



Université Abdelmalek Essaadi
Faculté des Sciences et Techniques de Tanger
Cycle Ingénieur Logiciels et Systèmes Intelligents
Année Universitaire 2023-2024



Mini Projet 2: MAUI & SQLite “Gestion de Calendrier”



Réalisé par:
DAAOUAN Mohammed
FRIKH Said

Encadré par:
Pr. Hassan ZILI

Table de Matieres

I. Introduction.....	3
Présentation du projet.....	3
Objectifs du projet.....	3
Technologies utilisées (MAUI, SQLite, MVVM).....	3
II. Analyse des besoins.....	3
Identification des utilisateurs cibles.....	3
Description des fonctionnalités requises.....	3
Cas d'utilisation.....	4
Cas d'utilisation 1 : Ouverture de l'application.....	4
Cas d'utilisation 2 : Gestion des événements.....	4
Cas d'utilisation 3 : Notification de rappel.....	4
III. Conception du système.....	4
Architecture générale.....	4
Modèle de données (schéma de la base de données SQLite).....	5
Schéma de la base de données SQLite :.....	5
IV. Développement.....	5
Configuration de l'environnement de développement.....	5
Création du tableau de bord (Dashboard).....	5
Gestion des événements (CRUD).....	5
Vue calendrier.....	5
V. Démonstration.....	6
Écran Principal.....	6
Ajout d'un Nouvel Événement.....	7
Affichage des Détails d'un Événement.....	8
❖ Fonctionnalités Clés.....	8
VI. Conclusion.....	9
Résumé des réalisations.....	9
Prochaines étapes possibles.....	9
Apprentissage et recommandations.....	9
VII. Annexes.....	9
Code source.....	9
VIII. Références.....	9

I. Introduction

Présentation du projet

Le projet que nous avons entrepris visait à développer une application de gestion de calendrier. Cette application devait permettre aux utilisateurs de créer, modifier, afficher et supprimer des événements, et devait être fonctionnelle sur plusieurs plateformes, notamment Windows, macOS, Android et iOS. Nous avons choisi d'utiliser .NET MAUI pour tirer parti de ses capacités multiplateformes.

Objectifs du projet

- Explorer les bases de .NET MAUI pour le développement d'applications multiplateformes.
- Utiliser SQLite pour la gestion des données locales de manière efficace.
- Créer une interface utilisateur intuitive et réactive pour une expérience utilisateur optimale.

Technologies utilisées (MAUI, SQLite, MVVM)

Pour atteindre ces objectifs, nous avons utilisé les technologies suivantes :

- **.NET MAUI** : pour le développement multiplateforme.
- **SQLite** : pour la gestion de la base de données locale.
- **MVVM** : pour une séparation claire entre la logique métier et la présentation.

II. Analyse des besoins

Identification des utilisateurs cibles

Nous avons identifié deux groupes principaux d'utilisateurs pour notre application :

- Les utilisateurs individuels qui cherchent à organiser leurs événements personnels de manière efficace.
- Les petites entreprises et équipes qui souhaitent coordonner leurs événements et réunions pour améliorer la productivité.

Description des fonctionnalités requises

Pour répondre aux besoins de ces utilisateurs, nous avons défini les fonctionnalités suivantes :

- **Tableau de bord (Dashboard)** : offrant une vue complète des événements du mois, de la semaine et du jour avec des indications de date/heure et de statut (à venir, terminé).

- **Gestion des événements (CRUD)** : permettant de créer, lire, mettre à jour et supprimer des événements.
- **Vue calendrier** : affichant le calendrier avec les événements marqués et permettant de cliquer sur un jour pour voir les détails des événements de ce jour.
- **Notifications de rappel** : envoyant des notifications locales pour rappeler les événements à venir.

Cas d'utilisation

R.G: Utilisateur : peut s'authentifier, voir ses événements, gérer ses événements et recevoir des rappels.

Cas d'utilisation 1 : Ouverture de l'application

- L'utilisateur ouvre l'application.
- L'application affiche directement le tableau de bord avec le calendrier mensuel.
- L'utilisateur peut immédiatement commencer à naviguer dans le calendrier et ajouter des événements.

Cas d'utilisation 2 : Gestion des événements

- L'utilisateur crée un nouvel événement en saisissant les détails (titre, date, heure, lieu).
- L'utilisateur modifie un événement existant.
- L'utilisateur consulte les détails d'un événement.
- L'utilisateur supprime un événement.

Cas d'utilisation 3 : Notification de rappel

- L'utilisateur définit un rappel pour un événement.
- L'application envoie une notification locale à l'heure définie.

III. Conception du système

Architecture générale

L'application suit le modèle MVVM (Model-View-ViewModel) pour séparer les préoccupations :

- **Model** : Représente les données de l'application.
- **View** : Représente l'interface utilisateur.
- **ViewModel** : Gère la logique de l'application et l'interaction entre le modèle et la vue.

Modèle de données (schéma de la base de données SQLite)

Schéma de la base de données SQLite :

- **Event** : (EventID, Title, Description, Date)

IV. Développement

Configuration de l'environnement de développement

Pour commencer le développement, nous avons configuré notre environnement de développement en installant .NET MAUI et en configurant Visual Studio. Nous avons également intégré SQLite pour la gestion des données locales.

Création du tableau de bord (Dashboard)

Le tableau de bord de l'application offre une vue complète des événements :

- **Vue mensuelle** : Affiche tous les événements du mois.
- **Vue hebdomadaire** : Affiche tous les événements de la semaine.
- **Vue quotidienne** : Affiche les événements du jour.
- Indication de date/heure et statut (à venir, terminé).

Gestion des événements (CRUD)

La gestion des événements comprend :

- **Création** : un formulaire permet d'ajouter un nouvel événement avec validation des champs.
- **Lecture** : les détails des événements peuvent être consultés.
- **Mise à jour** : un formulaire permet de modifier un événement existant.
- **Suppression** : les utilisateurs peuvent supprimer un événement après confirmation.

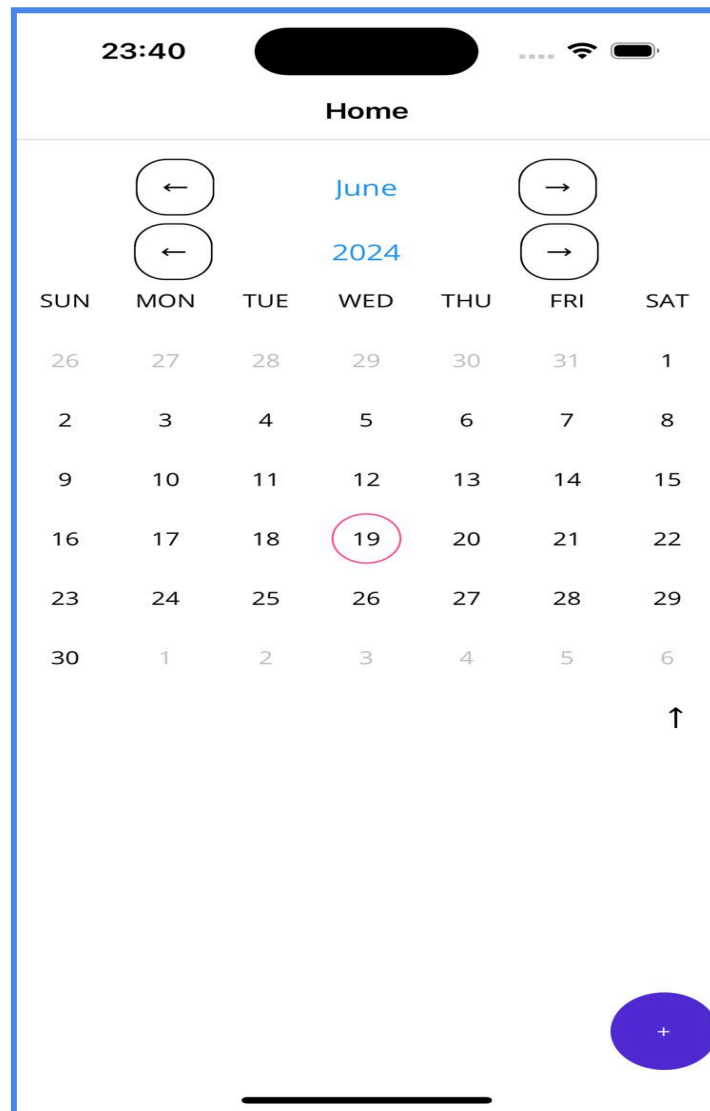
Vue calendrier

- Nous avons utilisé des contrôles de calendrier interactifs pour afficher les événements.
- Les utilisateurs peuvent cliquer sur un jour pour voir les détails des événements de ce jour.

V. Démonstration

Écran Principal

L'écran principal affiche le calendrier du mois en cours. Les utilisateurs peuvent naviguer entre les mois et les années grâce aux boutons de navigation situés de part et d'autre des indicateurs de mois et d'année. Chaque jour du mois est représenté, et les événements du jour sélectionné sont mis en évidence.



Ajout d'un Nouvel Événement

Lorsque l'utilisateur clique sur le bouton "+" en bas à droite de l'écran principal, un formulaire s'affiche permettant d'ajouter un nouvel événement. Ce formulaire comprend des champs pour le titre de l'événement, la description, et la date. Après avoir renseigné ces informations, l'utilisateur peut enregistrer l'événement en cliquant sur le bouton "Save".

The screenshot shows a mobile application interface. At the top, the status bar displays the time 23:41, a black pill-shaped notch, and icons for cellular signal, Wi-Fi, and battery. Below the status bar, the word "Home" is centered. The main content area shows a calendar for June 2024. Navigation arrows for months and years are at the top of the calendar grid. The days of the week (SUN to SAT) are listed above the dates. A modal form titled "Add Event" is overlaid on the calendar. It contains three input fields: "Nouveau événement" (Title), "Description de l'événement" (Description), and "6/19/2024" (Date). A blue "Save" button is at the bottom of the modal. A blue circular button with a white "+" icon is located at the bottom right of the screen, below the calendar. At the very bottom, a white bar contains the word "Done" in blue text.

SUN	MON	TUE	WED	THU	FRI	SAT
26	27	28	29	30	31	1
2	Add Event					8
9	Nouveau événement					15
16	Description de l'événement					22
23	6/19/2024					29
30	Save					6

Affichage des Détails d'un Événement

Lorsqu'un jour contenant des événements est sélectionné, les détails des événements du jour s'affichent en bas du calendrier. Chaque événement est répertorié avec son titre et sa description, offrant une vue d'ensemble claire et organisée des engagements de l'utilisateur pour la journée.



❖ Fonctionnalités Clés

- **Navigation Simplifiée** : La navigation entre les différents mois et années est facilitée par des boutons intuitifs.
- **Ajout Rapide d'Événements** : Les utilisateurs peuvent facilement ajouter de nouveaux événements via un formulaire accessible en un seul clic.
- **Vue Détail des Événements** : Chaque événement ajouté est clairement affiché, permettant aux utilisateurs de visualiser rapidement leurs engagements.

VI. Conclusion

Résumé des réalisations

Nous avons réussi à développer une application de gestion de calendrier fonctionnelle et multiplateforme. Nous avons intégré MAUI et SQLite de manière efficace et implémenté toutes les fonctionnalités principales, y compris les notifications de rappel.

Prochaines étapes possibles

Pour améliorer encore l'application, nous pourrions ajouter des fonctionnalités supplémentaires telles que la synchronisation avec des calendriers en ligne, améliorer l'interface utilisateur et optimiser les performances.

Apprentissage et recommandations

Ce projet nous a permis de mieux comprendre le développement multiplateforme avec .NET MAUI et la gestion de bases de données locales avec SQLite.

VII. Annexes

Code source

<https://github.com/Daaouan/.Net-MAUI-SQLite-Application-Calendar-Management-.git>

VIII. Références

[1] learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0

[2] www.sqlite.org/docs.html

[3] www.syncfusion.com/blogs/post/agenda-view-in-net-maui-scheduler-a-perfect-tool-for-modern-day-office-management