Projet Android

Classe : A2MSI

Groupe : LUMONI JOHANA, YANN DESAGNAT ET MELVIN SALVARESSE

Date de soumission : 19/12/2023

Projet android : SCHNELL



Table des matières

[Introduction 3](#_Toc153834736)

[Description de l’application 3](#_Toc153834737)

[Méthodologie de Développement 4](#_Toc153834738)

[Conception de l'Application 5](#_Toc153834739)

[Développement de l'Application 7](#_Toc153834740)

[Conclusion 9](#_Toc153834741)

[Axes d'Améliorations 9](#_Toc153834742)

# Introduction

Lors de la conception de ce projet Android, chaque membre de l'équipe a ressenti des doutes initiaux, principalement liés à la création d'une application fiable et répondant à nos idées les plus audacieuses. Nous cherchions vraiment à créer une application intéressante et amusante à la fois pour les utilisateurs et pour nous-mêmes.

Pendant la phase de brainstorming, nous avons proposé des fonctionnalités allant des plus basiques aux plus folles afin d'avoir un large éventail de choix. Même si notre application finale ne ressemble pas complètement à ce que nous avions initialement envisagé, tous les membres de l'équipe en sont sortis enrichis. Nous voulions créer une application intuitive et efficace permettant aux utilisateurs d'organiser leurs événements personnels, de créer des notes et de gérer leurs tâches pour être plus productifs.

Pourquoi avons-nous finalement choisi cette application ? En tant qu'étudiants en ingénierie en alternance, nous savons à quel point il peut être difficile de s'organiser pour gérer les différents examens et travaux notés, et c'est là qu'est née l'idée de créer une application Android.

Notre équipe est composée de Yann Desagnat, Johana Lumoni et Melvin Salvaresse. Malheureusement, durant le projet, Johana Lumoni et Yann Desagnat sont tombés malades, ce qui a impacté le développement. L'objectif principal de ce rapport est de présenter notre projet nommé Schnell.

# Description de l’application

"Schnell" est une application de productivité offrant des fonctionnalités telles que la gestion d'événements, la création de notes, la gestion de tâche et un bloc-notes. Nous avons initialement envisagé des fonctionnalités supplémentaires comme une messagerie entre utilisateurs, mais celles-ci ont été écartées en raison de contraintes de temps.

Les fonctionnalités visées de l'application étaient les suivantes :

* Gestion d'un Calendrier avec Événements & Actions (Calendrier Interactif) :
  + Création et gestion d'événements.
  + Attribution d'actions prioritaires aux événements.
* Gestion des Actions Prioritaires
  + Création, édition et suppression d'actions prioritaires.
  + Classement des actions par ordre de priorité.
* Option Bloc-Notes
  + Création, édition et suppression de notes.
  + Organisation des notes par catégories.
* Option Chronomètre
  + Fonctionnalité de chronomètre pour suivre le temps passé sur des tâches spécifiques.
  + Enregistrement du temps passé sur des actions.
* Connexion Profil
  + Gestion de l'authentification des utilisateurs.

Les fonctionnalités réelles de l’application :

* Gestion d'un Calendrier avec Événements & Actions (Calendrier Interactif) :
  + Création et gestion d'événements.
  + Attribution d'actions prioritaires aux événements.
* Option Bloc-Notes
  + Création, édition et suppression de notes.
  + Organisation des notes par catégories.
* Connexion Profil
  + Gestion de l'authentification des utilisateurs.

# Méthodologie de Développement

Nous avons développé le projet « Schnell » à l’aide du langage de programmation Java sur Android Studio et au moyen de Firebase pour la gestion des données. Les tâches étaient réparties entre les membres : Johana Lumoni s'est occupée des événements et des notes, Melvin Salvaresse de l'authentification et de la base de données, et Yann Desagnat des actions.

Lorsque nous rencontrions des difficultés, nous les signalions au professeur. Nous avons utilisé Git pour partager le code et Notion pour avoir une vue d'ensemble de l'avancement du projet. Nous avons inclus dans le planning Notion des actions prioritaires à réaliser en séance ainsi que des objectifs. Nous nous consultions tous les trois pour prendre des décisions concernant les différents aspects du projet.

En termes de planning, nous avons décidé de travailler sur le projet pendant nos heures de disponibilité prévues dans notre emploi du temps en séances. Cela semblait être la meilleure approche. Nous avons ainsi travaillé sur le projet 1 à 2 fois par semaine pendant 4 heures. Nous avons également créé des maquettes pour visualiser le projet.

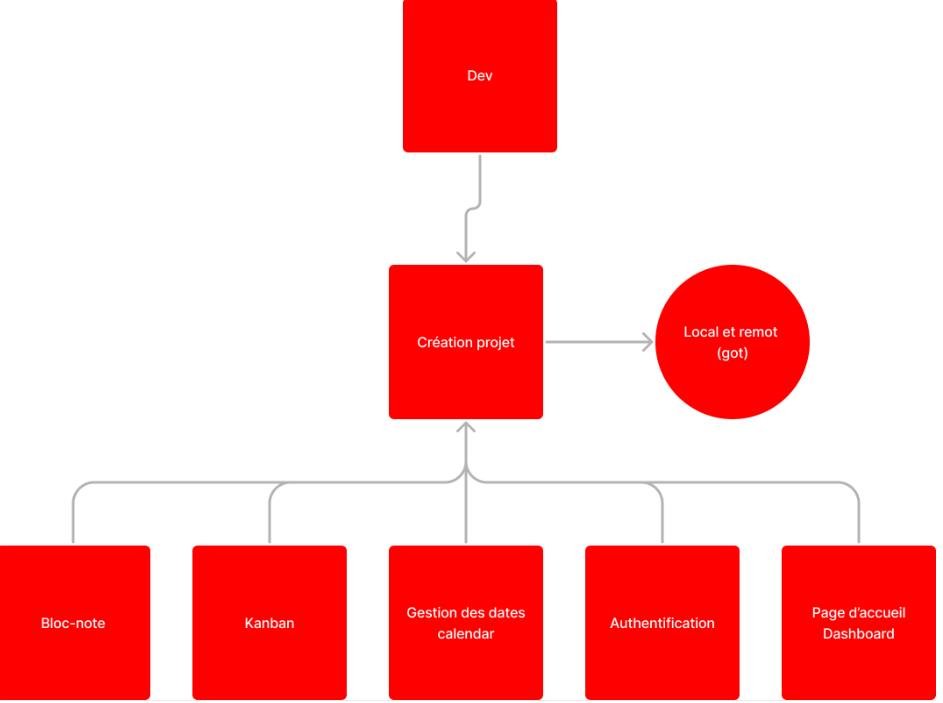


Figure 1: Page pour visualiser les pages du projet

# Conception de l'Application

Dans cette partie, nous allons vous détailler notre approche pour la conception de l'application, en expliquant le rôle de chaque membre dans le développement des différentes pages et fonctionnalités

Dans un premier temps, les membres de l’équipe ce sont mis d’accord sur la base de données que l’on souhaitait avoir :

* Utilisateur(idUtilisateur (PK),nom,prenom,email,motDePasse)
* Note(idNote (PK),contenu,idUtilisateur (FK),Categorie)
* Événement(idEvenement (PK),nom,description,date,idUtilisateur (FK))
* Action( idAction (PK),description,date,idEvenement (FK))

BASE DE DONNEE

<https://drive.google.com/file/d/1IKPPh5imrcOnwAf6GqpRBuZaFUjLOFD0/view?usp=sharing>

La base de données que nous avons créées :

CAPTURE ECRAN FIREBASE REAL DATABASE

Maintenant, que nous avons vu la base de données vous trouverez ci-dessous une explication par chaque membre de l’équipe des fichiers java et xml qu’ils ont créés :

Yann Desagnat

J’ai travaillé sur la page Trello pour la gestion des actions.

Melvin Salvaresse

Base de données : Nous avons utilisé une base de données Firebase pour stocker les données de l'application. Le modèle de données comprend les entités Utilisateur, Note, CatégorieNote, Événement, Action et plusieurs relations entre elles.

Johana Lumoni

Voici ce que j'ai accompli au final :

|  |  |
| --- | --- |
| Nom des fichiers XML et JAVA | Usage |
| HomeActivity | Page d'Accueil - J'ai développé une page d'accueil où les utilisateurs peuvent avoir accès au page du événements, calendrier et note avec Melvin Salvaresse définit l'interface utilisateur avec trois boutons, chacun étant associé à une action spécifique (ajout d'événement, ajout de note, affichage du calendrier). Lorsque l'utilisateur clique sur l'un de ces boutons, l'application navigue vers la page correspondante en utilisant des intentions. |
| AddEventAcivity | Page d'Ajout d'Événements - J'ai créé une page permettant aux utilisateurs de créer de nouveaux événements en spécifiant le nom, la description et la date de l'événement. |
| EventActivity | Page de Visualisation des Événements - J'ai mis en place une page qui permet aux utilisateurs de voir les détails de leurs événements existants.  permet il permet à l'utilisateur de visualiser les événements pour une date spécifique et avec openModifyDateActivity on peut modifier les évenements |
| Event | utilisée pour créer des objets d'événements Chaque objet d'événement a des propriétés telles que le nom, la description et la date de l'événement (composée du jour, du mois et de l'année)  Méthode updateDate : Cette méthode permet de mettre à jour la date d'un événement  Getters : Des méthodes sont fournies pour récupérer les différentes propriétés de l'événemen  Méthode getId : Il y a une méthode getId afin de donner un Id à un event |
| Event adapter | adaptateur personnalisé (EventAdapter) pour un RecyclerView L'adaptateur est conçu pour afficher une liste d'événements dans une interface utilisateur, où chaque élément de la liste représente un événement.  Initialise les composants visuels & bind Method : Associe les données d'un objet EventModel à la mise en page de l'élément de liste. - EventViewHolder Class  initialise l'adaptateur avec une liste d'objets EventModel - & onBindViewHolder : Associe les données d'un événement spécifique EventAdapter Class  openModifyDateActivity Method :  Méthode privée (openModifyDateActivity(Object id)) : Crée un Intent pour ouvrir l'activité de modification de date (ModifyDateActivity) en passant l'ID de l'événement en tant qu'extra. Cette méthode est appelée lorsque le bouton "Modifier la date" est cliqué. |
| EventModel | EventModel représente le modèle d'un événement |
| UpcomingEventsAdapter | pour alimenter une liste d'événements à venir dans un composant RecyclerView |
| AddNoteActivity.java | Cette activité est utilisée pour ajouter ou modifier des notes  Méthode onCreate – affichage de tout Récupère les données de l'intent qui a ouver cette activité. Si une note existante est passée en extra (NoteText), elle est affichée dans le champ de texte.  Méthode saveNoteToDatabase -Elle récupère le texte de la note et la catégorie sélectionnée. La logique pour enregistrer la note dans la base de données |
| NoteActivity | cette activité permet à l'utilisateur de voir les notes, d'en ajouter de nouvelles, de les éditer en cas de clic long, et de filtrer les notes par catégorie à l'aide d'un Spinner |
| CalendarActivity | permet de visualiser les évènements à venir en fonction de la date sélectionner sur le calendrier  Element calendat view pour affichage du calendrier et calendarView.setOnDateChangeListener pour les changements de date  recyclerViewUpcomingEvents affichage d’une liste déroulante des évenements à venir  methode getUpcomingEventsForDate pour obtenir les évenements à venir  Element calendat view pour affichage du calendrier et calendarView.setOnDateChangeListener pour les changements de date |
| ModifyDateActivity | cette activité permet à l'utilisateur de sélectionner une nouvelle date à l'aide d'un DatePicker et de sauvegarder les modifications en cliquant sur le bouton "Save Changes". |

J'ai également utilisé Firebase pour stocker les données des utilisateurs et leurs événements. Cela a permis une gestion efficace des données et la synchronisation en temps réel entre les appareils. De plus, j'ai utilisé l'API ChatGPT pour obtenir de l'aide dans le développement de certaines parties du projet, notamment la gestion des données Firebase et la résolution d'erreurs.

Maintenant que nous avons vu toutes les pages crées dans le projet « schnell » nous allons vous parler dans la partie suivante des défis rencontrés et de quelques fonctionnalités

# Développement de l'Application

Cette section aborde les défis techniques rencontrés, les solutions adoptées, et les exemples de code utilisés pour certaines fonctionnalités clés.

Diagramme de Séquence pour "Schnell"

Le diagramme de séquence démontre comment les différents composants de l'application interagissent entre eux lors de l'exécution d'une tâche spécifique. Prenons l'exemple de l'ajout d'un événement :

Acteurs : Utilisateur, Interface d'Accueil, Page d'Événement, Firebase Database

Processus :

Utilisateur : Sélectionne l'option pour ajouter un événement depuis l'Interface d'Accueil.

Interface d'Accueil : Redirige l'utilisateur vers la Page d'Événement.

Utilisateur : Remplit les détails de l'événement (titre, description, date) et soumet l'information.

Page d'Événement : Valide les données saisies et envoie la requête à Firebase Database.

Firebase Database : Stocke les détails de l'événement et envoie une confirmation de succès à la Page d'Événement.

Page d'Événement : Affiche un message de succès à l'Utilisateur.

Ce diagramme illustre la séquence d'actions depuis la sélection de l'option pour ajouter un événement jusqu'à la confirmation de son enregistrement.

Mise en œuvre des fonctionnalités clés

Intégration de services tiers (le cas échéant)

Tests unitaires et de régression

Gestion des erreurs et des exceptions

Tests et Validation

Nous avons mis en place une stratégie de test comprenant des tests unitaires, des tests d'intégration et des tests de régression. Les résultats des tests ont été positifs, et nous avons également sollicité la validation des utilisateurs pour recueillir leurs retours.

# Conclusion

Ce projet Android a été une expérience d'apprentissage enrichissante. Malgré les défis rencontrés en cours de route, nous avons réussi à développer une application fonctionnelle répondant aux besoins de base de gestion d'événements. Avec les améliorations à venir, nous espérons fournir une application encore meilleure à nos utilisateurs.

# Axes d'Améliorations

Ajout de la fonctionnalité de messagerie entre utilisateurs pour une meilleure coordination des événements.

Améliorations continues de l'interface utilisateur pour une expérience plus conviviale.

En résumé, ce projet a été un voyage d'apprentissage passionnant, et nous sommes fiers de ce que nous avons accompli jusqu'à présent.