

**GAUZI Romaric**

**PAILLARD Antoine**

**2A groupe 3**

---

# **Rapport du projet de développement mobile**



Année universitaire 2022-2023

# Sommaire:

## Introduction

### I) Fonctionnalités de l'application

- 1) Page matériel
- 2) Page location
- 3) Page client

### II) Choix de stockage des informations

- 1) Structure générale du projet
- 2) Organisation des données
- 3) Composant réutilisable créés
- 4) Lister les données
- 5) Formulaire de gestion des erreurs
- 6) Mode hors connexion

### III) Gestion de projet

- 1) Conception de l'application
- 2) Organisation du travail

## Conclusion

## Introduction

Ce rapport s'inscrit dans le cadre du projet final du module de développement mobile. L'objectif de ce projet était de créer une application mobile cliente du serveur ASP.NET Core MVC créé dans le cadre du projet de génie logiciel au semestre précédent. Ce rapport va illustrer et détailler le travail que nous avons réalisé tout au long de ce projet. Nous verrons dans un premier temps les fonctionnalités de notre application mobile, puis la manière dont nous avons choisi de stocker les informations utiles à l'application et enfin la façon dont nous nous sommes organisés tout au long de ce projet.

Petit rappel du contexte de notre application ASP.NET core MVC : son objectif était de pouvoir gérer la location de matériel à un ensemble de clients. L'application mobile que nous avons créée repose donc sur la même logique.

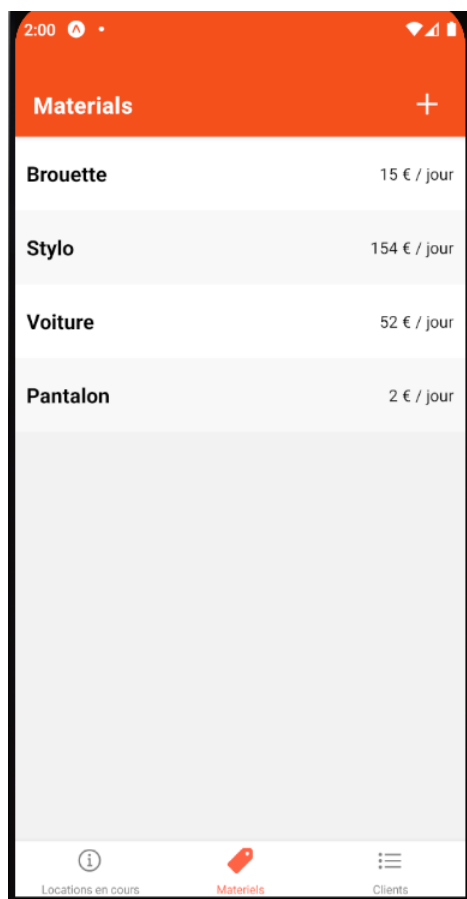
## I) Fonctionnalités de l'application

Notre application est constituée comme notre application web de 3 pages différentes. Il y a la page client, la page matériel et la page location. Nous allons par la suite détailler le fonctionnement de chacune d'entre elles.

### 1) Page matériel

Cette page permet de voir l'ensemble des matériels qui sont disponibles sur l'application. On peut voir le détail de chacun de ces matériels comme notamment l'ensemble de leur location ou encore le prix du matériel. On peut aussi ajouter, supprimer et modifier chacun de ces matériels.

Ci-dessous **la page principale** de la page Matériel qui permet de voir tout le matériel présent sur l'application



Ci-dessous **la page d'ajout d'un nouveau matériel** qui apparaît lorsque l'on clique sur le + en haut à droite de la page principal. On peut définir le nom de l'objet et le prix. Il suffit ensuite de cliquer sur enregistrer pour que ce matériel soit ajouté à ceux déjà existant.

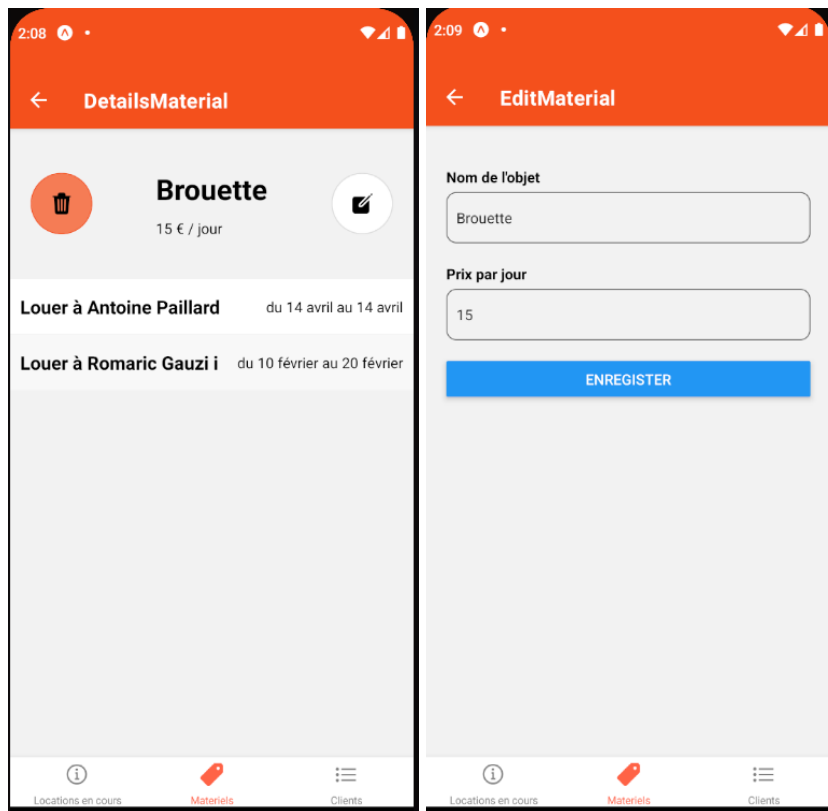
The image displays two side-by-side screenshots of a mobile application interface for creating a new material. Both screens have an orange header with a back arrow and the title 'CreateMaterial'. The status bar at the top shows the time (2:01 on the left, 2:03 on the right) and various icons.

The form consists of two input fields and a button:

- Nom de l'objet:** The first input field. In the left screenshot, it contains the placeholder text 'Nom de l'objet'. In the right screenshot, it contains the text 'jiojij'.
- Prix par jour:** The second input field. In the left screenshot, it contains the placeholder text 'Prix par jour'. In the right screenshot, it contains the text '12'.
- ENREGISTER:** A blue button located below the input fields.

At the bottom of each screen is a navigation bar with three items: 'Locations en cours' (with an information icon), 'Materiels' (with a red tag icon and highlighted in red), and 'Clients' (with a list icon).

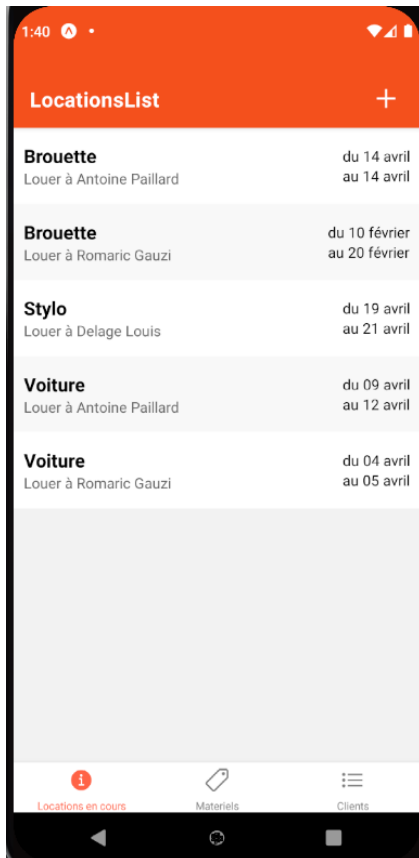
Ci dessous **la page de résumé d'un matériel**. On peut ensuite cliquer sur l'icône poubelle pour supprimer le matériel. On peut également le modifier en cliquant sur l'icône de modification.



## 2) Page location

Cette page permet de voir l'ensemble des locations qui ont été enregistrées. On peut effectuer toutes les modifications que l'on souhaite sur ces locations (ajout, modification, suppression). Les captures d'écran présentées dans la suite permettent d'illustrer ces actions.

Ci-dessous **la page principale** de la page Location qui permet de voir toutes les locations enregistrer.



Ci-dessous la **page d'ajout d'une nouvelle location** qui apparaît lorsque l'on clique sur le + en haut à droite de la page principal. On peut choisir la date de location que l'on souhaite, le locataire, le matériel à louer et ensuite enregistrer nos choix. La nouvelle location est ensuite enregistrée et apparaît sur la page principale.

The image displays three sequential screenshots of a mobile application interface for creating a new location, titled "CreateLocation".

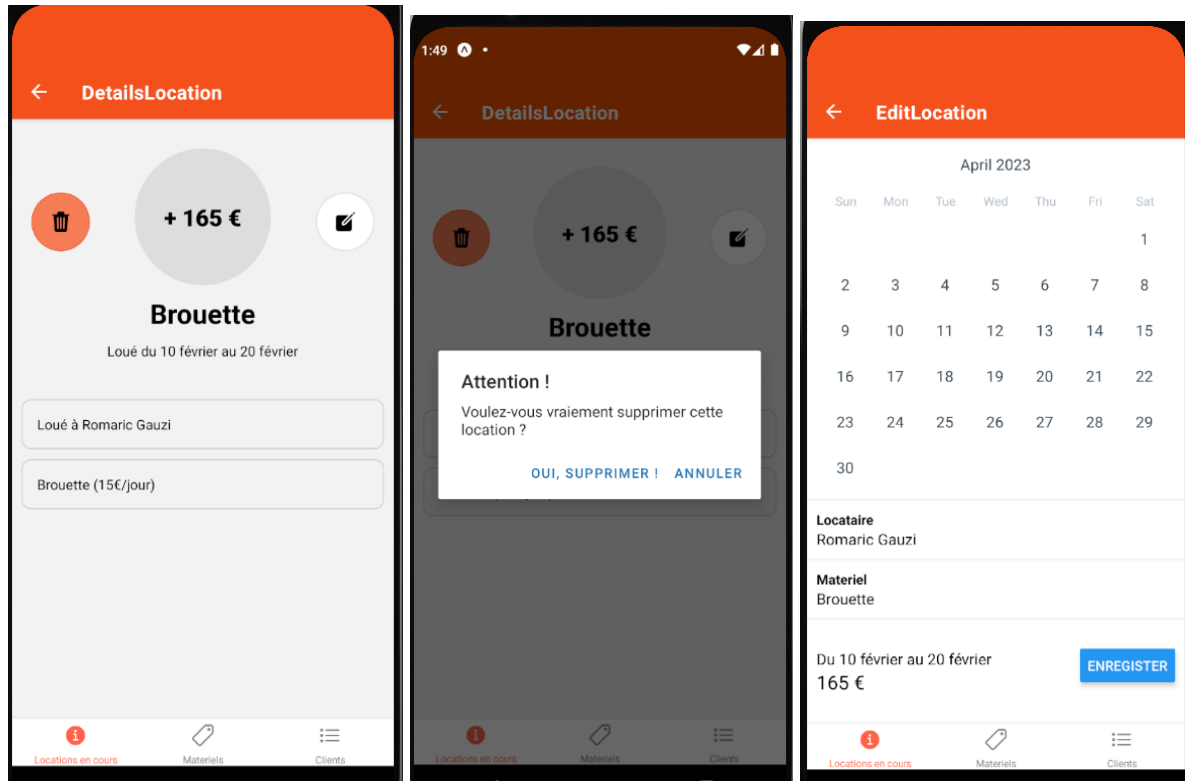
**Screenshot 1 (1:42):** Shows the initial state. At the top is a calendar for April 2023. Below the calendar, there are sections for "Locataire" (Antoine Paillard) and "Matériel" (Aucun matériel sélectionné). At the bottom, it says "Aucune date sélectionnée - €" and has an "ENREGISTRER" button. The bottom navigation bar includes "Locations en cours", "Matériels", and "Clients".

**Screenshot 2 (1:51):** Shows the calendar with dates 10, 11, 12, and 13 highlighted in orange. The "Locataire" section now shows "Antoine Paillard". The "Matériel" section shows "Voiture". The date range is "Du 10 avril au 13 avril" with a price of "208 €". The "ENREGISTRER" button is still present.

**Screenshot 3 (1:52):** Shows the "Matériel" selection screen. It has a search bar "Rechercher un 'Matériel'", a list of items (Brouette, Stylo, Voiture, Pantalon), and a close button (X) at the bottom.



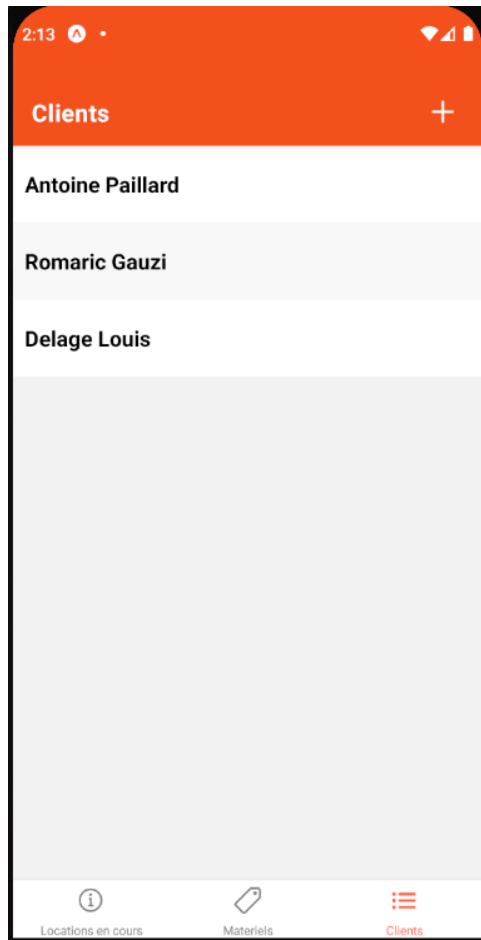
Ci-dessous **la page de résumé d'une location** qui apparaît lorsque l'on clique sur une location à partir de la page d'accueil. On peut ensuite décider de la supprimer avec l'icône poubelle ou de la modifier avec l'icône modifiée. La page de modification est similaire à celle qui apparaît lors de la création d'une nouvelle location.



### 3) Page client

Cette page permet de voir l'ensemble des clients qui sont inscrits sur l'application. On peut voir le détail de chacun d'entre eux avec notamment les locations qui leur sont associées. Enfin, on peut effectuer les actions classiques d'ajout, de modification et de suppression pour chacun d'entre eux.

Ci-dessous **la page principale** de la page client qui permet de voir tous les clients enregistrer sur l'application.



Ci-dessous **la page d'ajout d'un nouveau client** qui permet de renseigner les informations nécessaires à la création d'un client. Il suffit ensuite de cliquer sur enregistrer pour que ce nouveau client soit ajouté.

The image displays two side-by-side screenshots of a mobile application interface for creating a new client. Both screens have an orange header with a back arrow and the title "CreateClient".

**Left Screenshot (2:17):** Shows the form with empty input fields for the following fields:

- Nom:** Input field with placeholder text "Nom".
- Prenom:** Input field with placeholder text "Prenom".
- Telephone:** Input field with placeholder text "Telephone".
- Email:** Input field with placeholder text "Email".
- Adresse:** Input field with placeholder text "Adresse".

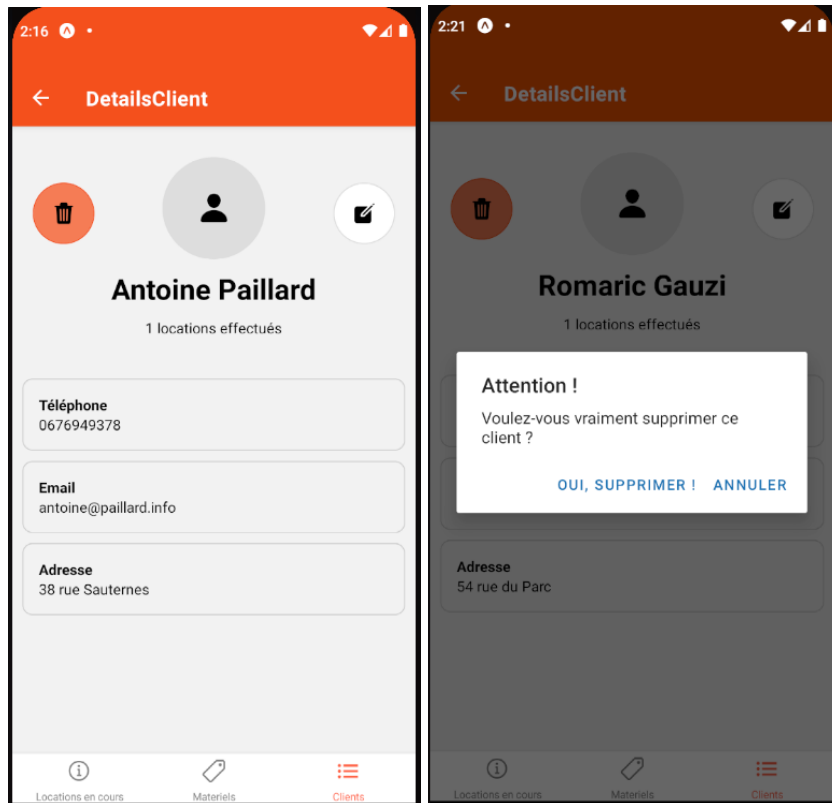
A blue button labeled "ENREGISTER" is positioned below the input fields. The bottom navigation bar includes three icons: a location pin for "Locations en cours", a tag for "Materiels", and a list icon for "Clients".

**Right Screenshot (2:18):** Shows the same form with sample data entered:

- Nom:** Input field containing "kopk".
- Prenom:** Input field containing "kopkop".
- Telephone:** Input field containing "0676949378".
- Email:** Input field containing "jjiojioj".
- Adresse:** Input field containing "jiojioj".

The blue "ENREGISTER" button and the bottom navigation bar are identical to the left screenshot.

Ci-dessous la page de résumé d'un client qui permet de voir ses informations personnelles. On peut ensuite décider de supprimer ce client en cliquant sur l'icône poubelle. Ou alors de le modifier en cliquant sur l'icône de modification.

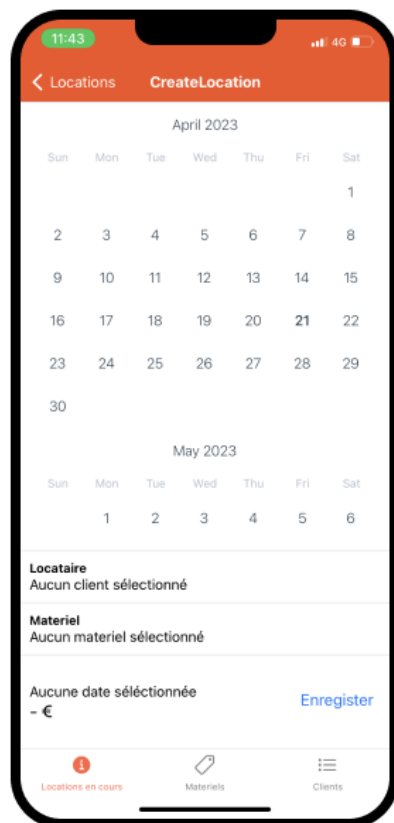


## II) Choix techniques

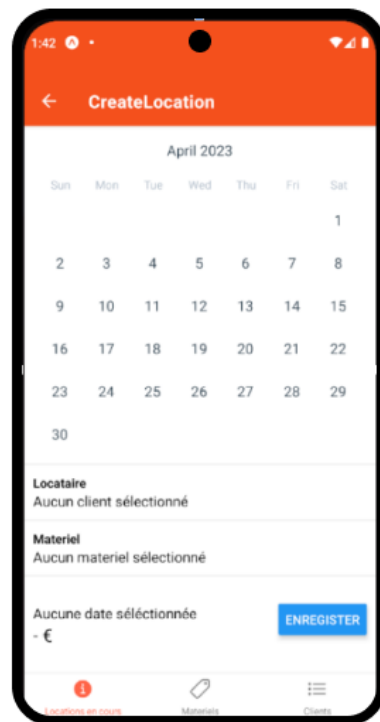
L'objectif de cette partie est de détailler les choix techniques effectués pour le développement de notre application.

### 1) Structure générale du projet

L'application utilise la structure générale proposée par expo avec la séparation des composants, des screens, de la navigation et des classes métiers liée aux accès externes. Pour le développement de l'interface, nous avons fait le choix de ne pas utiliser de kit de composant. Cela nous a permis de construire nos propres composants qui utilisent pleinement les composants natifs des téléphones et d'avoir par conséquent un rendu propre à chaque appareil.



**IOS**



**Android**

*Différence de visuel entre IOS et Android*

## 2) Organisation des données

Nous avons fait le choix de rester sur une structure très simple avec une pile réservée à chaque grand acteur de notre service (Location, Matériel et Client), ces 3 piles se trouvent dans le répertoire “/navigation”. De manière simple, chaque pile a un écran de listage, un autre de détail, un de création et un dernier de modification, ce qui fait que nous avons 12 écrans différents :

	... Location(s)	... Matériau(x)	... Client(s)
Liste des ...	LocationListScreen.js	MaterialListScreen.js	ClientScreen.js
Détails d'un(e) ...	LocationDetailsScreen.js	MaterialDetailsScreen.js	ClientDetailsScreen.js
Création d'un(e) ...	LocationCreateScreen.js	MaterialCreateScreen.js	ClientCreateScreen.js
Modification d'un(e)	LocationEditScreen.js	MaterialEditScreen.js	ClientEditScreen.js

Chaque écran a besoin de consommer les données de notre API pour fonctionner. C'est pourquoi 3 services API ont été créés afin de répondre aux besoins de gérer les Locations, les matériaux ou les clients. On retrouve dans ces services la capacité de récupérer la liste des éléments, un élément en particulier mais aussi les fonctions de création et de modification. Cela permet d'interagir avec l'api sans se poser de question, sachant qu'il arrive plusieurs fois d'utiliser une même route à 2 endroits de l'application. À cela s'ajoute la création pour chaque service d'un modèle représentant la données récupérées. Ce modèle permet notamment de faciliter la récupération des données, la création de nouvelles entrées ainsi que de se prémunir contre les erreurs de saisons dans la base de données :

Location	Matériel	Client
id: int; jourDebut : MM/DD/YYYY jourFin: MM/DD/YYYY materielId : int material : Material clientId : int; client : Client	id : int nom : string prixParJour : int masquer : boolean assignationPeriodes : []	id : int nom : string prixParJour : string masquer : boolean assignationPeriodes : []

### 3) Composant réutilisable créés

Tous les composants utilisés dans l'application ont été construits pour l'application. Voici une présentation de ceux développés, de leur usage et la justification de leur création :

**SelectSearch & SelectSearchModal** : Sélection d'un élément dans une liste avec recherche

Notre application a besoin de créer des joints entre les objets créés, pour cela, nous ne voulions pas simplement utiliser un composant select qui n'est pas très ergonomique lorsqu'il y a beaucoup d'options possibles. C'est notamment notre cas lorsque l'on souhaite sélectionner le locataire ou l'objet à louer. Ce composant est découpé en 2, une première partie est liée à l'affichage de la valeur sélectionnée, la deuxième concerne la modale qui s'ouvre lorsque l'on veut changer sa valeur. Ce composant intègre un système de gestion des erreurs

**SelecteurPeriode (Basé sur "react-native-calendars")** : Sélectionner une période.

Afin de choisir la date d'une location, un sélecteur de date est indispensable. Basé sur la librairie react-native-calendars, nous avons transformé ce composant en un sélecteur de période.

**FormInput** : Champ pour saisir des informations.

Ce composant permet de saisir d'informations en incluant un label et un système de retour des erreurs de saisie

**CircleButton** : Button rond avec une icon

**CircleButtonDanger** : CircleButton avec prévention contre les erreurs

**InformationBox** : Contenu stylisé pour les petites informations

#### 4) Lister les données

Nous avons plusieurs pages qui permettent de lister les données. Chacune de ces pages récupère la liste de l'élément via le service concerné. Les données récupérées sont ensuite listées dans une FlatList avec pour chacune un composant adapté à l'affichage de la donnée. Pour les locations, nous avons décidé d'afficher le nom du location, l'objet ainsi que la date de début et de fin tandis que pour le client, un simple nom et prénom suffit. De plus, chacune des flatlist intègre un refresh contrôle qui permet de recharger la liste des données.

<b>Brouette</b> Louer à Romaric Gauzi	du 10 février au 20 février	<b>Romaric Gauzi</b>
--	--------------------------------	----------------------

*Item affichant une location et item affichant un client*

#### 5) Formulaire et gestion des erreurs (Locations)

L'application offre la possibilité de modifier les données, pour cela, nous avons eu besoin de créer des pages qui permettent de renseigner des données ainsi que de les envoyer sur le serveur. La page la plus intéressante pour présenter cela est la création d'une location. En effet, contrairement à un formulaire texte comme pourrait l'être un Client ou un Material, celle-ci demande de saisir un IdClient et IdMaterial. C'est pourquoi nous avons fait le choix de sauvegarder ces données :

Variable d'état	Type	Rôle
periode	{ startDate: Date, endDate: Date }	Période sélectionnée, cette variable sera modifiée par le composant de sélection de Date
client	Client	Client sélectionné, nous avons fait le choix de sauvegarder directement le client plutôt que son ID seulement, notamment pour l'affichage du client sélectionné (Nom et prénom)
material	Material	Material sélectionné
periodError	String	Erreur rencontrée lors de la saisie du formulaire pour la période
clientError	String	Erreur rencontrée lors de la saisie du formulaire sur le client
materialError	String	Erreur rencontrée lors de la saisie du formulaire sur le matériel



Les données d'erreur sont mis à jour à chaque soumission et sont passé aux composants de formulaire comme "SelectSearch " ou "FormInput" qui gère les erreurs

The image displays two side-by-side screenshots of a mobile application interface, both showing error messages in red text.

**Left Screenshot (CreateLocation):**

- Header: 10:44, 4G, battery icon.
- Navigation: < Locations, CreateLocation.
- Calendar: Shows April 2023 and May 2023. The date 1 is selected in April.
- Locataire: Romaric Gauzi.
- Materiel: Vous devez selectionner le materiel à louer.
- Period: Vous devez selectionner une periode - €.
- Buttons: Enregister (blue), and a bottom bar with Locations en cours, Materiels, and Clients.

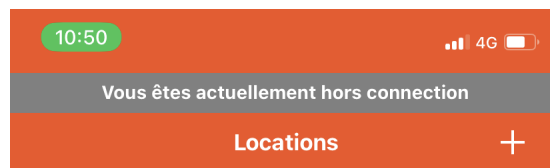
**Right Screenshot (CreateMaterial):**

- Header: 10:44, 4G, battery icon.
- Navigation: < Materials, CreateMaterial.
- Nom de l'objet: Ordinateur.
- Prix par jour: Prix par jour.
- Error message: Vous devez définir un prix.
- Buttons: Enregister (blue).
- Keyboard: Visible at the bottom.

Exemple de gestion des erreurs dans les formulaires

## 6) Mode hors connexion

Il est possible que le smartphone de l'utilisateur passe en mode hors connexion, c'est pourquoi nous avons développé l'application de sorte à ce que l'utilisateur soit informé sur le statut de connexion de son téléphone. Pour cela, nous avons, à la racine du projet créer un `InternetProvider` qui permettra d'accéder de manière globale au statut de connexion. Ce statut est géré à la racine du projet avec `NetInfo`. Ce mode hors connexion permet à l'application d'aller par exemple chercher les données de locations en cache. En effet, nous pensons que les données de locations peuvent par exemple être utilisées à l'utilisateur lorsqu'il n'a pas internet.



*Bandeau d'hors connexion*

### III) Gestion de projet

#### 1) Conception de l'application

Nous avons décidé de concevoir une application de manière à ce qu'elle soit complètement fonctionnelle et utilisable en l'état. Nous trouvions intéressant de nous donner comme objectif d'imaginer que cette application puisse être utilisable par une entreprise de location de matériel. On peut imaginer par exemple que la réception d'un hôtel puisse utiliser notre application. En effet, il pourrait y avoir des appareils à raclette ou des jeux de société à louer à leur client et notre application permettrait de les aider à mettre en place de manière simple cela.

Pour cette raison, nous avons décidé avant de créer cette application de réfléchir ensemble au besoin fondamental de ceux qui utiliseraient notre application. Nous avons détecté trois grands types de données dont il aurait besoin à savoir :

- Du matériel à louer
- Des clients
- Des locations pour lier client et matériel.

C'est cette réflexion qui nous a amenés à créer trois onglets dans notre application pour gérer ces trois composantes. Nous nous sommes demandés ensuite pour chacune d'entre elles quelles seraient les actions qui seraient nécessaires à l'utilisateur. Nous en avons relevé 4 :

- Ajouter
- Supprimer
- Modifier
- Voir le détail

Pour cette raison, nous avons fait en sorte que l'utilisateur puisse pour chacun de nos trois types de données réaliser ces 4 actions. Pour finir afin de rendre notre application la plus intuitive et simple à utiliser, nous avons fait en sorte de respecter une certaine logique identique en ces 3 onglets de manière à ce que des actions similaires soient réalisées de la même façon. Par exemple, pour supprimer une entité, il existe sur les 3 pages le même logo situé au même endroit qui effectue la même action.

#### 2) Organisation du travail

Notre niveau de compétence et d'expérience dans l'utilisation des technologies utilisées pour la création de l'application mobile était hétérogène au sein de notre équipe. Cela nous a poussés naturellement à nous assigner des rôles différents pour la réalisation de ce projet. Romaric est devenu le chef de projet et c'est lui qui nous a réparti les tâches et supervisé l'ensemble du code. De l'autre côté, avec un rôle plus exécutant, il y avait Antoine. Cette différence de niveau nous a été bénéfique pour tous les deux. D'un côté, le plus

expérimenté a pu partager ces connaissances et joué un rôle de direction qu'il ne connaissait pas totalement et de l'autre le moins expérimenté a pu profiter de l'expertise de son collègue. Cette organisation nous a permis d'avoir une bonne efficacité et de prendre du plaisir à participer à ce projet.

Nous avons pu avancer en parallèle en permanence avec une série de tâches à réaliser de courte durée de manière à pouvoir éviter les blocages et maximiser les échanges entre nous. Ces petites tâches étaient par exemple : utilisation de l'API Matériel, Utilisation de l'API Location, Création de la page principale...

Après avoir fini la partie code, nous sommes partagés la rédaction du rapport en deux grandes parties que nous avons rédigées séparément. Après cela, nous avons pu mutuellement lire et corrigé la partie écrite par l'autre.

## Conclusion

Les pistes d'amélioration que nous pourrions envisager pour notre application seraient par exemple :

- Modifier le style de l'application afin de lui créer une identité visuelle au travers d'un logo ou d'un code couleur différent de celui que nous avons choisi.
- Ajouter la possibilité de créer plusieurs fois le même matériel et les regrouper ensemble. On pourrait imaginer que l'hôtel possède plusieurs fois le même jeu et donc pour lui éviter de le créer deux fois sur l'application qu'il puisse le dupliquer.
- Améliorer le calendrier. En effet, bien que fonctionnel, le rendu visuel fourni ne nous satisfait pas complètement notamment parce que les périodes où l'appareil est déjà en prêt n'apparaissent pas clairement de manière lisible.

Pour finir ce rapport, nous voulions dire que nous avons tous les deux pris du plaisir de manière personnelle à réaliser ce projet qui clôture ce module de développement mobile.