3259.2 Développement mobile II– Rapport technique – ISC3il-b

|  |
| --- |
| **Shake ToDo** |

|  |
| --- |
| Étudiants participant à ce travail :  **Titus Abele, ISC3il-a Nicolas Aubert, ISC3il-b Tim Peck, ISC3il-a**  Présenté à :  **Aïcha Rizzotti Guillaume Digier**  Restitution du rapport : **12.05.2023**  Période : **2022 – 2023**  École : **HE-Arc, Neuchâtel** |

**Application de gestion des tâches, réalisée avec Android Studio**

Table des matières

[1 - Introduction 2](#_Toc133567209)

[2 - Analyse 3](#_Toc133567210)

[2.1 - Objectifs 3](#_Toc133567211)

[2.1.1 - Principaux 3](#_Toc133567212)

[2.1.2 - Secondaires 3](#_Toc133567213)

[2.2 - Maquettes 3](#_Toc133567214)

[3 - Conception 5](#_Toc133567215)

[3.1 - Base de données 5](#_Toc133567216)

[3.2 - Capteurs 5](#_Toc133567217)

[3.2.1 - Prise de photos 5](#_Toc133567218)

[3.2.2 - Accéléromètre 5](#_Toc133567219)

[3.3 - Notifications 5](#_Toc133567220)

[3.4 - Settings 5](#_Toc133567221)

[4 - Résultats 6](#_Toc133567222)

[4.1 - Visualisation des ToDos 6](#_Toc133567223)

[4.2 - Ajout d’un ToDo 6](#_Toc133567224)

[4.3 - Suppression d’un ToDo 6](#_Toc133567225)

[4.4 - Modification d’un ToDo 6](#_Toc133567226)

[4.5 - Compléter un ToDo 6](#_Toc133567227)

[5 - Limitations et perspectives 7](#_Toc133567228)

[6 - Conclusion 8](#_Toc133567229)

[7 - Annexes I](#_Toc133567230)

[7.1 - Table des illustrations I](#_Toc133567231)

[7.2 - Bibliographies et références II](#_Toc133567232)

[7.2.1 - Sites Web II](#_Toc133567233)

[7.2.2 - Livres II](#_Toc133567234)

[7.2.3 - Autres II](#_Toc133567235)

1. Introduction

L'application a été construite en utilisant le langage de programmation Kotlin et Android Studio comme environnement de développement. L'application est construite sur l'Android Framework, qui fournit une base solide pour la fonctionnalité et la performance de l'application. Le cadre Android comprend une variété d'outils et d'API qui sont essentiels pour le développement de toute application Android, y compris le SDK Android, le Runtime Android et la bibliothèque de support Android. Cette présentation fournira un aperçu détaillé des fonctionnalités, de la conception et de la mise en œuvre de l'application, ainsi que de tous les défis rencontrés au cours du processus de développement et de leurs solutions. En outre, il inclura les résultats des tests utilisateurs et les plans futurs pour l'application.

Le développement de l'application s'est appuyé sur l'environnement de développement intégré (IDE) Android Studio. Kotlin, le langage de programmation utilisé pour l'application, est un langage moderne, concis et expressif qui est totalement interopérable avec Java et qui offre plusieurs fonctionnalités telles que la sécurité des nullités, les fonctions d'extension et les coroutines qui améliorent la lisibilité et la maintenabilité du code. Android Studio, quant à lui, est l'IDE officiel pour le développement d'Android. Il propose une variété d'outils et de fonctionnalités qui simplifient le processus de développement, notamment un éditeur de mise en page visuelle, un éditeur de code et un débogueur. Ensemble, Kotlin et Android Studio constituent une plateforme de développement puissante et efficace qui permet de créer des applications mobiles de haute qualité.

L'application, ShakeToDo, est conçue pour fournir aux utilisateurs un moyen simple et efficace de gérer leurs tâches. La principale fonction de l'application est de permettre à l'utilisateur de créer des "ToDo", c'est-à-dire des tâches qui doivent être accomplies avant une date limite. L'utilisateur peut facilement ajouter des images à une tâche pour la visualiser et la rendre plus mémorable. En outre, l'application offre une fonction unique qui permet à l'utilisateur de secouer son téléphone lorsqu'une tâche est sélectionnée pour la marquer comme "terminée". Cette fonction constitue un moyen rapide et pratique pour l'utilisateur de suivre ses progrès et de rester au fait de ses tâches. Globalement, ShakeToDo offre une interface conviviale et visuellement attrayante qui rend la gestion des tâches simple et efficace pour l'utilisateur.

1. Analyse
   1. Objectifs
      1. Principaux

* CRUD ToDos
* Interface utilisateur simple et intuitive
* Intégration caméra et accéléromètre pour ajouter des interactions avancés
  + 1. Secondaires
* Ajout de fonctionnalité "alarme" ou "notification" pour rappel à l'utilisateur
  1. Maquettes



Figure - Maquettes de l'application

L'interface de l'application ShakeToDo est conçue pour être intuitive et facile à utiliser pour les utilisateurs. Elle présente une interface visuellement attrayante qui permet à l'utilisateur de visualiser facilement ses tâches et de les gérer efficacement. La principale fonction de l'application est accessible à partir de l'écran principal, où l'utilisateur peut voir la liste de ses tâches en cours, avec des options pour ajouter une nouvelle tâche, marquer une tâche comme terminée ou supprimer une tâche existante.

Les tâches sont affichées de manière claire et concise, avec une description courte, une date limite et éventuellement une image associée.

Les éléments de l'interface sont disposés de manière logique et facile à comprendre pour l'utilisateur, avec des boutons et des icônes clairement étiquetés pour les différentes fonctions. Les menus et les sous-menus sont également bien organisés pour une navigation facile. La fonction unique de secouer le téléphone pour marquer une tâche comme terminée est également intégrée de manière transparente dans l'interface, avec des instructions claires pour l'utilisateur. En somme, l'interface de ShakeToDo est conçue pour offrir une expérience utilisateur fluide et agréable pour la gestion des tâches.

1. Conception
   1. Base de données

* DOA, Room
* ExecutorService pour effectuer une action CRUD, sans pour autant bloquer l’interface
  + Exécuté dans un thread séparé
* ??
  1. Capteurs
     1. Prise de photos
* Package androidx.camera
  + Demande de permissions (appareil photo et micro)
* Activité dédiée pour ça -> Réutilisable facilement (elle retourne l’image capturée)
* Image enregistrée dans la galerie photos à l’aide de MediaStore
  + 1. Accéléromètre
* Dans l’activité qui affiche un ToDo
* SensorManager
  + Gérer les accès aux capteurs
  + Ajout d’un listener sur le capteur accéléromètre
* Dans ce listener :
  + Récupération de l’accélération, comparaison avec l’accélération précédente -> Delta
  + Si ce delta dépasse une certaine valeur = secousse du téléphone
* La valeur du delta à dépasser est récupérée via les Shared Preferences, présentés dans le chapitre suivant, qui parle du menu des paramètres

Balancer ça à gepeto pour qu’il génère des paragraphes

* 1. Settings
* Les capteurs (sensibilité, valeurs trouvées) changent en fonction du téléphone -> Solution : ajouter un menu « paramètres » permettant à l’utilisateur de saisir les valeurs qu’il souhaite afin d’adapter la configuration à son téléphone
* Slider qui permet de définir la sensibilité de détection de secousses (min = 5, max = 25)
* Lorsque la valeur de ce slider est modifiée, celle-ci est enregistrée dans les Shared Preferences (stockage local, persistant, sous forme de clé-valeur)
  + Preferences, unique à une activité, privée
  + Shared Preferences, commune à toute l’application (toutes les activités) -> Pile ce que l’on souhaite
* Cette valeur est ensuite récupérée lors de la détection de secousse
* Les Shared Preferences sont dépréciées -> Il est conseillé d’utiliser DataStore, plus récent, async pour de meilleures performances

Balancer ça à gepeto pour qu’il génère des paragraphes

* 1. Notifications
* Lorsqu’un todo est ajouté
* Background service
* ??

1. Résultats
   1. Visualisation des ToDos
   2. Ajout d’un ToDo
   3. Suppression d’un ToDo
   4. Modification d’un ToDo
   5. Compléter un ToDo
2. Limitations et perspectives

La réalisation de l'application mobile de gestion de tâches à l'aide d'Android Studio a permis de créer une application pratique et fonctionnelle pour la gestion des todos. Cependant, il y a des limitations et des perspectives à considérer pour améliorer l'application.

Tout d'abord, une limitation importante est la visualisation des todos sous forme de calendrier. Actuellement, l'application ne permet pas de voir les tâches dans un calendrier, ce qui est une fonctionnalité très demandée par les utilisateurs. Pour améliorer l'application, il serait donc important d'implémenter une vue calendrier pour la gestion des tâches. De plus, il serait utile de permettre la synchronisation de l'application avec un autre calendrier, comme Google Calendar, pour une gestion plus globale et une synchronisation en temps réel.

Une autre limitation de l'application est liée à la version d'API utilisée. Actuellement, l'application utilise la version 26 d'API, ce qui est relativement ancien. Cela signifie qu'il est possible que l'application ne fonctionne pas toujours avec les nouvelles versions d'Android. Pour améliorer l'application, il serait donc nécessaire de mettre à jour la version d'API utilisée afin de s'assurer que l'application fonctionne correctement sur toutes les versions d'Android.

En outre, il y a eu des soucis avec Gradle et Android Manifest, car les versions utilisées ont affecté les imports. Pour améliorer l'application, il serait donc important de s'assurer que les versions utilisées sont compatibles avec les imports et de mettre à jour si nécessaire.

Une perspective intéressante pour améliorer l'application serait d'ajouter une fonctionnalité de Markdown pour les textes des todos. Cela permettrait aux utilisateurs de formater le texte de leurs tâches, en ajoutant des titres, des liens ou des images, pour une meilleure organisation et une meilleure lisibilité.

En outre, actuellement, l'application ne permet qu'une seule image par todo. Il serait donc intéressant d'ajouter la possibilité d'ajouter plusieurs images pour chaque tâche, pour une meilleure visualisation et organisation des tâches.

Une autre limitation importante est liée aux notifications. Actuellement, les notifications ne fonctionnent pas avec toutes les versions des API, ce qui peut causer des problèmes pour les utilisateurs qui ne reçoivent pas de notifications pour leurs tâches. Pour améliorer l'application, il serait donc important de trouver une solution pour permettre aux notifications de fonctionner sur toutes les versions des API.

Enfin, une perspective intéressante serait d'ajouter une fonctionnalité d'alarme pour les tâches importantes. Cela permettrait aux utilisateurs de ne pas oublier les tâches importantes et d'être rappelés à temps pour les réaliser.

En somme, l'application mobile de gestion de tâches créée à l'aide d'Android Studio a été très utile pour les utilisateurs, mais il reste des limitations et des perspectives à considérer pour améliorer l'application et répondre aux besoins des utilisateurs de manière plus efficace.

1. Conclusion

En conclusion, l'application mobile de gestion de tâches développée à l'aide d'Android Studio est fonctionnelle et a rempli tous les objectifs prévus initialement. Les utilisateurs peuvent facilement gérer leur liste de tâches, en ajoutant du texte et des images pour une meilleure organisation. De plus, la bonne UI/UX permet une utilisation intuitive de l'application, ce qui est un élément clé pour une expérience utilisateur satisfaisante.

En outre, l'application est évolutive grâce à son architecture bien conçue. Les fonctionnalités futures, telles que la visualisation de la liste de tâches sous forme de calendrier, la synchronisation avec un autre calendrier ou la possibilité d'ajouter plusieurs images pour chaque tâche, peuvent être implémentées facilement en utilisant les mêmes principes d'architecture.

En résumé, l'application mobile de gestion de tâches créée à l'aide d'Android Studio est une solution pratique et efficace pour la gestion des tâches. Elle a rempli tous les objectifs initialement prévus, avec une bonne UI/UX et une architecture évolutive. Cette application peut être une réponse satisfaisante aux besoins des utilisateurs pour la gestion de leurs tâches quotidiennes.

1. Annexes
   1. Table des illustrations

[Figure 1 - Maquettes de l'application 4](#_Toc133567108)

* 1. Bibliographies et références
     1. Sites Web

Prendre des photos. (n.d.). *Android Developers*. <https://developer.android.com/training/camera-deprecated/photobasics?hl=fr>

Enregistrer des données dans une base de données locale à l’aide de Room. (n.d.). *Android Developers*. <https://developer.android.com/training/data-storage/room?hl=fr>

Mettre à jour les composants de l’interface utilisateur avec NavigationUI. (n.d.). *Android Developers*. <https://developer.android.com/guide/navigation/navigation-ui?hl=fr>

*Notifications are working on an emulator but not on a real device*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/51667324/notifications-are-working-on-an-emulator-but-not-on-a-real-device>

Créer des icônes d’application. (n.d.). *Android Developers*. <https://developer.android.com/studio/write/create-app-icons?hl=fr>

Spinners. (n.d.). *Android Developers*. <https://developer.android.com/develop/ui/views/components/spinner>

GeeksforGeeks. (2022). How to Detect Shake Event in Android. *GeeksforGeeks*. <https://www.geeksforgeeks.org/how-to-detect-shake-event-in-android/>