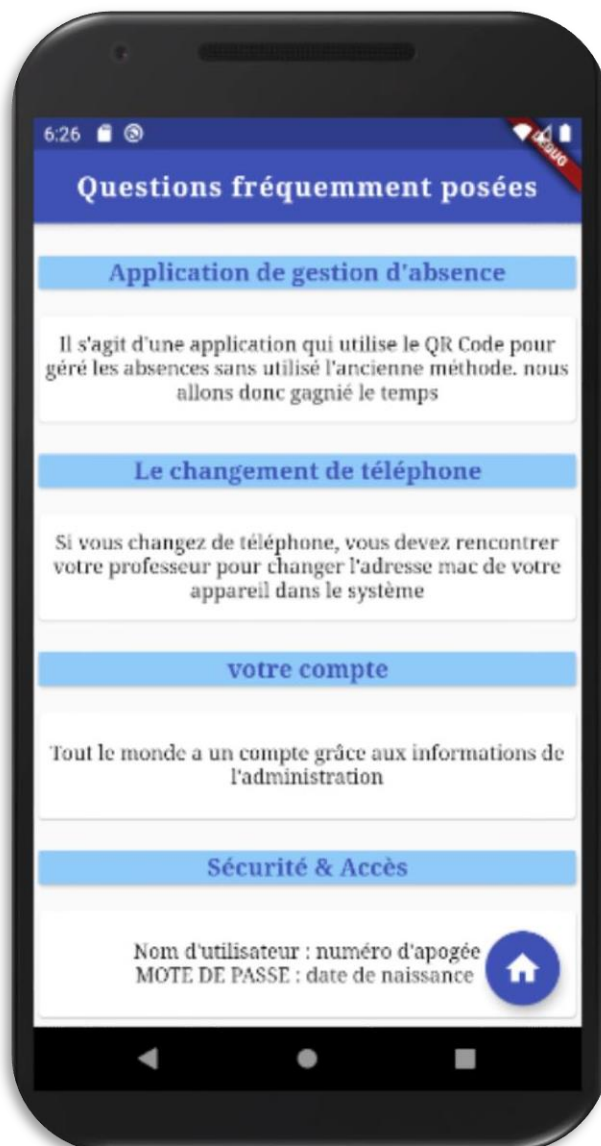


Figure 27 : la page FAQ.

➤ **Clique fermer**

Pour fermer le tiroir (drawer) de l'application.

➤ **Clique sur Se connecter**

On va accéder à la page d'accueil de notre application si les informations correctes sinon on va voir un message d'erreur. Grâce à la vérification fait par le fichier Node.js qui va vérifier si le nom d'utilisateur et le mot de passe existe dans la base de données. Si existe on va accès à la page d'accueil sinon on va voir ce message.

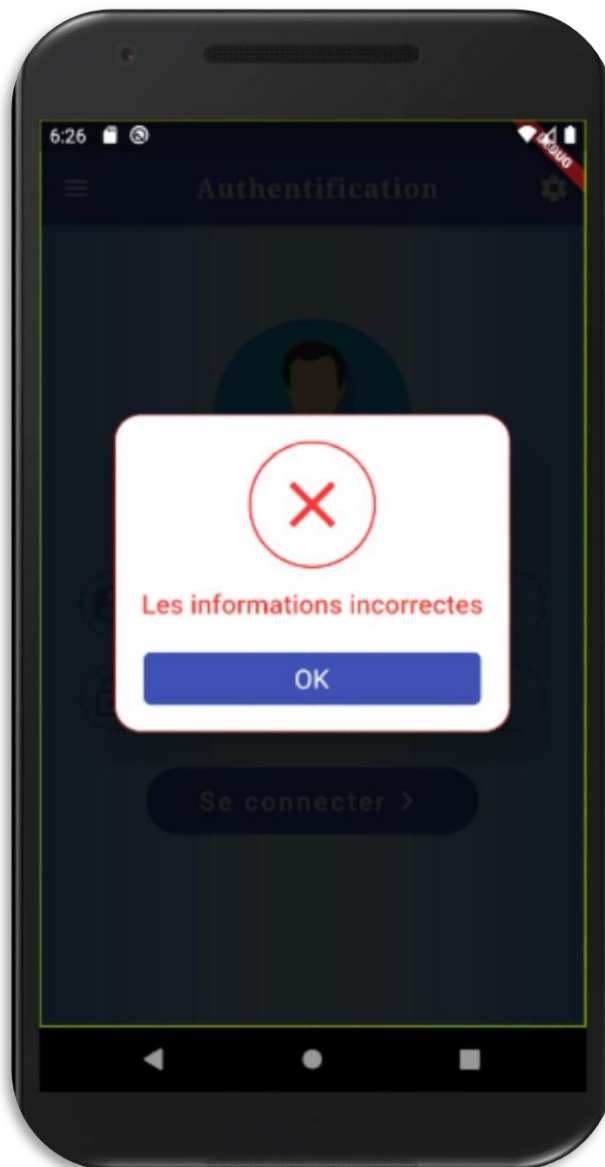


Figure 28 :Le message des informations incorrectes.

- **Si vous n'avez pas saisi toutes les informations tu vas voir le message ci-dessous :**

Dans l'application flutter on va vérifier si les deux champs pleins sinon on va avoir ce message.

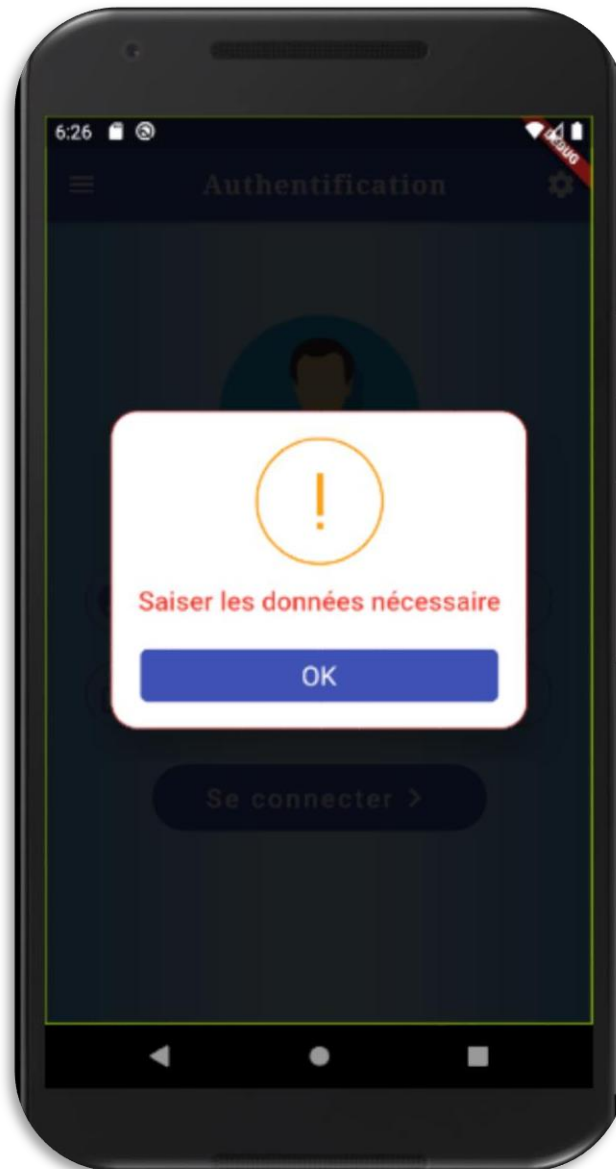


Figure 29 : le message de saisir les données nécessaire.

➤ Page d'accueil

Ici nous avons une liste du bouton pour accéder à l'accueil et pour accéder à la page de guidance et aussi à la page paramètre et au profil d'étudiant. Plus un buttons pour scanner le QR code afficher par le data show.

Et dans la barre supérieure nous avons un bouton pour la déconnexion.



Figure 30 : page d'accueil.

➤ Page de guidance

Dans cette page nous avons une vidéo qui explique comment utiliser cette application étape par étape.

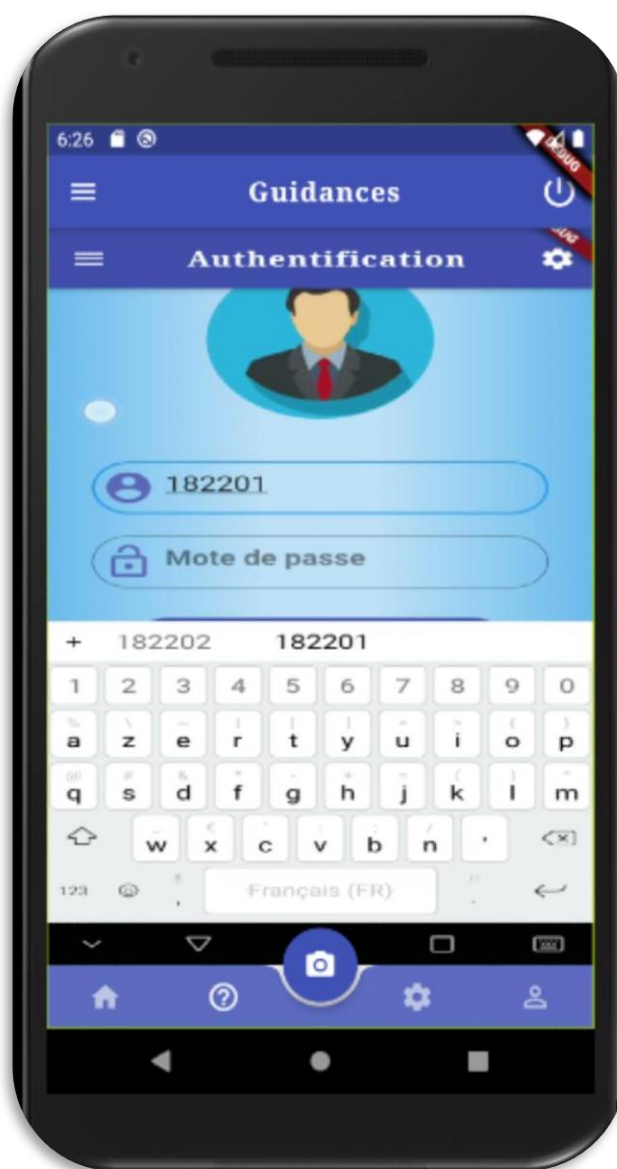


Figure 31 : la page guidance

➤ Page de paramétrer

Comme on a déjà vu dans cette page nous avons deux boutons un pour changer la langue et l'autre pour changer le mode.

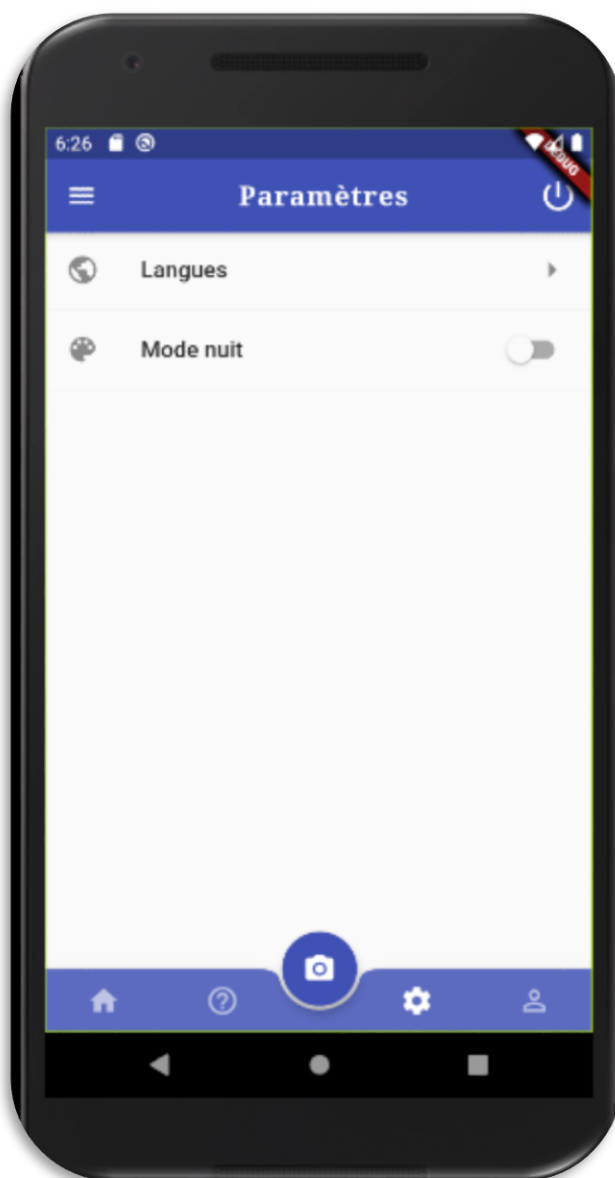


Figure 32 : la page paramétrer (après se connecter)

➤ Page de profil

Dans cette interface nous avons le nom complet d'étudiant, le numéro d'apogée, la date de naissance, le nom de filières, plus id de téléphone.

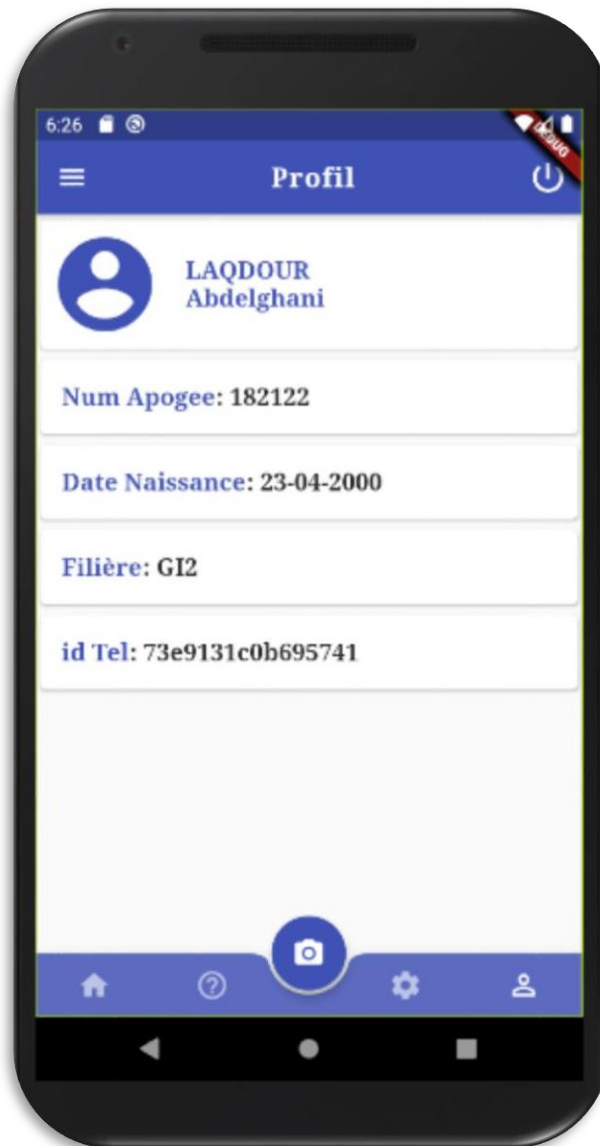


Figure 33 : la page profil.

➤ **Clique sur le bouton pour scanner le QR Code**

Si on clique sur le bouton pour scanner le QR Code nous verrons la caméra fonctionne automatiquement après votre autorisation.

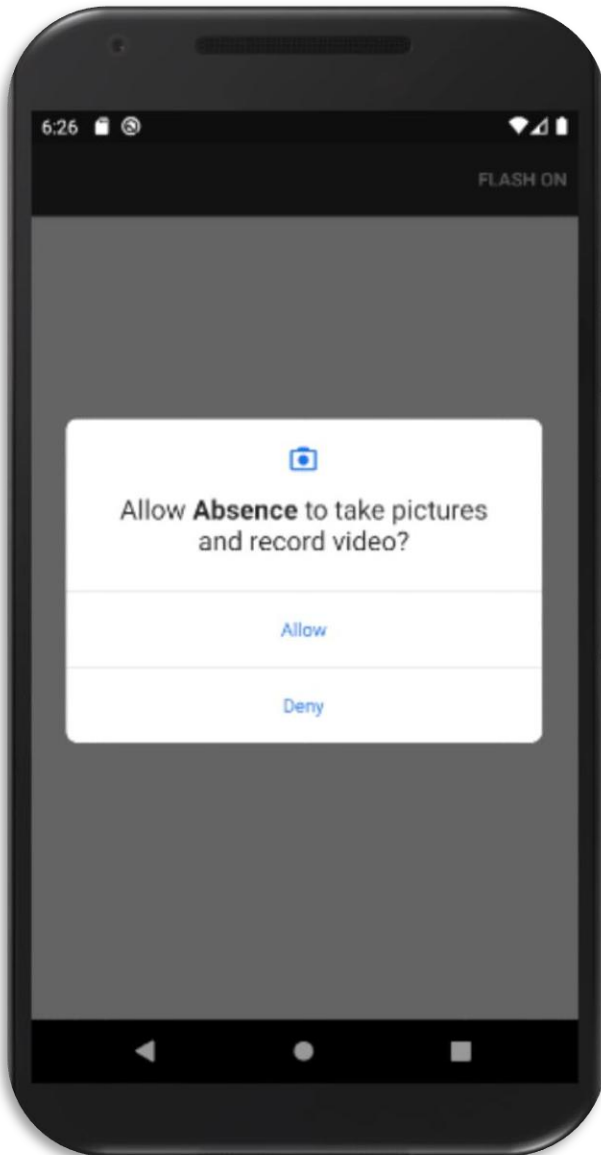


Figure 34 : demande d'autorisation pour utiliser la caméra



figure 35 : Scanner de QR Code

- Si vous avez cliqué sur 'Deny' c'est-à-dire refusé l'accès à la caméra tu vas voir ce message-là :

Grâce à des exceptions utilisées dans le code flutter.

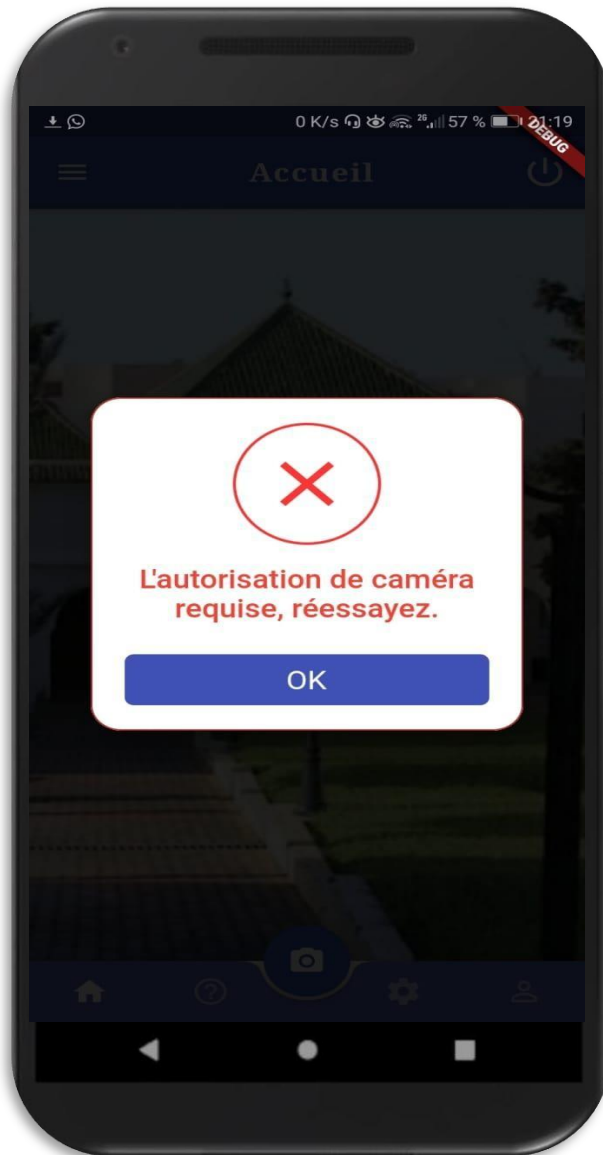


Figure 36 : l'autorisation de caméra refuse.

- **Si vous avez sortie avant de scanner de QR code tu vas voir ce message-là :**

Grâce à des exceptions utilisées dans le code flutter pour gérer ce cas-là.

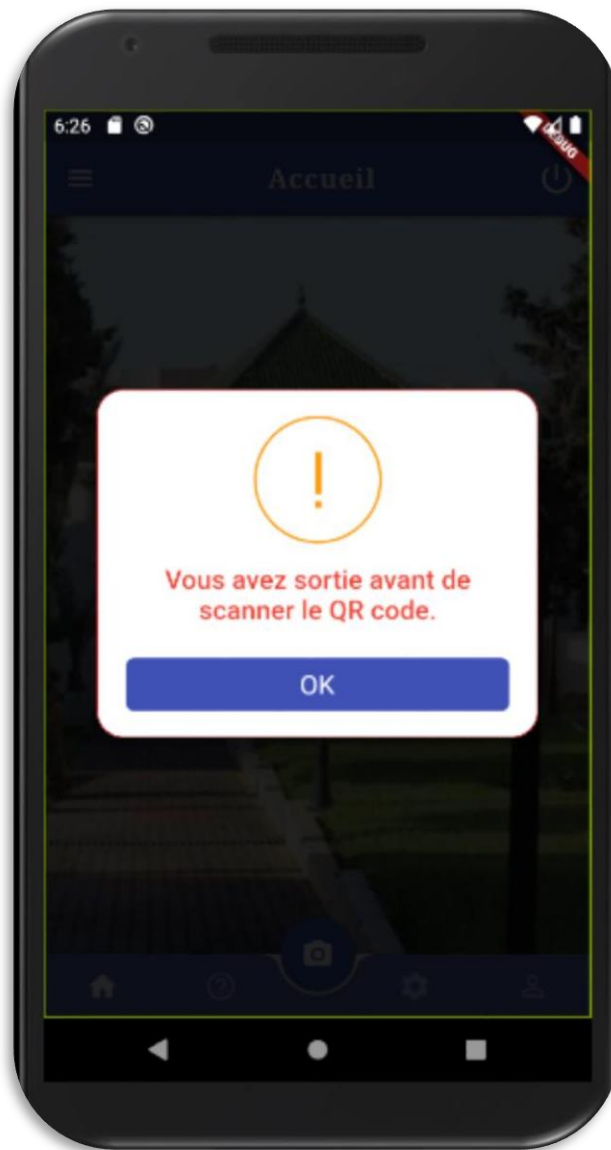


Figure 37 : message si vous avez sortie avant de scanner le QR code

- **Si vous avez scanné le QR code avec succès, tu vas voir ce message-là :**

Grâce à résultat retourner par le fichier Node.js qui signifie que ces informations scanner existe dans la base de données c'est- à-dire le fichier node.js si il a trouvé ces informations scanner il va changer l'état d'étudiant dans la table cours (absence au présent) et il va retourner une réponse (true). Dans on va voir ce message-là

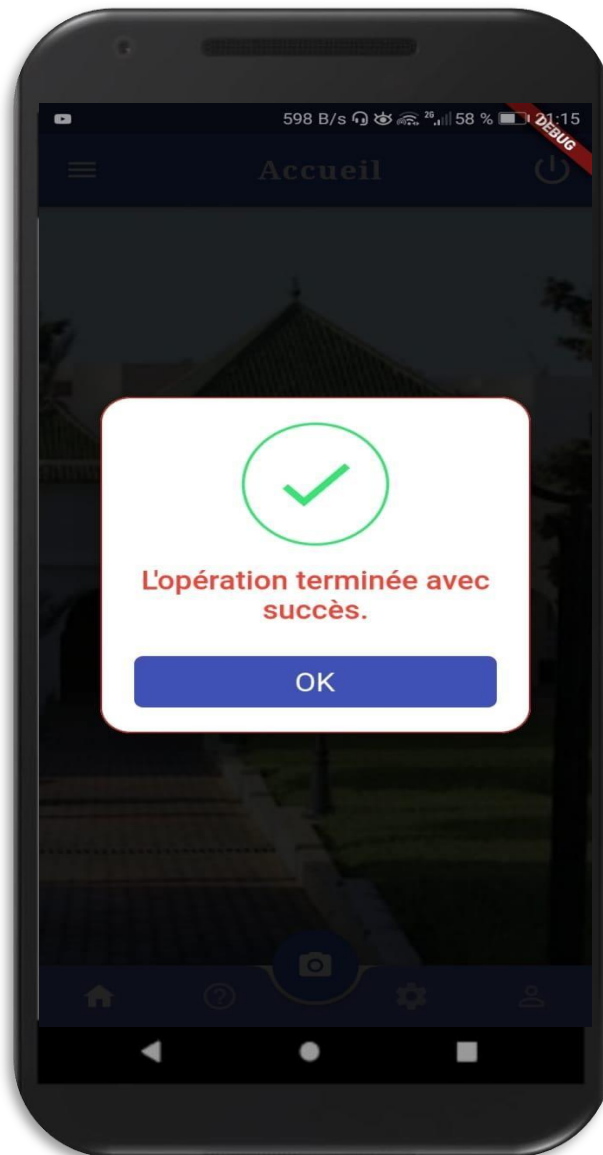


Figure 38 : message si vous avez scanné le QR avec succès.

- **Si vous avez scanné un QR code pas correcte tu vas avoir ce message-là :**

Grâce à la vérification fait par le fichier Node.js qui est dans le serveur qui va vérifier si les informations scannées (dans le QR code) existent dans la base de données ou non. Si non on va afficher ce message.

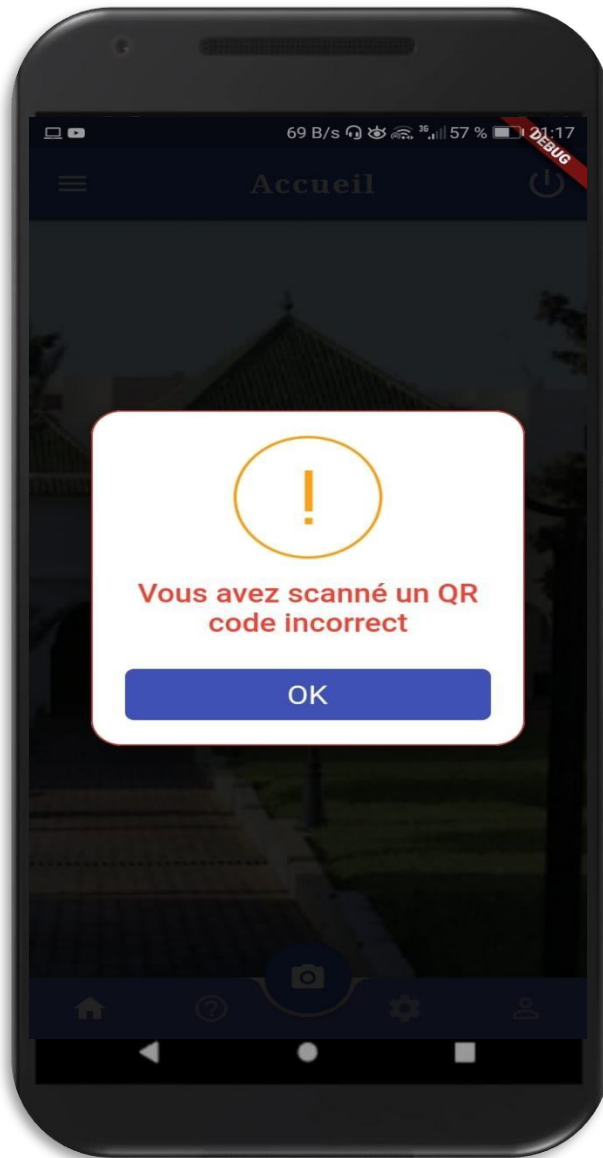


Figure 39 : message si vous avez scanné un QR Code pas correcte

- **Si vous avez scanné un QR code ancien ou d'une autre filière pas votre filière tu vas voir ce message-là :**

Ici les informations scanner on va le transféré vers le fichier Node.js qui est dans le serveur, pour voir si les informations existe dans la base de données ou non. si oui on va changer l'état d'étudiant (absence au présent) sinon on va afficher ce message-là .

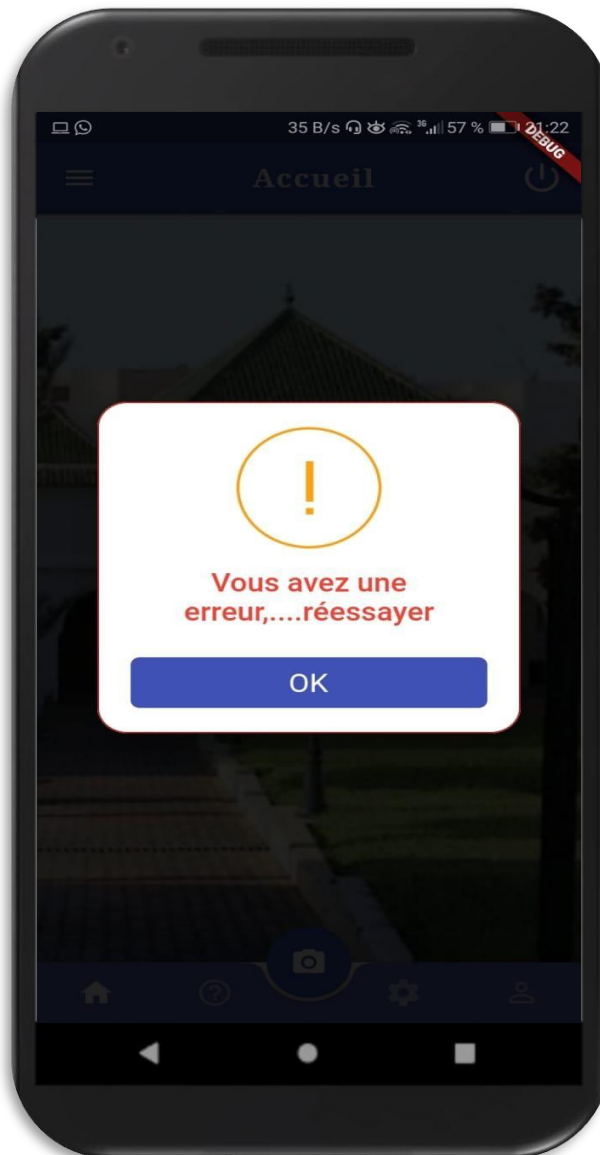


Figure 40 : le message si vous avez scanné un ancien QR code.

➤ **Clique sur le buttons de déconnexion**

Si on clique sur le bouton de déconnexion on va déconnecter et on va aller à l'interface Login.

Conclusion

Nous avons abordé le benchmarking après avoir présenté le contexte générale du projet, puis le cahier de charge, par la suite nous avons mis en place la partie conception selon les diagrammes du langage de conception UML, après on a présenté l'environnement et outils de travail, finalement on a présenté les étapes de réalisations et quelques fonctions utilisées dans le code, ainsi que les tests effectués sur l'application.

L'objectif de notre projet de fin d'étude est la réalisation d'une application mobile pour faciliter la gestion des absences et gagner le temps perdu avec la méthode manuelle,

Cette réalisation a été effectuée après trois étapes principales, la première c'est l'analyse et conception de projet, la deuxième étape c'est l'étude des outils proposés et leurs capacités pour qu'on puisse passer à la dernière phase qui est la réalisation.

Dans le cadre de notre projet, nous avons été permis de réaliser deux tâches avec deux Framework différents, **Flutter** pour l'application mobile et **Node.js** pour les interactions avec la base de données.

C'était un travail durant lequel nous avons pu développer et enrichir nos connaissances en informatique, et ce projet a été pour nous une occasion favorable car il nous permet de découvrir des nouvelles technologies et aussi comment travailler en groupe.

Tout au long de l'élaboration du projet, nous avons rencontré des difficultés tant au niveau conceptuel qu'au niveau de la réalisation. Tout de même, nous avons réussi à les surpasser pour présenter en fin de compte une application opérationnelle avec une seule limite est l'échange des téléphones entre les étudiants, mais nous pouvons dépasser cette limite en ajoutant la vérification avec la reconnaissance faciale ou l'empreintes digitales.

Webographie

https://pub.dev/packages/barcode_scan

<https://api.flutter.dev/flutter/widgets/widgets-library.html>

<https://stackoverflow.com>

<https://www.udemy.com/course/free-flutter-beginner-tutorial-build-own-app/>

https://www.w3schools.com/nodejs/nodejs_http.asp