République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Licence en Informatique Générale

Thème

Intitulé de votre mémoire

Réalisé par

M. NOM Prénom
M. NOM Prénom
M. NOM Prénom
M. NOM Prénom
M. NOM Prénom
M. NOM Prénom

Devant le jury composé de

Président :	M. NOM Prénom	Grade de l'enseignant	Université de Béjaïa
Examinateur :	M. NOM Prénom	Grade de l'enseignant	Université de Béjaïa
Examinatrice :	M. NOM Prénom	Grade de l'enseignante	Université de Béjaïa
Encadrant :	M. NOM Prénom	Grade de l'enseignant	Université de Béjaïa

Promotion 2017 - 2018

Remerciements

Texte ...

Texte ...

Texte ...

Texte ...

Texte ...

Dédicaces

Texte ...

$\mathcal{M}$. NOM Prénom

Texte ...

$\mathcal{M}^{lle}$ NOM Prénom

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

SE	Système d'Exploitation
UML	Unified Modeling Language
UP	Unified Process

Introduction générale

Texte ...

Chapitre 1

Contexte et problématique

1.1 Introduction

Pour réaliser une application de qualité il faut un plan, une méthodologie. La première étape de cette méthodologie est l'analyse qui va nous permettre de répertorier les fonctionnalités principales de l'application, tandis que l'étape de conception va nous permettre de modéliser les solutions suite à l'analyse en ayant recours au langage de modélisation choisi. Le langage UML(Unified Modeling Language) sera présenté au cours de ce chapitre, puis nous parlerons des applications mobiles et du contexte du projet actuel puis on finira avec un cahier des charge.

1.2 UML

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel. Il est destiné à décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie également les notations nécessaires aux différentes activités d'un processus de développement d'applications et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression des besoins jusqu'à l'étape de réalisation.[1]

E n fait, et comme son nom l'indique, UML n'a pas l'ambition d'être exactement une méthode : c'est un langage. UML est donc non seulement un outil intéressant, mais une norme qui s'impose en technologie à objets et à la quelle se sont rangés tous les grands acteurs du domaine, acteurs qui ont d'ailleurs contribué à son élaboration. [2]

1.3 Application mobile

Une application mobile n'est rien d'autre qu'un logiciel téléchargeable que l'on installe facilement sur nos smartphones(mobile intelligent) comme on ferait sur notre ordinateur. Le téléchargement de l'application mobile se fait suivant deux options :

— Sur téléphone par le biais de connexion Internet.
— Sur ordinateur en le branchant avec le téléphone mobile.

Il existe trois types distincts selon leurs spécificités techniques qui sont :

Application Native : Ces applications sont liés au système d'exploitation sur le quel sont installées car elles utilisent des caractéristiques reliées à celui-ci. Elle sont écrites dans un langage adapté au système d'exploitation en question. Ce type d'application est accessible seulement sur les systèmes d'exploitation auxquelles sont destinées. Ces plate-formes retirent 25% du prix de vente pour une application native payante.

Application Web : Ce sont toutes les applications conçues grâce aux outils de développent web actuels (HTML, CSS, JavaScript..). Elles sont accessible sur tout les mobiles via un navigateur Web ce qui la rend plus intéressante sur le point de vue financier car les coûts de développement sont réduits vue qu'on développe une seule application qui est compatible avec tous les smartphones quelque soit leur système.

Application Hybride : sont des applications qui incorporent les deux principes de développement des types précédemment cités. Les caractéristiques des applications web et celles des application native. Elles pourront être distribuées sur les plate-formes de téléchargement telles que l'Apple Store (iOS), Play Store (Android) ou encore Windows Store (Windows Phone). L'utilisateur peut donc installer ces applications et consulter leur contenu sans avoir à passer par un navigateur web. Ce type d'application mobile minimise les charges et la durée de son développement même si cela sera au détriment du perfectionnement et de la qualité qui caractérise l'application native.

La figure suivante illustre les trois types cités.

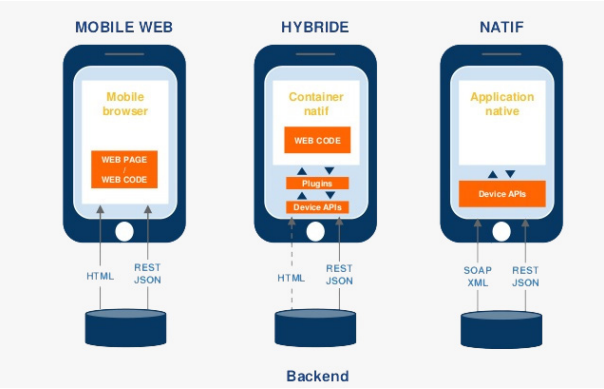


FIGURE 1.1 – Les différents types d'applications.

1.4 Contexte du projet

On dit toujours que le temps est la chose la plus précieuse car on peut tout acheter sauf le temps. Quoi qu'on fasse, une journée durera toujours 24 heures et le rythme de vie de la société moderne nous fait sentir que l'on est en perpétuellement en manque de temps. Donc la solution serait qu'on organise mieux notre temps . En utilisant une des plus grandes inventions modernes accessibles à tous qui est le smartphone ceci en mettant entre vos mains un outil aussi simple qu'efficace, capable de vous accompagnez tout au long de votre journée pour vous rappeler ce que vous avez prévu à quelle heure et bien plus encore.

Voila pourquoi on a décider de réaliser ce projet. On a décidé de faire une application dont le mot d'ordre est simplicité. Tout au long de ce travail, nous allons raisonné avec le principe du rasoir d'Ockham également appelé principe de simplicité. Vous allez voir une conception simple, des diagrammes très légers pour aboutir a un résultat fidèle à ce principe que nous avons pas décelé dans les applications testées.

Vous avez tendance à arriver en retard ou carrément à rater des rendez-vous car vous aviez prévu autre chose au même moment sans le savoir ? Vous faites pleins d'activités et vous n'arrivez plus à vous situer ? Vous ne savez pas quoi répondre quand on vous demande si vous êtes libre à tels moment ? Désormais, vous n'allez plus plus à réfléchir pour répondre à cette question grâce a notre application.

1.5 Cahier des charges

l'application à développer aura pour mission d'offrir une représentation des événements et des activités de l'utilisateur pendant les jours de la semaine. Pour cela, l'application devra répondre à ces besoins avec les fonctionnalités suivantes :

- Permettre à l'utilisateur d'organiser ses activités et les regrouper dans de différents calendriers.
- Permettre à l'utilisateur d'ajouter des activités dans le calendrier qui leur convient.
- Offrir une interface intuitive à l'utilisateur pour afficher ses activités.
- Générer des alertes/notifications pour les activités correspondantes.

1.6 Conclusion

Texte ...

Chapitre 2

Spécification des besoins et conception

2.1 Introduction

2.2 Spécification et analyse des besoins

Une phase décisive du processus de développement d'une application. Elle permet d'identifier les acteurs ainsi que formaliser les besoins fonctionnels et non fonctionnels, déduire les différents cas d'utilisation à partir des besoins fonctionnels et les détaillés. On conclut cette section avec des maquettes IHM et leurs liens de navigation.puis on passeras a l'analyse des besoins pour formuler le modèle du domaine et le diagrammes des classes participantes.

2.2.1 Identification des besoins

Besoins fonctionnels Permettre a l'utilisateur d'ajouter des événements et de les administrer dans les calendriers qui leur correspondent qu'il aura préalablement pris le soin de créer selon ses besoins. Il devra être rappelé et informé des événements pour lesquels il a préalablement défini les alertes, même lorsque l'application n'est pas active. Avoir une représentation détaillées sur son emploi du temps général (tous les calendriers)voire qu'un seul. Adapter la représentation pour afficher une seule journée ,3 jours, une semaine. Permettre à l'utilisateur de modifier à tout moment les informations déjà saisies.

Besoins non fonctionnels L'application doit remplir des critères non fonctionnels comme :

- La fiabilité : Si application crache et que l'utilisateur n'a pas reçu l'alerte, Cela peut s'avérer problématique.
- L'utilisabilité : L'utilisateur doit pouvoir maîtriser le fonctionnement de l'application facilement et rapidement.
- Performance : On est amené à vérifier notre emploi du temps plusieurs fois par jour, l'application doit être accessible rapidement pour ne pas contrarier l'utilisateur.

2.2.2 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes utilisés pour donner une vision globale du comportement fonctionnel d'une application. Ils permettent de recueillir, d'analyser et d'organiser les besoins. Il s'agit donc de la première étape UML d'analyse d'un système. Les éléments de diagrammes de cas d'utilisation sont :

- Acteur : Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom inscrit dessous.
- Cas d'utilisation : est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service. Il se représente par une ellipse contenant le nom du cas.
- Relations d'association : Une relation d'association est un chemin de communication entre un acteur et un cas d'utilisation et est représenté par un trait continue [3].

Dans ce qui suit, nous décrivons le diagramme de cas d'utilisation associé à l'utilisateur de l'application à réaliser. L'application est supposé fournir une représentation simplifiée de l'emploi du temps d'une personne (ex :étudiant). Suite à ça, nous avons identifié un seul acteur qui est l'utilisateur lui-même.

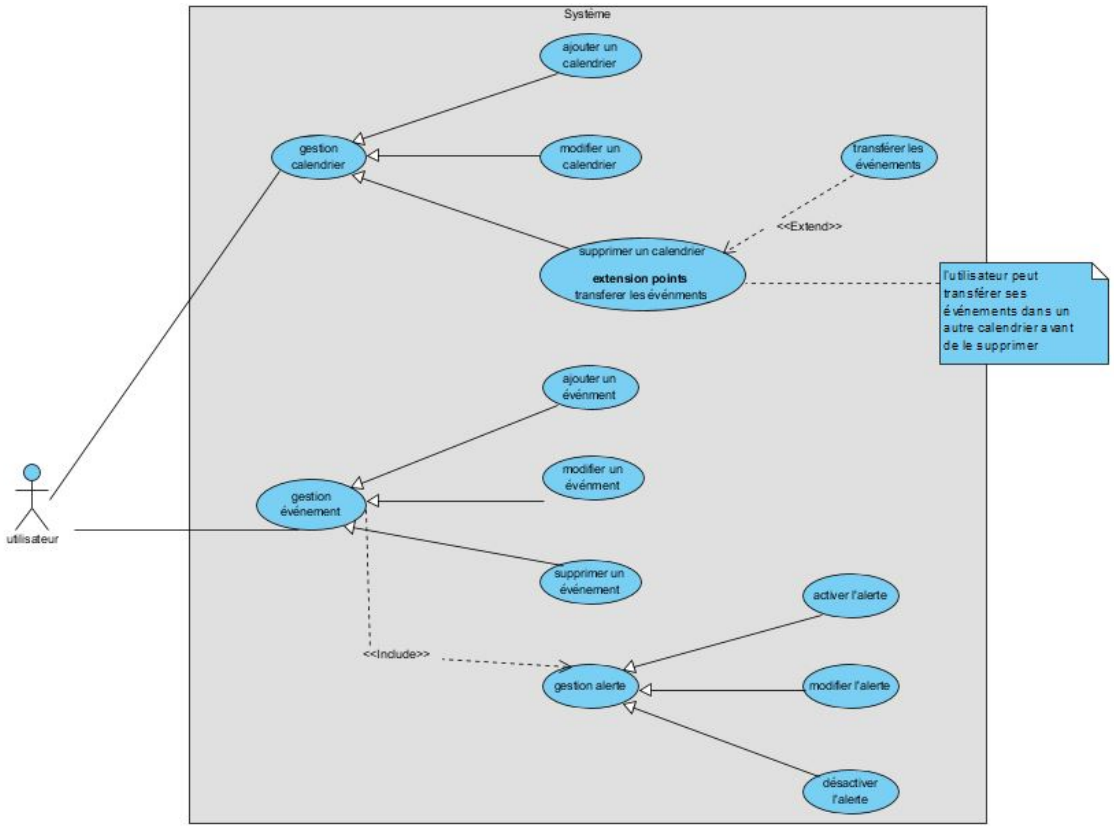


FIGURE 2.1 – Diagramme de cas d'utilisation.

Le diagramme de la figure 2.1 représente les différents cas d'utilisation associés à l'utilisateur. Ce dernier peut librement gérer ses calendriers et ses événements. La suppression d'un calendrier peut être étendue par « transférer les événements ». La gestion d'un événement inclut automatiquement «la gestion de l'alerte».

2.2.3 Maquettes IHM et navigation

Une maquette est un produit jetable permettant aux utilisateurs d'avoir une vue concrète mais non définitive des interfaces de la future application. On parle aussi de prototypage d'IHM (Interface Homme Machine). Une maquette permet de bien cerner les fonctionnalités de l'application à réaliser.[4]

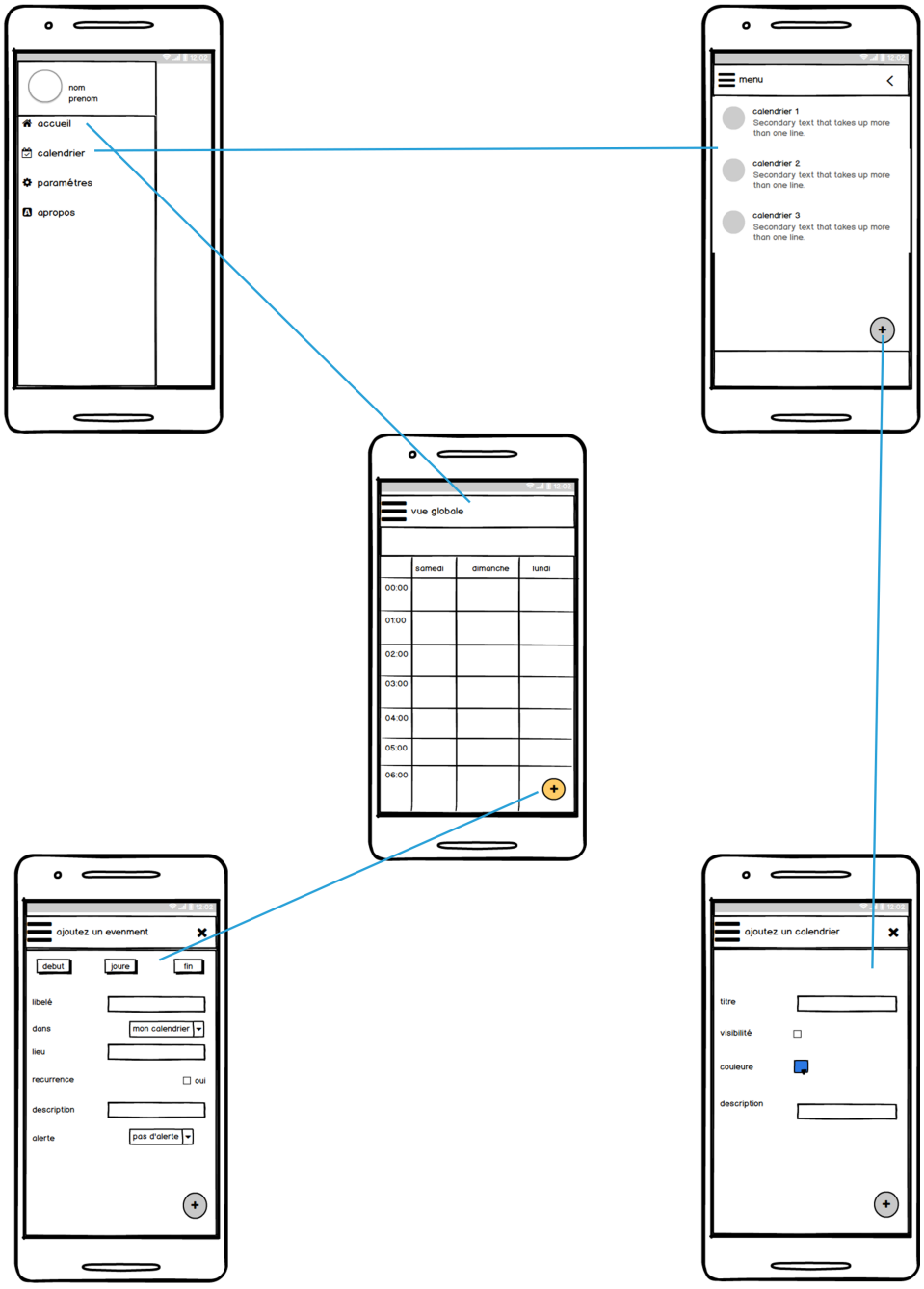


FIGURE 2.2 – Maquette IHM et navigation

Dans cette figure2.2, on présente les interfaces principales et les liens qui les relient, vous remarquerez que nous avons essayé de rendre les interfaces aussi simple, minimaliste et intuitive

qu'on a pu car nous estimons que ceci est capitale pour une utilisation fluide et rapide.

2.2.4 Le modèle du domaine

La modélisation des besoins par des cas d'utilisation s'apparente à une analyse fonctionnelle classique. L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins (sections . Elle peut être menée avant, en parallèle ou après cette dernière.[2] En analysant le domaine on peut concevoir le modèle du domaine qui représente les objets du monde réel dans ce domaine. Ces objets doivent contenir des attributs. Il est important de ne pas modéliser un concept par un attribut au lieu d'une classe et vice versa .

Dans la figure 2.3, on a établit le modèle du domaine de l'application à réaliser.

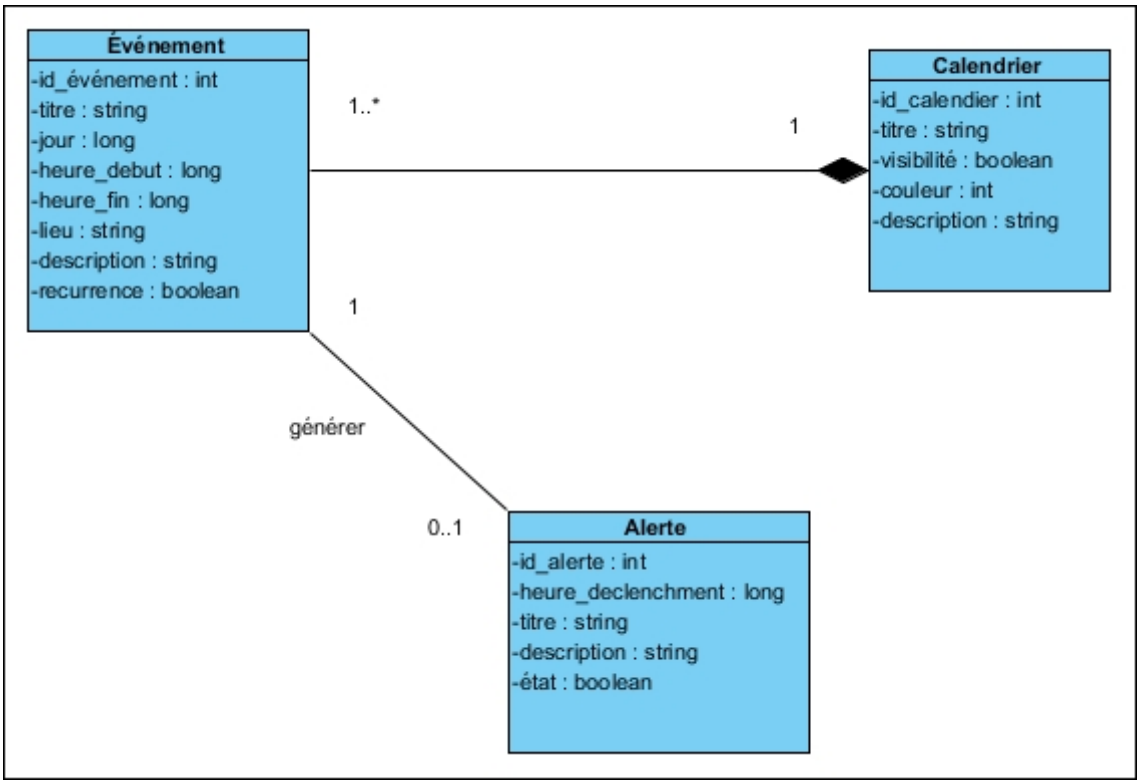


FIGURE 2.3 – Modèle du domaine.

2.2.5 Diagramme des classes participantes

Autrement dit les DCP, ils décrivent chaque cas d'utilisation c'est confrontation des diagrammes de cas d'utilisation ,des interfaces graphiques et du modèle du domaine, les trois principales classes d'analyse et leurs relations. Les règles : Pour compléter ce travail d'identification, nous allons ajouter des attributs et des opérations dans les classes d'analyse, ainsi que des associations entre elles.

— Les contrôles vont seulement posséder des opérations. Ces opérations montrent la logique de l'application, les règles transverses à plusieurs entités, bref les comportements du système informatique.
— Les dialogues vont posséder des attributs et des opérations. Les attributs représenteront des champs de saisie ou des résultats. Les résultats seront distingués en utilisant la notation de l'attribut dérivé. Les opérations représenteront des actions de l'utilisateur sur l'IHM.
— les classes entités réalisées à partir du modèle du domaine.

2.3 Conception

Après avoir tracé les grandes lignes de phase de spécification de besoins, mettons l'accent maintenant sur une phase fondamentale dans le cycle de vie d'un logiciel, la phase de conception. Dans ce chapitre, nous commençons par présenter les différents diagrammes d'interaction système. Ensuite élaborer le diagramme conceptuel. Enfin la phase de conception s'achèvera par construire les tables de modèle logique de données(MLD).

2.3.1 Diagramme d'interactions

Ces diagrammes permettent de modéliser comment les objets communiquent entre eux cite5. Pour cela, nous utiliserons encore les trois types de classes d'analyse, on représente les interactions dans un format où chaque nouvel objet est ajouté en haut à droite. On représente la ligne de vie de chaque objet par un trait pointillé vertical. Cette ligne de vie sert de point de départ ou d'arrivée à des messages représentés eux-mêmes par des flèches horizontales. Par convention, le temps coule de haut en bas. Il indique ainsi visuellement la séquence relative des envois et réceptions de messages.

nous avons décidé d'inclure deux diagrammes seulement qui concerne l'ajout d'un événement et la suppression d'un calendrier.

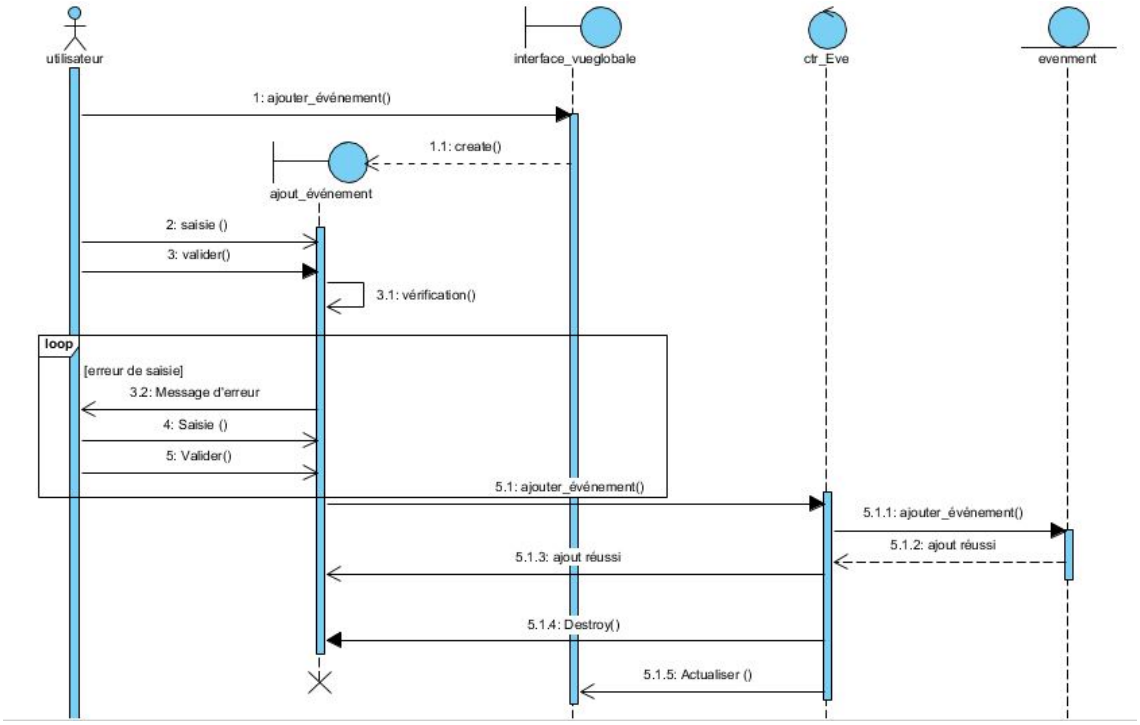


FIGURE 2.4 – Diagramme d'interaction «Ajout événement ».

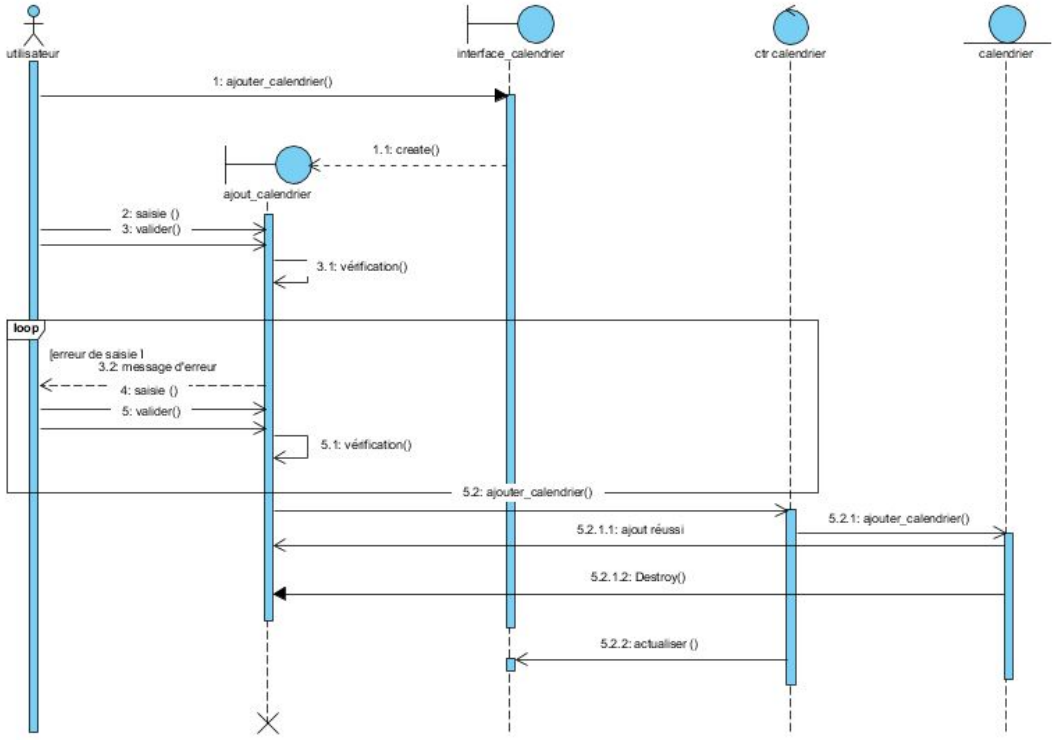


FIGURE 2.5 – Diagramme d'interaction «Suppression d'un calendrier».

2.3.2 Diagramme de classe ce conceptions

L'objectif de cette étape est de produire le diagramme de classes qui servira pour l'implémentation, Les diagrammes d'interaction vont permettre d'élaborer le diagramme de classes de conception, celui-ci est un document indispensable qui représente la vue de conception statique d'un système [6].

2.3.3 Modèle relationnel

Le modèle relationnel représente la base de données comme un ensemble de tables, sans préjuger de la façon dont les informations sont stockées dans la machine. Les tables constituent donc la structure logique du modèle relationnel. Au niveau physique, le système est libre d'utiliser n'importe quelle technique de stockage dès lors qu'il est possible de relier ces structures à des tables au niveau logique. Les tables ne représentent donc qu'une abstraction de l'enregistrement physique des données en mémoire [5].

Dans notre contexte, nous avons établi ce schéma :

— Calendrier(**id_calendrier**,titre,visibilité,couleur,description).

— Événement(**id_evenement**, titre, jour, heure_debut, heure_fin, lieu, description, récurrence,#**id_calendrier**,#**id_alerte**).

— Alerte(**id_alerte**, heure_declenchement, titre, description, état).

2.4 Conclusion

Texte ...

Chapitre 3

Implémentation

3.1 Introduction

Ce dernier chapitre est consacré à la partie pratique de la réalisation de notre projet. Dans un premier temps, nous allons énumérer les différents outils de développement qui nous ont permis de mener à bien notre application mobile. Ensuite, nous allons présenté les différents langage de programmation utilisés, les librairies et enfin les différentes interfaces de notre application.

3.2 Environnement de développement

3.2.1 Android Studio

Android Studio est un environnement de développement intégré(EDI) permettant de développer des applications sous Android. Développé par Google, il se base sur l'EDI IntelliJ de JetBrains. Il offre les outils nécessaires pour développer des applications mobiles natives destinées à Android. Ainsi, il permet d'éditer des fichiers Java/Kotlin pour la partie programmation et des fichiers XML pour la partie graphique.

3.2.2 Git et GitHub

Git est un logiciel libre de gestion de versions, sous licence publique générale GNU 2. GitHub est un service web de gestion et d'hébergement de projet de développement logiciel utilisant le logiciel Git.

3.3 Outils de développement

3.3.1 SDK de Android

Le SDK (Software Developpement Kit) de Android est un ensemble d'outils de développement essentiel au développement d'application mobile sous Android, il inclut ainsi de différents outils

tel qu'un débogueur, de la documentation, un émulateur basé sur QEMU et un ensemble de bibliothèques logicielles.

3.3.2 JDK

Le JDK (Java Developement Kit) est un ensemble d'outils et de bibliothèques logicielles destinées à la programmation Java. Il est nécessaire notamment pour la compilation du code Java qui sera transformé en bytecode pour être exécuté par la Java Virtual Machine(JVM).

3.4 Langage de programmation

Java est un langage de programmation orienté objet, puissant, il a la particularité d'être portable c'est-à-dire avoir la possibilité d'exécuter les programmes écrits en Java sous n'importe quel système d'exploitation grâce a la JVM incluse dans le JDK.

XML eXtensible Markup Language (Langage de balisage extensible en français) est un métalangage informatique de balisage générique. Il permet ainsi de structurer des données grâce à des balises.

3.5 Persistance des données

Comme nous l'avons vu durant les chapitres précédents, il est primordial de stocker en permanence les données. Ceci dit, une base de données locale est suffisante dans notre contexte.

SQLite est une bibliothèque de Android qui propose un moteur de base de données relationnelles accessible par SQL.

Room est une librairie de base de données développée par Google. Elle est une couche d'abstraction à SQLite. En effet, Room facilite la gestion de la base de données, de sa création à la lecture des données en passant par leur mise à jour de manière fluide pour exploiter toute la puissance de SQLite. Son principal atout est de détecter les erreurs de SQL à la compilation du code. Elle offre aussi la possibilité d'exécuter les requêtes SQL dans différents Threads évitant ainsi de se surcharger le Thread principal, Elle permet aussi de mettre en cache des données lors de l'absence d'une connexion Internet. Elle est composée de :

Entités : Les entités est l'ensemble de classes dont chacune d'elle représente une table dans la base de données.

DAOs(Data Access Objects) : Ce sont des interfaces qui ont pour rôle de gérer toutes les requêtes SQL, elle agit comme un intermédiaire entre la base de données et le reste de l'application. Chaque entité doit avoir son propre DAO.

Base de données : Elle contient toutes les tables et toutes les données stockées.

La figure suivante résume l'architecture de Room

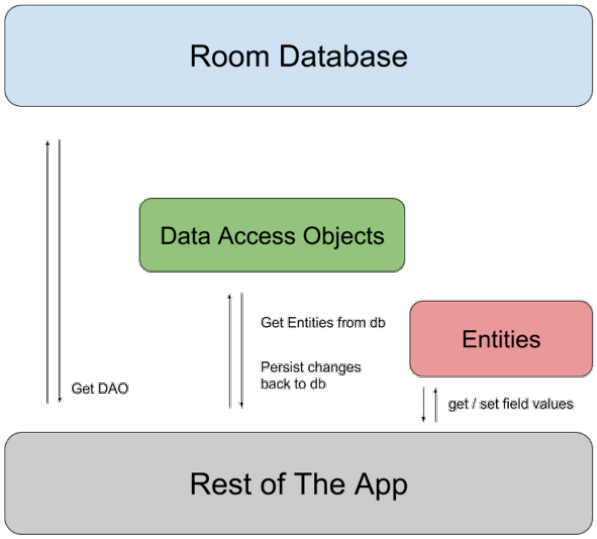


FIGURE 3.1 – Architecture de Room.

3.6 Librairies utilisées

WeekView est une librairie qui affiche la vue d'un calendrier. Elle a été utile pour l'affichage des événements sur un calendrier, elle implémente également plusieurs fonctionnalités rendant ainsi l'application plus intuitive.

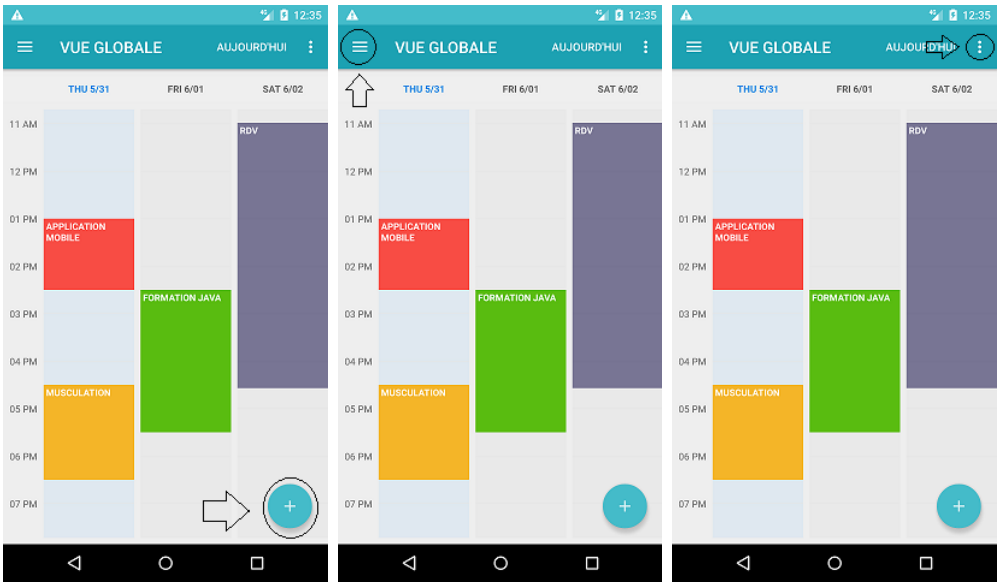
ColorPicker est comme son nom l'indique, une librairie qui permet de choisir une ou plusieurs couleurs.

Il est à noter que les librairies utilisées étaient libres, open-source et disponibles sur GitHub.

3.7 Présentation des interfaces

3.7.1 Interface d'accueil

Cette première figure présente l'interface d'accueil, ceci est la première interface affichée à l'utilisateur au lancement de l'application. Son principal composant est une vue calendrier, il regroupe les différents événements préalablement ajoutés par l'utilisateur. Chaque événement est affiché avec la couleur du calendrier auquel il appartient. Aussi, nous avons un bouton flottant(figure 3.2(a)) qui permet d'ajouter des événements, un menu latéral pour naviguer(figure 3.2(b)) entre les différentes activités de l'application et enfin un menu qui ouvre une liste déroulante pour changer la vue du calendrier (figure 3.2(c)).



(a) Bouton flottant (b) Bouton menu (c) Bouton option

FIGURE 3.2 – Interface d'accueil

3.7.2 Interface d'ajout d'un calendrier

La figure 3.3 représente l'interface d'ajout d'un calendrier, elle nous permet d'ajouter un calendrier. Pour cela, il suffit de remplir les champs et de faire le choix de la couleur avec laquelle seront affichés les événements de appartenant au calendrier.

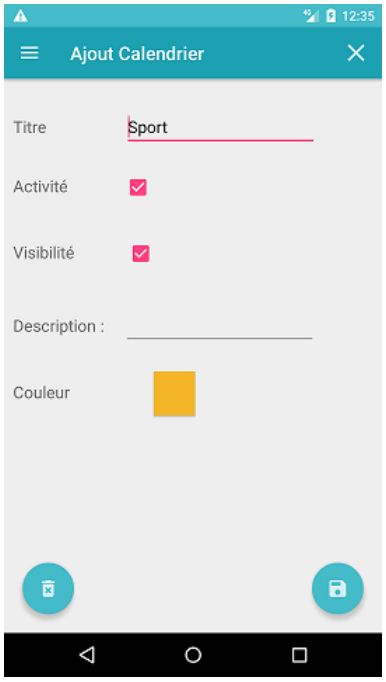


FIGURE 3.3 – Ajout d'un calendrier.

3.7.3 Interface d'ajout d'événement

La figure ci-dessous représente l'interface d'ajout d'un événement. L'utilisateur est invité à remplir les champs, à choisir à quel calendrier appartient-il et enfin décider à quel moment il souhaite être alerté voire désactiver complètement l'alerte.

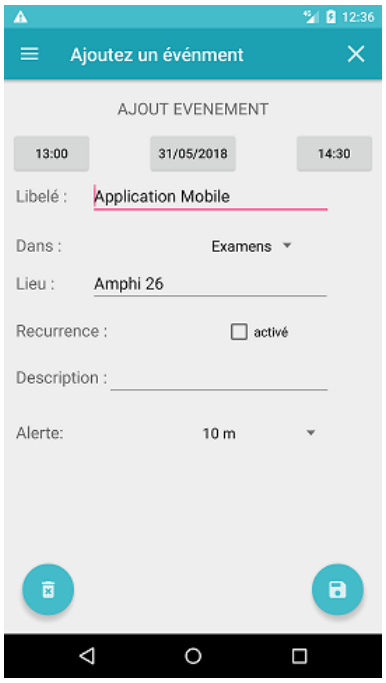


FIGURE 3.4 – Ajout d'un événement.

3.8 Conclusion

Dans ce dernier chapitre, nous avons vu les différents environnements de développement et décrit les outils utilisés notamment GitHub qui a été un outil clé pour notre travail collaboratif. Nous avons vu également les différents langage de programmation. Concernant la base de données, Room a été un véritable plus avec sa facilité à gérer et implémenter la base de données. Aussi, les différentes librairies qui nous ont permis de gagner un temps précieux. Enfin, nous avons vu les différentes interfaces principales qui composent notre application.

Conclusion générale et perspectives

Texte ...

En guise de perspectives, nous envisageons de ...

Bibliographie

[1] Pascal Roques, *UML 2 Modéliser une Application Web*, 4^{e} édition, Eyrolles, Paris, 2008.

[2] Laurent AUDIBERT,*UML 2 de l'apprentissage a la pratique* ,Publié le 31 octobre 2006 - Mis à jour le 12 janvier 2009 ,sur :https ://laurent-audibert.developpez.com/Cours-UML/

[3] Laurent Audibert,*UML 2 : de l'apprentissage à la pratique*,Ellipse,2^{e},2009

[4] http ://niedercorn.free.fr/iris/iris1/uml/uml10.pdf.

[5] http ://www.ai.univ-paris8.fr/ lysop/bd/seance4-ModeleRel.pdf.

[6] http ://tvaira.free.fr/dev/fiches/fiche-c16-diagramme_classes_conception.pdf.

Résumé

Texte en français ...

Mots clés : *mot clé 1, mot clé 2, mot clé 3, mot clé 4, mot clé 5*

Abstract

Texte en anglais ...

Keywords : *mot clé 1, mot clé 2, mot clé 3, mot clé 4, mot clé 5*