TaskHub Mobile App Report

1. Introduction

TaskHub is a modern task management mobile application developed entirely in Kotlin. The app aims to streamline task organization and enhance user productivity by providing a comprehensive set of features and intuitive user interface. The primary goal of TaskHub is to empower users to efficiently manage their tasks, stay organized, and collaborate effectively.

2. Features and Functionality

TaskHub offers a range of features categorized into three tiers: Bronze, Silver, and Gold.

Bronze Category

- Task Creation and Organization:\* TaskHub allows users to easily create and organize tasks, enabling them to stay on top of their responsibilities. Users can input task details, set priorities, and categorize tasks for better management.

- Completion Marking:\* The app provides a simple mechanism for marking tasks as completed. Users can track their progress and visually identify completed tasks, helping them stay motivated and focused.

- Reminders:\* TaskHub sends timely reminders to users for upcoming deadlines. These reminders ensure that users stay informed about their pending tasks and can take necessary actions promptly.

Silver Category (70%-80%)

- Subtasks:\* TaskHub supports breaking down tasks into smaller, manageable subtasks. This feature enables users to divide complex tasks into more granular and actionable steps, improving overall workflow and productivity.

- Progress Tracking:\* The app offers progress tracking capabilities, allowing users to monitor the completion status of their tasks. Users can easily view the percentage of completed tasks and identify areas that require attention.

- Timelines with Start and End Dates:\* TaskHub allows users to set timelines for their tasks by specifying start and end dates. This feature helps users establish clear deadlines and milestones, ensuring better time management and task prioritization.

Gold Category (80%-100%)

- Ability to Attach Resources:\* TaskHub provides the ability to attach relevant documents, files, or notes directly to tasks. This feature streamlines organization and collaboration by keeping all task-related resources in one place, easily accessible to users.

3. Technical Implementation

Android Manifest (AndroidManifest.xml)

The Android Manifest file (AndroidManifest.xml) contains the application configuration and declares the main activity of the app. It specifies the application name, icon, theme, and other essential settings. The manifest file also defines the necessary permissions and attributes required by the app.

Gradle Build Configuration (build.gradle.kts)

The Gradle build configuration file (build.gradle.kts) manages the app's build process and dependencies. It specifies the Android SDK version, compile options, and other build-related settings. The file also includes the necessary plugin dependencies, such as Kotlin Android Extensions, Dagger Hilt, and Compose Destinations.

MainActivity (MainActivity.kt)

The MainActivity class serves as the entry point of the app. It sets up the main activity and initializes the Compose UI. The setContent block defines the root composable of the app, which is the ParentNav composable. The ParentNav composable handles the bottom navigation and manages the different screens of the app.

4. UI and Design

Compose UI Framework

TaskHub utilizes the Jetpack Compose UI framework for building its user interface. Compose provides a declarative and efficient way to create interactive and responsive UI components. The app leverages Compose's Material Design components and follows the Material Design guidelines for a consistent and visually appealing user experience.

Navigation

TaskHub implements bottom navigation using the ParentNav composable. This composable handles the navigation between different screens of the app, such as the task list, calendar, and settings screens. The navigation is seamlessly integrated with the Compose Destinations library, which simplifies the navigation setup and provides type-safe navigation.

UI Components and Libraries

TaskHub incorporates various UI components and libraries to enhance the user experience:

- Compose Destinations:\* Used for navigation between different screens of the app.

- Feather Icons Pack:\* Provides a collection of high-quality icons for visual elements.

- Compose Calendar:\* Allows users to select dates and view tasks in a calendar format.

- Compose Date Picker Dialog:\* Enables users to input dates using a user-friendly dialog.

- Compose Number Selector Dialog:\* Allows users to input numbers using a customizable dialog.

- Custom Toast Library:\* Displays informative and visually appealing toast messages to users.

5. Data Management

Room Database

TaskHub utilizes the Room persistence library for local data storage. Room provides an abstraction layer over SQLite, making it easier to work with databases in Android. The app defines entity classes that represent the data models and creates a database configuration using Room annotations. This allows for efficient storage and retrieval of task-related data.

Data Serialization

The app uses the GSON library for JSON serialization and deserialization. GSON simplifies the process of converting Java objects to JSON and vice versa. This is particularly useful when storing complex data structures or communicating with web services.

6. Dependency Injection

Dagger Hilt

TaskHub integrates Dagger Hilt for dependency injection. Hilt is a dependency injection library built on top of Dagger, specifically designed for Android apps. It simplifies the setup and management of dependencies by providing annotations and code generation. The app uses Hilt annotations to inject dependencies into various components, such as view models and repositories, promoting loose coupling and testability.

7. Testing

Unit Tests

TaskHub includes unit tests written using the JUnit testing framework. These tests verify the correctness of individual components and functions in isolation. The presence of unit tests ensures the reliability and maintainability of the codebase.

Android Instrumentation Tests

The app also incorporates Android instrumentation tests using the Espresso testing framework. Espresso allows for UI testing by simulating user interactions and assertions on the app's UI components. These tests validate the app's functionality and user experience, ensuring a high-quality and bug-free application.

8. Conclusion

TaskHub is a feature-rich task management mobile app developed using Kotlin. It provides users with a comprehensive set of tools to organize, track, and collaborate on tasks effectively. The app leverages modern Android development practices and libraries, such as Jetpack Compose, Room, Dagger Hilt, and various UI components, to deliver a smooth and intuitive user experience.

The app's architecture follows best practices, with a clear separation of concerns and modular design. The use of dependency injection and testing frameworks ensures maintainability and reliability of the codebase.

TaskHub has the potential to significantly enhance user productivity and streamline task management processes. Future enhancements and roadmap items could include additional features such as task sharing, collaboration, and integration with popular productivity tools.

This report provides a comprehensive overview of the TaskHub mobile app, covering its features, technical implementation, UI design, data management, dependency injection, testing, and future considerations. It highlights the app's potential to revolutionize task management and boost user productivity.