

Midterm Project: Building a Simple Task Management App in Android Using Kotlin

Send to teams and google form before 27.10.2024 23:59. Write report in pdf. Report must be according to the template. Plagiarism of code or report leads to 0.

Link to google form

https://docs.google.com/forms/d/e/1FAIpQLSc7dGq5gnUqvBIQ3wW7E-TjnQmHQZI_uj_9g3OLofseSERUow/viewform?usp=sf_link

Project Overview

This project involves developing a simple task management application for Android using Kotlin. The app will allow users to create, view, update, and delete tasks, providing a user-friendly interface. The project will cover key concepts of Android development, Kotlin syntax, object-oriented programming, handling user input, and managing the activity and fragment lifecycle.

Key Technologies and Libraries Used

- Android Studio (IDE)
- Kotlin
- Android Jetpack (Lifecycle, ViewModel)
- RecyclerView (for displaying lists)

Project Requirements

1. **Overview of Android Development: Intro to Kotlin**
 - Set up the Android development environment using Android Studio.
 - Get familiar with Kotlin syntax and features, emphasizing how it improves Android development.
2. **Functions and Lambdas in Kotlin**
 - Write functions to encapsulate common functionality in the app.
 - Use lambdas for concise callbacks, especially in UI components.
3. **OOP in Kotlin**
 - Implement object-oriented programming principles using Kotlin classes and data classes to model tasks.
4. **Working with Collections in Kotlin**
 - Use collections (e.g., lists, maps) to manage tasks and demonstrate operations on these collections.
5. **Android Layout**
 - Design user interfaces using XML layout files.

- Utilize different layout types (LinearLayout, ConstraintLayout) to organize UI elements.
- 6. **Activity: Handling User Input and Events**
 - Create an Activity for displaying tasks and handling user interactions.
 - Implement user input handling (e.g., text input for adding tasks).
- 7. **Activity Lifecycle**
 - Understand and implement the activity lifecycle methods (onCreate, onStart, onResume, onPause, onStop, onDestroy).
- 8. **Fragments and Fragment Lifecycle**
 - Create fragments to modularize the UI and manage the task details.
 - Implement the fragment lifecycle and manage communication between fragments and activities.

Project Report Template

Project Report: Building a Simple Task Management App in Android Using Kotlin Times new roman 12 pt, single line spacing.
Minimum number of pages is 10.

Table of Contents

1. **Executive Summary**
2. **Introduction**
3. **Project Objectives**
4. **Overview of Android Development and Kotlin**
5. **Functions and Lambdas in Kotlin**
6. **Object-Oriented Programming in Kotlin**
7. **Working with Collections in Kotlin**
8. **Android Layout Design**
9. **Activity and User Input Handling**
10. **Activity Lifecycle Management**
11. **Fragments and Fragment Lifecycle**
12. **Conclusion**
13. **References**
14. **Appendices**

1. Executive Summary

Provide a brief overview of the project's goals, the technologies used, and the outcomes achieved.

2. Introduction

Introduce the importance of mobile applications and the role of Kotlin in modern Android development. Discuss the motivation behind creating a task management app.

3. Project Objectives

List the specific objectives of the project, such as developing a functional mobile application, understanding the Android lifecycle, and utilizing Kotlin features.

4. Overview of Android Development and Kotlin

- **Development Environment:** Detail the setup of Android Studio and the configuration for Kotlin development.
- **Kotlin Overview:** Discuss Kotlin features and how they enhance Android development compared to Java.

5. Functions and Lambdas in Kotlin

- **Function Implementation:** Provide examples of functions created for the app (e.g., adding a task, retrieving tasks).
- **Using Lambdas:** Illustrate the use of lambdas in handling UI events and callbacks.

6. Object-Oriented Programming in Kotlin

- **Classes and Data Classes:** Describe how classes and data classes are used to represent tasks, including properties and methods.
- **Encapsulation:** Discuss how OOP principles are applied to organize code.

7. Working with Collections in Kotlin

- **Collection Usage:** Detail how lists or maps are used to manage tasks.
- **Operations on Collections:** Provide examples of common operations (e.g., filtering, sorting) performed on the collections.

8. Android Layout Design

- **XML Layouts:** Explain the design of the app's user interface using XML layout files.
- **Layout Types:** Discuss the choice of layout types and their suitability for the app.

9. Activity and User Input Handling

- **Activity Creation:** Describe the main activity created to display tasks.

- **User Input Handling:** Detail how user inputs (e.g., task title, description) are collected and processed.

10. Activity Lifecycle Management

- **Lifecycle Methods:** Discuss the importance of lifecycle methods in managing the app's state.
- **Example Implementation:** Provide examples of overriding lifecycle methods in the activity.

11. Fragments and Fragment Lifecycle

- **Fragment Creation:** Describe how fragments are created and used within the app.
- **Fragment Lifecycle:** Explain the lifecycle of fragments and how they interact with the main activity.

12. Conclusion

Reflect on the project outcomes, the effectiveness of the technologies used, and suggestions for future enhancements or features.

13. References

List all resources, documentation, and articles referenced throughout the project.

14. Appendices

Include additional diagrams, code snippets, or data that support the report but are too detailed for the main sections.