**Bản Nháp Báo Cáo Niên Luận Cơ Sở**

## Giới thiệu bài toán:

In today's fast-paced world, individuals often struggle to keep track of important information and tasks efficiently. The problem lies in the inefficiency and disorganization that often accompanies traditional note-taking methods. With the abundance of information we encounter daily, it becomes increasingly challenging to capture and manage essential details effectively.

As a result, there is a growing demand for a simple and user-friendly note-taking application that can simplify the process of recording, organizing, and accessing information on mobile devices. To address this problem, we createad the "Simple Note" app, designed specifically for the Android operating system.

The central objective of this topic is to develop a user-friendly and efficient note-taking application for Android users. The app aims to provide seamless functionality for creating text-based notes and checklists, setting reminders for tasks, and tracking progress of the reminder’s tasks. By addressing these objectives, the app seeks to improve users' ability to organize information effectively and enhance their overall productivity.

## Công nghệ áp dụng:

### Android Studio

Android Studio is an integrated development environment (IDE) specifically designed for Android app development. Developed by Google, it provides a comprehensive set of tools for building, testing, and debugging Android applications. It streamlines the development process as well as offering seamless integration with the Android SDK, enabling developers to access a wide range of APIs and libraries to enhance their app functionalities.

### Kotlin

Kotlin is a modern programming language developed by JetBrains, known for its conciseness, safety, and interoperability with existing Java code. Officially supported by Google for Android app development, Kotlin offers features such as null safety, extension functions, and coroutines, enhancing developer productivity and code readability.

### SQLite

SQLite is a lightweight, self-contained, and open-source relational database management system (RDBMS) widely used in embedded systems and mobile applications. Designed for simplicity and efficiency, SQLite requires minimal setup and configuration, making it ideal for small to medium-sized. Despite its compact size, SQLite supports most of the SQL features found in larger database engines, including transactions, triggers, and joins. It operates directly on the device's storage, eliminating the need for a separate server process, which results in faster access times and reduced overhead.

## Giải pháp:

### A diagram of a person with blue circles Description automatically generated3.1. Use Case 1 (Navigation Drawer):

|  |  |
| --- | --- |
| **Use case:** Navigation Drawer | **ID:** UC1 |
| **Main actor:** User | **Priority:** 1 |
| **Brief description:** A menu allowing user to move between activities. | |
| **Trigger:** User taps on the hamberger button on the toolbar.  **Type:** External. | |
| **Relationship:**  **+ Association:** Go to Note menu, Go to Checklist Menu, Go to Reminder Menu, Go to Progress Tracker Menu.  **+ Include:** Open Note Menu, Open Checklist Menu, Open Reminder Menu, Open Progress Tracker Menu  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “ Go to Note menu” button.  2. The app opens the note activity menu screen. | |
| **Alternative flow:** 1. If the user is already on the note activity menu.  2. The user taps on the “ Go to Checklist menu” button to open the checklist menu.  3. The user taps on the “ Go to Reminder menu” button to open the Reminder menu.  4. The user taps on the “ Go to Progress Tracker menu” button to open the progress tracker menu. | |

- Inflating the hamburger icon on the toolbar:

A screenshot of a computer screen

Description automatically generated

A computer screen shot of a computer code

Description automatically generated- Setting the open/close behaviour of the navigation drawer:

A black background with white text

Description automatically generated- Checking current activity method:

- Moving to another activity with the navigation drawer:  
A screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

### A diagram of a person with text Description automatically generated3.2. Use case 2 (Note Menu):

|  |  |
| --- | --- |
| **Use case:** Note Menu | **ID:** UC2 |
| **Main actor:** | **Priority:** 1 |
| **Brief description:** the note menu where user can add, edit or delete notes. | |
| **Trigger:** User opens the app or move into the note menu by the navigation drawer.  **Type:** External. | |
| **Relationship:**  **+ Association:** Open navigation Menu, Add new Node, View existing Notes.  **+ Include:** Open the add new Note activity.  **+ Extend:** Delete Note, Edit Note.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps the “Add new Note” button on the toolbar.  2. <<Use case 3>>. | |
| **Alternative flow:** 1. If the user already created another note before, they can tap on the “Delete Note” button to delete that note.  2. If the user already created another note before, they can tap on the “Edit Note” button to edit that note. <<Use case 4>>.  3. The user can move to another acitivity by tapping on the “Open navigation Menu” button. <<Use case 1>>. | |

A computer screen with text

Description automatically generated- Inflating the “Add Note” button on the toolbar:

- Setting the behaviour of the “Add Note” Button:  
A screen shot of a computer

Description automatically generated

- Displaying the notes on the recycle view throught the get all notes method:A screenshot of a computer program

Description automatically generated

- Inflating the recycle view:  
A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated- Setting the behaviour of the “Add Note” and “Delete Note” buttons:

A computer screen with text

Description automatically generated- The delete note method:

### A diagram of a user Description automatically generated3.3. Use case 3 (Add Note):

|  |  |
| --- | --- |
| **Use case:** Add Note | **ID:** UC3 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The activity where the user can add a new note. | |
| **Trigger:** The user taps on the “Add New Note” button on the toolbar in the Note Menu activity.  **Type:** External. | |
| **Relationship:**  **+ Association:** Enter Note Title, Enter Note Description, Save Note.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user writes the note title to the “Enter Note Title” edit text.  2. The user writes the note content to the “Enter Note Description” edit text.  3. The user taps the “Save Note” button to create and save a new note to the database. | |
| **Alternative flow:** 1. If the “Save Note” button is taps first, the newly created note will have an empty title and content field.  2. If the user taps on the android backpress button “<”, they will be returned to the note menu. | |

A screen shot of a computer code

Description automatically generated- Insert note method:

- Setting the behaviour for the “Save Note” button:

A screen shot of a computer code

Description automatically generated

### A diagram of a person with text Description automatically generated3.4. Use case 4 (Edit Note):

|  |  |
| --- | --- |
| **Use case:** Edit Note | **ID:** UC4 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The activity where user can edit an existing note. | |
| **Trigger** The user taps on the “Edit Note” on a note item in the note menu’s recycle view.  **Type:** External. | |
| **Relationship:**  **+ Association:** Edit Note Title, Edit Note Description, Save Edited Note.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user edits the note title to the “Edit Note Title” edit text.  2. The user edits the note content to the “Edit Note Description” edit text.  3. The user taps the “Save Edited Note” button to save the edited note to the database. | |
| **Alternative flow:**  1. If the user taps on the android backpress button “<”, they will be returned to the note menu. | |

- Getting the editing note’s ID method:  
A computer screen shot of a program code

Description automatically generated

A screen shot of a computer code

Description automatically generated- The update note method:

- Getting the editing note’s ID and setting the “Update Button” hehaviour:

A screen shot of a computer program

Description automatically generated

### A diagram of a person with blue circles Description automatically generated3.5. Use case 5 (Checklist Menu):

|  |  |
| --- | --- |
| **Use case:** Checklist Menu | **ID:** UC5 |
| **Main actor:** User | **Priority:** 1 |
| **Brief description:** The menu where user can create, update and delete checklists. | |
| **Trigger:** The user taps the “Go To Checklist Menu” button on the navigation drawer.  **Type:** External. | |
| **Relationship:**  **+ Association:** Open Navigation drawer, Add new Checklist, View existing checklist.  **+ Include:** Open the add new checklist dialog.  **+ Extend:** Delete Checklist, Edit Checklist.  **+ Generalization:** None. | |
| **Normal flow:**  1. User taps on the “Add new Checklist” button.  2. <<Use case 6>>  3. User taps on the “Edit Checklist” button to edit a checklist.  4. <<Use case 7>> | |
| **Alternative flow:** 1. If the user taps on the “Delete Checklist” button to delete said checklist.  2. The user can move to another acitivity by tapping on the “Open navigation Menu” button. <<Use case 1>>. | |

A computer screen with text

Description automatically generated- Inflating the “Add Checklist” button on the toolbar:

A screen shot of a computer code

Description automatically generated- Setting the behaviour of the “Add Checklist” Button:

- Get all checklists for the recycle view: A computer screen shot of a computer code

Description automatically generated

- Inflating the recycle view:  
A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated- Setting the behaviour of the “Add Checklist” and “Delete Checklist” buttons:

- The delete checklist and all of its checklist items method:

A computer screen shot of a checklist

Description automatically generated

### A diagram of a person with text Description automatically generated3.6. Use case 6 (Add Checklist dialog):

|  |  |
| --- | --- |
| **Use case:** Add Checklist dialog | **ID:** UC6 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The dialog where the user create a new checklist with a title. | |
| **Trigger:** The user taps the “Add new Checklist” button on the toolbar.  **Type:** External. | |
| **Relationship:**  **+ Association:** Enter Checklist Title, Cancel, Confirm.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user enter the checklist title on the “Enter Checklist Title” edit text.  2. The user taps on the “Confirm” button to create a new checklist. | |
| **Alternative flow:**  1. If the user taps on the “Cancel” button to return to the checklist menu.  2. If the user taps on the “Confirm” button with the “Enter Checklist Title” empty,  a new checklist is created with the title empty.  3. If the user taps on the android backpress button “<”, they will be returned to the checklist menu. | |

- The method for showing the add checklist dialog when the “Add new Checklist” button is clicked and setting the behaviour for the “Cancel” and “Confirm” buttons.

A computer screen shot of a program

Description automatically generated

A screen shot of a computer

Description automatically generated- Add checklist method:

### A diagram of a person with blue circles Description automatically generatedUse Case 7 (Edit Checklist dialog):

|  |  |
| --- | --- |
| **Use case:** Edit Checklist | **ID:** UC7 |
| **Main actor:** User | **Priority:** 1 |
| **Brief description:** The activity where the user edits the checklist title, adding or deleting checklist items. | |
| **Trigger:** The user taps the “Edit Checklist” button of a checklist item in the recycle view.  **Type:** External. | |
| **Relationship:**  **+ Association:** Save Checklist, Add Checklist Item, Edit Checklist Title, View Item.  **+ Include:** Open Add new item dialog.  **+ Extend:** Check, Delete Item, Edit Item Title.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “Add Checklist Item”.  2. <<Use case 8>>  3. The user taps on the “Save Checklist” to save all changes made.  4. The user taps on the “Check” button of each of the item to check the checkbox. | |
| **Alternative flow:**  1. The user can edit the checklist title by tapping on and rewrite the text on the  “Edit Checklist Title” text view.  2. If the user taps on the “Delete Item”, a checklist.  3. If the user taps on the android backpress button “<”, they will be returned to the checklist menu. | |

- Inflate and set the behaviour for the “Save Checklist” button, “Add Checklist Item” button, “Edit Checklist Title” text view and “View Item” recycle view.

A computer screen shot of a program

Description automatically generated

- Managing the behaviour of the checkbox.

A computer screen shot of a program

Description automatically generated

- Setting the behaviour of the “Delete Item” button.

A screen shot of a computer code

Description automatically generated

- Insert item and get current checklist ID method.

A computer screen shot of a program

Description automatically generated

- Update item and delete item method.

A screen shot of a computer code

Description automatically generated

- Get all checklist item (for the recycle view):

A computer screen with text on it

Description automatically generated

- Update the checkbox state and get item by ID method.

A computer screen shot of a program code

Description automatically generated

### A diagram of a person with blue circles Description automatically generated3.8. Use case 8 (Add Item dialog):

|  |  |
| --- | --- |
| **Use case:** Add Item dialog | **ID:** UC8 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The dialog where user can add a new checklist item. | |
| **Trigger:** The user taps on the “Add Checklist Item” button in the Edit Checklist activity.  **Type:** External. | |
| **Relationship:**  **+ Association:** Enter Checklist Item Content, Cancel, Confirm.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user enter the checklist item content by tapping the “Enter Checklist Item Content” text view.  2. The user taps the “Confirm” button to create a new checklist item | |
| **Alternative flow:**  1. If the user taps on the “Cancel” button to return to the checklist menu.  2. If the user taps on the “Confirm” button with the “Enter Checklist Item Content” empty, a new checklist item is created with a empty content.  3. If the user taps on the android backpress button “<”, the dialog will be closed. | |

- Setting the behaviour of the “Confirm” button, “Cancel” button and the “Enter Checklist Item content” text view.

A computer screen shot of a program

Description automatically generated

- Update the checklist item content method.

A screen shot of a computer program

Description automatically generated

### A diagram of a person Description automatically generated3.9. Use case 9 (Reminder Menu):

|  |  |
| --- | --- |
| **Use case:** Reminder Menu | **ID:** UC9 |
| **Main actor:** User | **Priority:** 1 |
| **Brief description:** The menu where user can create, update and delete reminders. | |
| **Trigger:** The user taps the “Go To Reminder Menu” button on the navigation drawer.  **Type:** External. | |
| **Relationship:**  **+ Association:** Open Navigation Drawer, Add new Reminder, View existing Reminder.  **+ Include:** Open the add new Reminder Dialog.  **+ Extend:** Delete Reminder, Edit Reminder, Change State.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “Add new Reminder” button.  2. <<Use case 10>>  3. When the reminder set time is matched with real time, the notification will be shown. The user can swipe left or right to close the notification. | |
| **Alternative flow:**  1. If the user already created another reminder before, they can tap on the “Delete Reminder” button to delete that reminder.  2. If the user already created another reminder before, they can tap on the “Edit Reminder” button to edit that reminder. <<Use case 11>>.  3. The newly created reminder is initiated with the notification set to active, to set the notification state to inactive the user can tap on the “Change State” switch and vice versa.  4. The user can move to another acitivity by tapping on the “Open navigation Menu” button. <<Use case 1>>.  5. If the user taps on the notification, they will be brought to the Progress Tracker activity <<Use Case 12>> | |

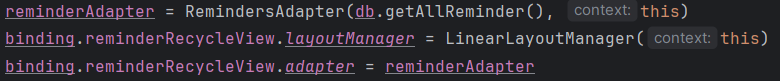
A screen shot of a computer

Description automatically generated- Inflate the “Add new Reminder” button on the toolbar:

A screen shot of a computer program

Description automatically generated- Set behaviour for the “Add new Reminder” button:

- Inflating the recycle view:



- Display the reminders on the recycle view through the get all reminder method:

A computer screen shot of a program

Description automatically generated

- Setting the behaviour of the “Delete Reminder” button and the notification switch:  
A computer screen shot of a program

Description automatically generated



A computer screen shot of a program

Description automatically generated- Delete reminder and update reminder state methods:

- The broadcast reciever for the notification:

A screen shot of a computer program

Description automatically generated

- Create notification:

A computer screen with text on it

Description automatically generated

- Schedule notification method:

A computer screen shot of a program

Description automatically generated

### A diagram of a person with text Description automatically generated3.10. Use case 10 (Add Reminder dialog):

|  |  |
| --- | --- |
| **Use case:** Add Reminder dialog | **ID:** UC10 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The dialog where user can add a new reminder. | |
| **Trigger:** The user taps on the “Add new Reminder” button in the toolbar.  **Type:** External. | |
| **Relationship:**  **+ Association:** Pick Time, Pick Date, Enter Reminder Name, Cancel, Confirm.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “Pick Time” button to open the time picker dialog to choose the time.  2. The user taps on the “PickDate” button to open the date picker dialog to choose the date.  3. The user writes the name of the reminder in the “Enter Reminder Name” text view.  4. The user taps the “Confirm” button to create a new reminder. | |
| **Alternative flow:**  1. If the user taps on the “Cancel” button to return to the reminder menu.  2. If the user taps on the “Confirm” button with the “ Enter Reminder Name ” empty, a new reminder is created with a empty content and the time and date set to the current time and date.  3. If the user taps on the android backpress button “<”, the dialog will be closed. | |

- Time picker:

A computer screen with text and images

Description automatically generated

- Date picker:

A computer screen with text and numbers

Description automatically generated

A computer screen shot of a program

Description automatically generated- The add new reminder dialog:  
A computer screen shot of a program

Description automatically generated

- The insert reminder method:

A screen shot of a computer program

Description automatically generated

### A diagram of a person with text Description automatically generated3.11. Use case 11 (Edit Reminder dialog):

|  |  |
| --- | --- |
| **Use case:** Edit Reminder dialog | **ID:** UC11 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The dialog where user can edit a reminder. | |
| **Trigger:** The user taps on the “Edit Reminder” button on one of the reminder item on the recycle view of the reminder menu.  **Type:** External. | |
| **Relationship:**  **+ Association:** Pick Time, Pick Date, Edit Reminder Name, Cancel, Confirm.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “Pick Time” button to open the time picker dialog to change the time.  2. The user taps on the “PickDate” button to open the date picker dialog to change the date.  3. The user edits the name of the reminder in the “Enter Reminder Name” text view.  4. The user taps the “Confirm” button to confirm the changes. | |
| **Alternative flow:**  1. If the user taps on the “Cancel” button to return to the reminder menu.  2. If the user taps on the android backpress button “<”, the dialog will be closed. | |

- The Edit Reminder dialog:

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

- Time picker dialog:

A computer screen with text and numbers

Description automatically generated

- The date picker dialog:

A computer screen with text and numbers

Description automatically generated

- The update reminder method:

A screen shot of a computer program

Description automatically generated

### A diagram of a person with blue circles Description automatically generated3.12. Use case 12 (Progress Tracker Menu):

|  |  |
| --- | --- |
| **Use case:** Progress Tracker Menu | **ID:** UC12 |
| **Main actor:** User | **Priority:** 1 |
| **Brief description:** The menu where user can task progress. | |
| **Trigger:**  The user taps the “Go To Checklist Menu” button on the navigation drawer or taps on the notification.  **Type:** External. | |
| **Relationship:**  **+ Association:** Open Navigation Drawer Show pie chart, Rate Task Delete All Progress.  **+ Include:** Open The Rate Task dialog.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The uesr taps on the “Rate Task” button.  2. <<User case 13>> | |
| **Alternative flow:** 1. If the user taps on the “Delete All Progress” button all data on the pie chart will be deleted.  2. The user can move to another acitivity by tapping on the “Open navigation Menu” button. <<Use case 1>>. | |

A screenshot of a computer

Description automatically generated- Setting up “Show pie chart”:

A screen shot of a computer program

Description automatically generated- Setting behaviour for the “Rate Task” and “Delete All Progress” buttons:

- Set pie chart method:

A screen shot of a computer program

Description automatically generated

- Update pie chart method:

A screen shot of a computer program

Description automatically generated

- Delete and update pie chart methods:

A computer screen shot of a program code

Description automatically generated

### A diagram of a user Description automatically generated3.13. Use case 13 (Rate Task dialog):

|  |  |
| --- | --- |
| **Use case:** Rate Task dialog | **ID:** UC13 |
| **Main actor:** User | **Priority:** 2 |
| **Brief description:** The dialog where user can rate a task. | |
| **Trigger:** The user taps on the “Rate Task” button on the progress tracker menu.  **Type:** External. | |
| **Relationship:**  **+ Association:** Rate Late, Rate Unfinished, Rate On Time.  **+ Include:** None.  **+ Extend:** None.  **+ Generalization:** None. | |
| **Normal flow:**  1. The user taps on the “Rate Late” button to increate the late portion of the pie chart. | |
| **Alternative flow:**  1. The user taps on the “Rate Unfinished” button to increate the late portion of the pie chart.  2. The user taps on the “Rate On Time” button to increate the late portion of the pie chart.  3. If the user taps on the android backpress button “<”, the dialog will be closed. | |

- The Rate Task dialog:

A computer screen shot of a program

Description automatically generated

- The get late task method:

A computer screen shot of a program code

Description automatically generated

- The get unfinished task method:

A screen shot of a computer

Description automatically generated

- The get on time task method:

A computer screen shot of a program code

Description automatically generated

## Kết quả đạt được:

In conclusion, the development of the simple note-taking application on the Android platform is a success. By incorporating essential features such as note creation, checklist management, reminder scheduling, and task progress tracking, we have created a simple but effective tool to aid users in their daily organization. Leveraging the efficiency of Kotlin and the reliability of SQLite, we ensured a smooth and stable user experience, contributing to the overall satisfaction with the app. Meeting our goals regarding product quality and functionality, our app facilitates efficient and convenient task management for users, empowering them to stay organized and productive.