

Membres : Yaye Touti Dia
Mohamed Gueye

Intitulé du projet : Agenda Électronique

Introduction

Il s'agit de réaliser un projet de fin de module de Programmation Mobile Android qui consiste à réaliser une application mobile pour répondre à un besoin spécifique.

Ce projet, avec un thème bien défini, emploie un cahier de charge avec des spécifications classiques qui sont à la base de toute application mobile.

I. Description

1. Thématique

Pour notre projet de module de Programmation Mobile, notre thème consiste à mettre en place un "Agenda Électronique". Ce sujet porte donc sur une application Android permettant de gérer les événements dans le temps pour une meilleure organisation de son quotidien.

2. Contexte

Dans nos vies bien remplies et ultra connectées, chaque jour, chaque heure, chaque minute est désormais comptée, décortiquée, comptabilisée, facturée...

Des questions comme " C'est quand le rendez-vous avec mon médecin", " A quelle date est mon entretien"... dessinent notre futur et constituent des éléments auxquels nous sommes confrontés tous les jours.

3. Problématique

S'organiser au quotidien reste un défi de taille. Mais par une multitude de petits gestes, on parvient à reprendre le contrôle. Tout n'est qu'une question de rangement et de planification.

Avoir son propre agenda permettant d'organiser nos journées devient une utilité dans la gestion de nos activités quotidiennes.

4. Finalité

Notre projet consiste à développer une application d'agenda permettant de planifier et gérer un ensemble de tâches quotidiennes, hebdomadaires et mensuelles et d'avoir un suivi instantané de ces dernières.

Par gestion on entend ajouter, modifier supprimer des événements selon une date, d'être notifié à temps réel de ces événements.

5. Public ciblé

L'application vise un grand public. Ce public fait référence à toute personne voulant gérer ces futurs événements depuis leur création jusqu'à l'heure ultime notamment dans le monde professionnel et éducatif (étudiants) où la gestion et l'organisation des activités constituent un enjeu majeur.

II. Étapes suivies et choix technique

1. Interface graphique

Dans la conception des interfaces graphiques, pour ne pas encombrer le futur utilisateur et pour offrir une meilleure expérience, nous avons limité le nombre d'activités en organisant plusieurs fonctionnalités sous forme de menus simples. Dès lors, par simple lecture, l'utilisateur pourra se repérer de manière intuitive.

- ✓ Une palette de couleur simple qui n'est pas trop exposée et identique sur toutes les vues a été mise en place.
- ✓ Dans chaque vue, le nombre de fonctionnalités est considérablement réduit pour éviter d'encombrer le contenu et des interfaces toutes simples pour des fonctionnalités mineures.
- ✓ Notre application s'adapte aussi bien à l'affichage en mode portrait qu'en mode paysage. Elle s'adapte aussi à la fois sur smartphone comme sur tablette. Les layouts changent donc en fonction du mode d'affichage et de l'appareil utilisé.
- ✓ Concernant les fonctionnalités implémentées, on a fait appel à quelques outils d'Android pour optimiser le plus possible nos programmes
 - Pour l'implémentation du calendrier, nous avons utilisé les **GridView** qui représentent les vues sous forme de grilles. Avec l'utilisation des **ArrayAdapter**, on assiste à une meilleure utilisation des composants de la grille pour une meilleure manipulation à la fois des vues et des données de la grille.
 - Pour adapter les vues avec les actions utilisateurs, on a aussi fait appel au **RecyclerView** afin de gérer facilement les actions utilisateur sur chaque élément de la grille.
 - Pour les boîtes de dialogues les **AlertDialogs** ont été utilisées pour
 - Ajouter, modifier et supprimer des événements (**AlertDialog**)
 - Définir et personnaliser les heures et la date (**TimePickerDialog**)

2. Internationalisation

Pour l'adaptation aux différentes langues, Nous avons opté pour deux langues de navigation dans notre application qui sont l'**anglais** et le **français**.

L'application s'adapte par rapport au format de la **langue locale** configuré sur l'appareil.

3. Mode de navigation

Pour pouvoir gérer son agenda, c'est-à-dire ajouter, modifier, supprimer ou même voir les détails relatifs à un événement, il est obligatoire de se connecter avec son login et mot de passe correcte. En plus, pour plus de sécurité, il l'est demandé à l'utilisateur de s'authentifier à nouveau pour pouvoir supprimer un événement afin de prendre toute précaution nécessaire pour toute opération irréversible

4. Création d'un compte utilisateur

Pour la création de compte utilisateur, vu que notre application est un agenda personnel, il n'est pas nécessaire d'avoir plusieurs comptes utilisateur.

De ce fait, le login et le mot de passe écrite à la première connexion sont enregistrés dans les préférences et c'est avec ces derniers qu'il pourrait avoir accès à son agenda aux prochaines connexions.

C'est avec ce compte que l'utilisateur pourra se connecter pour accéder aux fonctionnalités de l'application, recevoir des notifications et quelques informations personnelles.

5. Authentification pour la connexion à un compte

Pour cette partie, l'utilisateur doit renseigner son login et le mot de passe. Puis y'aura une vérification dans la référence où on a enregistré les données dès la première connexion.

Pour la première connexion, on vérifie si la référence est vide ce qui veut dire que l'utilisateur vient de télécharger l'application et que c'est ça première connexion. Puis on enregistre le login et le mot de passe de la première connexion comme données de connexion.

6. Persistance des données

Dans cette partie, on a opté pour deux classes pour gérer la base.

- Une Classe structuration des données

Cette classe est uniquement utilisée pour la structure de la base donnée et non pas au contenu de la base. Elle offre une bonne alternative si une éventuelle modification de la structure de la base se présente.

Cette classe gère le nom, les tables et les attributs de la table. Cela permettra de ne pas faire de confusion sur les noms d'attributs lors de la manipulation des données

- Classe DBOpenHelper

C'est dans cette classe où se fait le traitement des données de la base. Elle gère les différents accès, les types de requêtes et les fonctions pour manipuler les données et des opérations complexes.

Dans cette classe se trouve tous les fonctions pour avoir accès aux données pour créer, enregistrer, supprimer, modifier et pour récupérer des valeurs dans la base de données

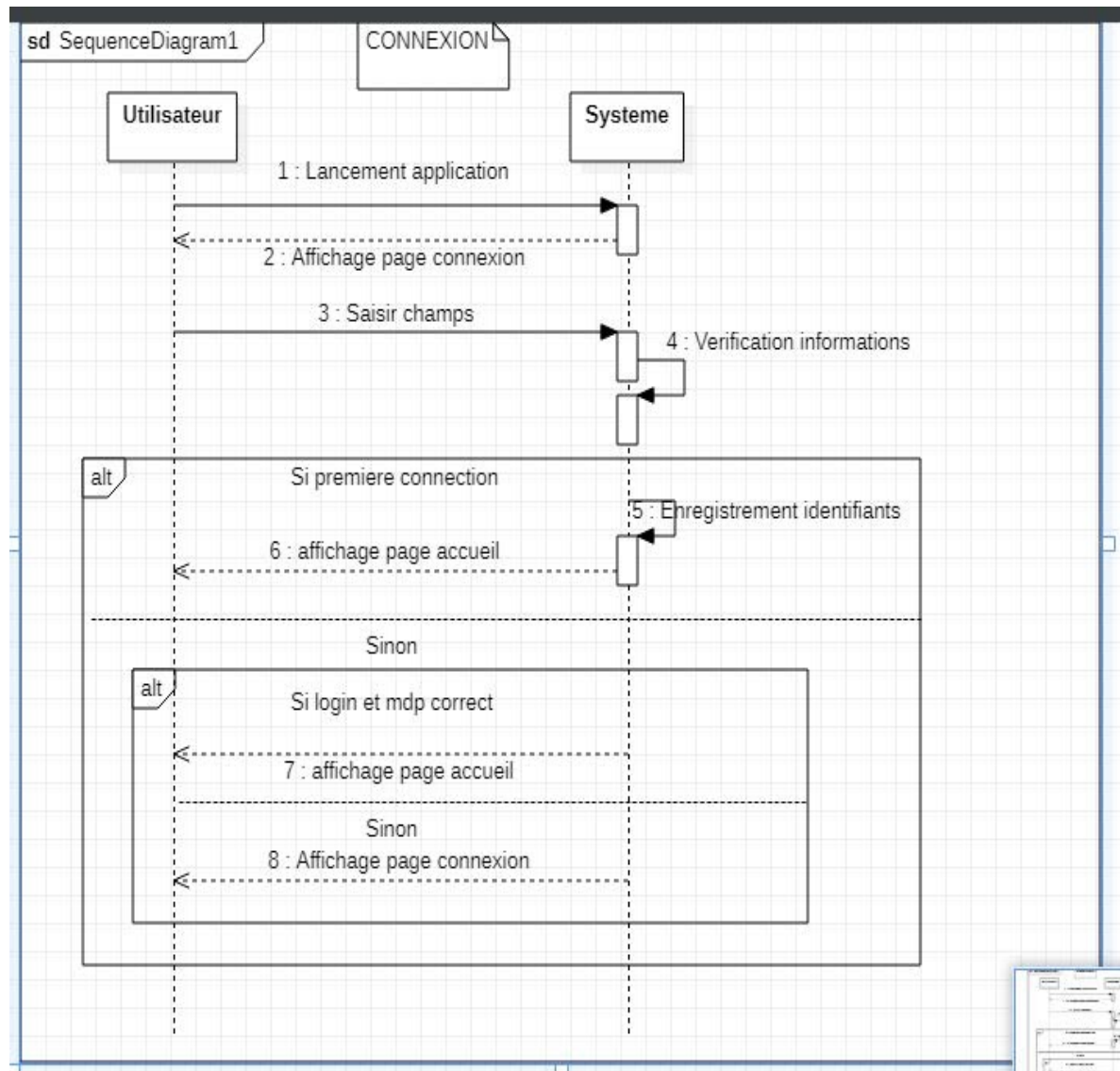
7. Capteurs et API

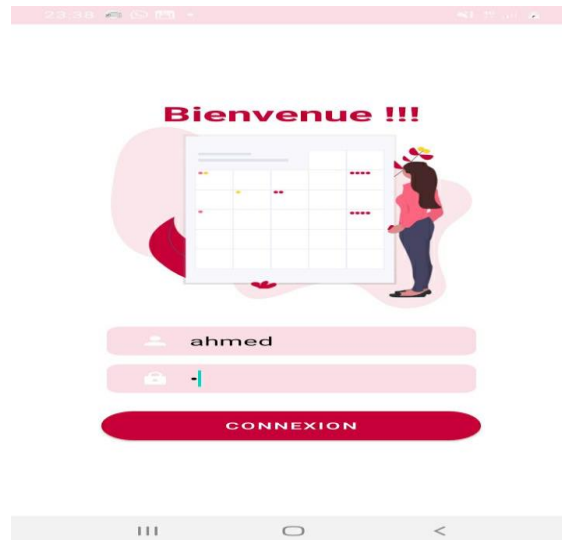
Pour permettre à l'utilisateur de saisir des endroits dans des événements et pour la géolocalisation, nous avons utilisé le capteur de localisation. Avec les permissions pour utiliser le capteur de GPS et l'API Google Maps avec une clé, un itinéraire est proposé à l'utilisateur utilisant la géolocalisation depuis sa position vers le lieu de l'événement.

III. Scénarios de navigation

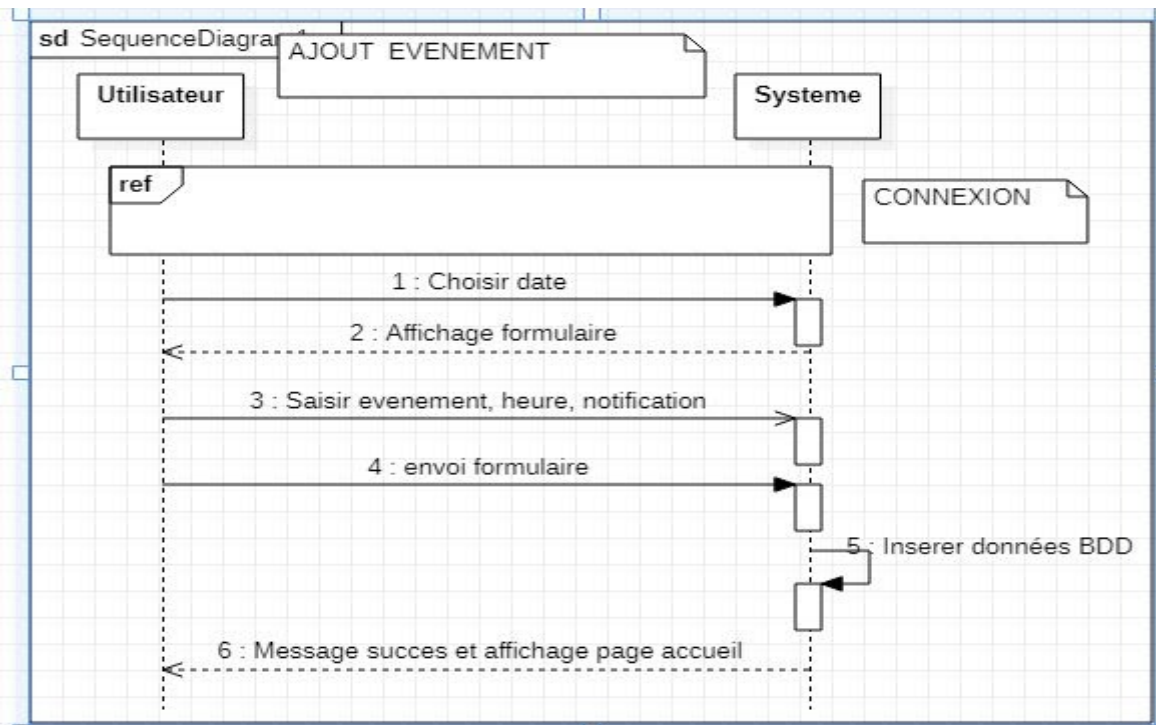
Ci-dessous les diagrammes de séquence des différents scénarios de l'utilisation de l'application

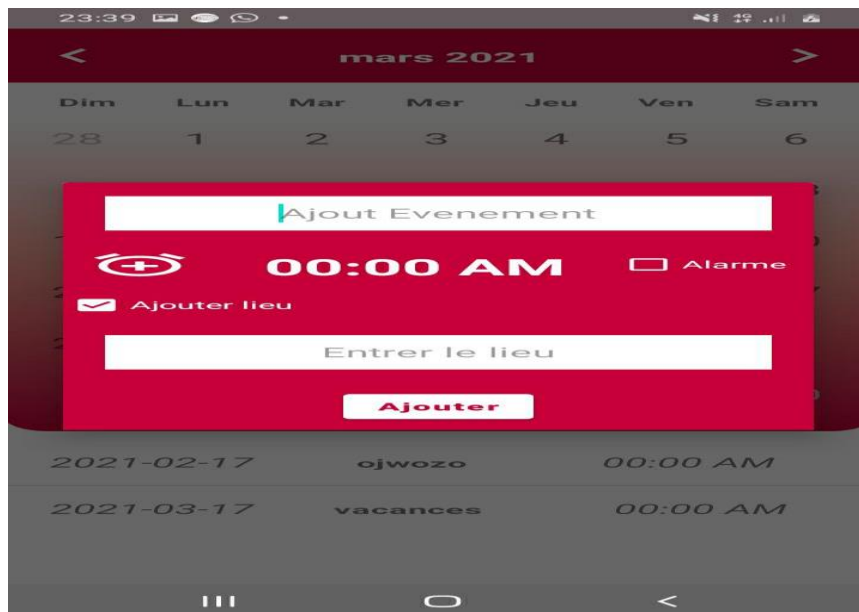
- **Création de compte et connexion à l'application**



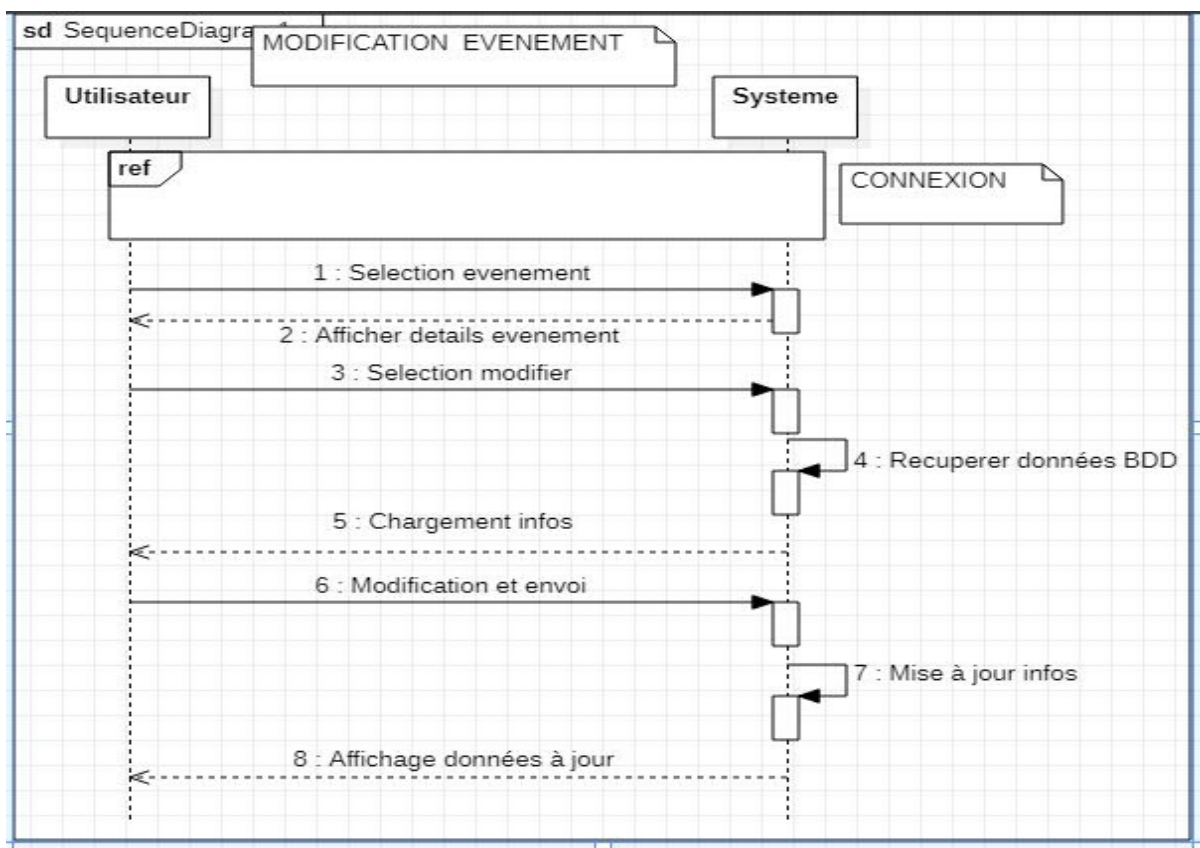


- Ajout d'un événement

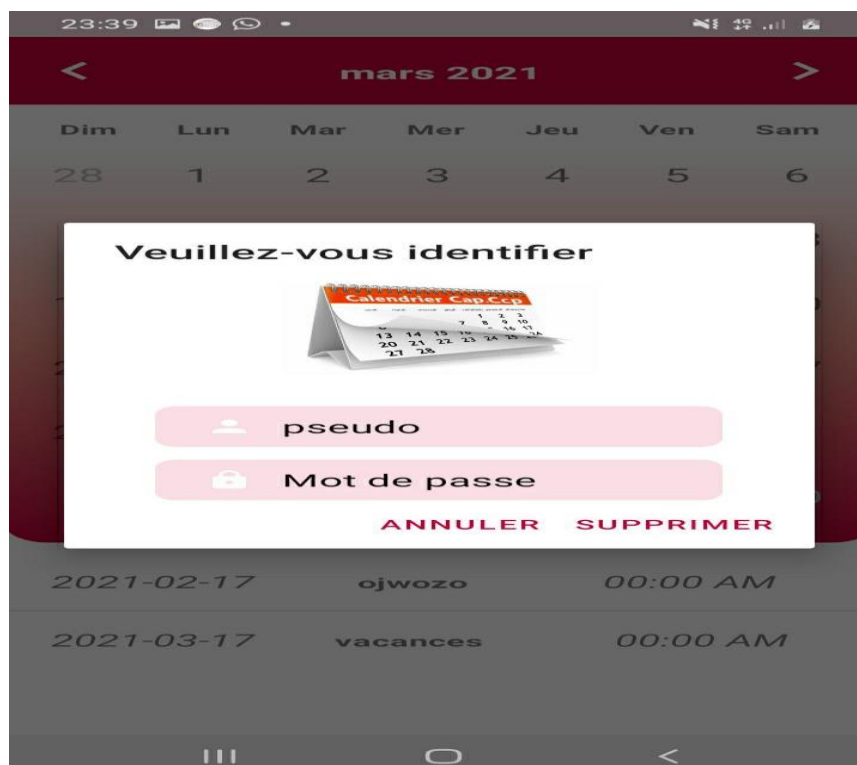
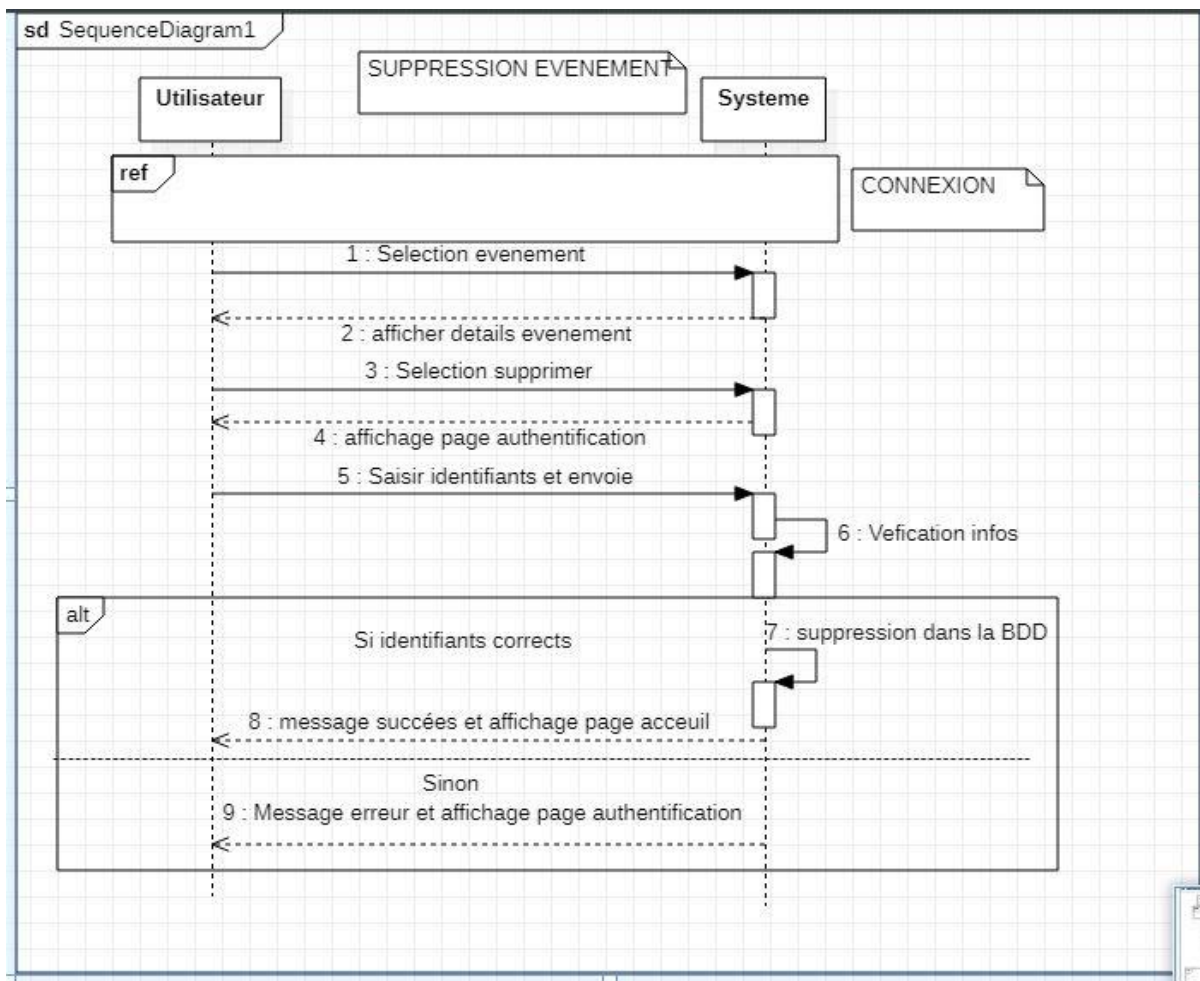




- **Modification événement**



- **Suppression événement**



IV. Difficultés et solutions

- ✚ Une des principales difficultés rencontrées lors du développement est l'implémentation de la fonctionnalité de notification et de l'alarme. Au début nous avons implémenté une librairie qui était obsolète et qui ne pouvait pas marcher avec notre version d'Android qui est l'api 30. C'est par la suite qu'on a fait les mises à jour nécessaires.
- ✚ Avec les listView qu'on avait utilisé pour gérer les éléments du calendrier. On a eu du mal à manipuler la grille. Cette démarche fut fastidieuse car les ListView ne sont pas très optimales avec les grilles. C'est par la suite qu'on a adopté les RecyclerView et les GridAdapter qui nous ont permis une meilleure manipulation de la grille.
- ✚ La géolocalisation nous a aussi causé quelques problèmes. On partait du principe de définir la position de l'utilisateur en récupérant sa latitude et longitude mais finalement l'API Google Maps nous a permis non seulement de déterminer la position exacte de l'utilisateur mais aussi de proposer un itinéraire.

V. Les tâches réalisées par chacun et le temps mis pour l'implémentation des tâches

Pour la répartition des tâches, elle a été définie comme suite :

- Yaye Touty DIA
 - Définition de la structure de la base de donnée
 - Implémentation de la base de données (
 - Gestion des préférences utilisateurs
 - Implémentation des différentes méthodes de sélection, mise à jour et suppression des données
 - Définition de la structure d'un événement du calendrier
 - Implémentation des classes pour gérer un événement
 - Création de la classe pour gérer les notifications
 - Intégration API Google et capteur
 - Test d'intégration
- Mohamed GUEYE
 - Gestion de tout ce qui est interface, layout et vue
 - Gestion des couleurs, drawables, thèmes etc.
 - Internationalisation de l'application
 - Gestion des différentes permissions et des problèmes de dépendances
 - Adaptation aux différents appareils et aux différents modes d'affichages.
 - Gestion de la sécurité pour la connectivité
 - Gestion des scénarios de navigation (événement utilisateur, activité, vue, etc.)

- Intégration API Google et capteur
- Test unitaire

Il faut noter que toutes ces tâches ont été effectuées en parallèle du processus de développement. A chaque étape, on revenait en arrière pour d'éventuelles modifications si nécessaire. Cela a permis le fait que chacun de nous comprenne le code dans son intégralité.

VI. Le planning initial pour la répartition des tâches

L'ensemble des tâches pour le planning initial a été réparti comme suit :

- ✚ Ensemble, nous avons réalisé les tâches suivantes :
 - ❖ Etude de la conception de l'application
 - ❖ Étude de la base de données avec les différentes tables à intégrer et leur dépendance.
 - ❖ Etude des différents services à développer
 - ❖ Découpage de l'application en différentes classes, activités ainsi que les layouts et fragments associés
 - ❖ Optimisation du code avec ajout éventuel de nouvelles fonctionnalités
 - ❖ Mise au point du processus de développement
- ✚ Mohamed GUEYE
 - ❖ Maquettage des différentes interfaces
 - ❖ Implémentation des activités et classe pour le back-end
 - ❖ Implémentation des layouts ainsi que le style du front-end
 - ❖ Internationalisation
 - ❖ Intégration de l'API Google Maps
- ✚ Yaye Touti DIA
 - ❖ Implémentation Base de
 - ❖ L'envoi de notification
 - ❖ Les capteurs de localisation
 - ❖ Intégration de l'API Google Calendar
 - ❖ Adaptation de l'application aux différents supports
 - ❖ Test d'intégration

VII. Les améliorations pertinentes pour l'application

Il serait pertinent d'ajouter des fonctionnalités telles que :

- Pouvoir se connecter avec ses emprunts digitaux
- De spécifier les jours fériés dans le calendrier