

****Introduction:****

The purpose of this report is to outline the requirements analysis, feasibility analysis, and Software Requirements Specification (SRS) for the development of a Java compiler app. The app aims to allow users to write and compile Java code directly on their mobile devices. The report will also cover the key features of the app.

****Requirements Analysis:****

To determine the requirements for the Java compiler app, the following aspects need to be considered:

1. **Functional Requirements:**

- Provide a code editor with syntax highlighting and basic text manipulation features.
- Enable users to enter and edit Java code.
- Support compilation of Java code using the Java Compiler API.
- Display compilation output, including error messages and success status.
- Allow users to create new Java files and delete existing files.
- Handle file-related operations securely within the app's private storage.

2. **Non-Functional Requirements:**

- Ensure the app is user-friendly with an intuitive and attractive user interface.
- Optimize app performance to provide a smooth and responsive experience.
- Ensure compatibility with Android devices running a specific version of Android OS.
- Adhere to Android security guidelines to protect user data and device security.
- Provide error handling and informative error messages for runtime errors and exceptions.

****Feasibility Analysis:****

A feasibility analysis helps assess the viability and practicality of developing the Java compiler app. The following areas should be evaluated:

1. **Technical Feasibility:**

- The Java Compiler API provided by the Android SDK allows for programmatic compilation of Java code.
- The availability of Android development tools, frameworks, and libraries supports the implementation of the app's features.
- The app's required functionalities align with the capabilities of modern Android devices.

2. **Operational Feasibility:**

- The Java compiler app caters to developers and students who require a mobile solution for quick code compilation.
- The app enhances portability and convenience, allowing users to compile Java code on the go without relying on a computer.

3. **Economic Feasibility:**

- The development of the Java compiler app should be financially viable based on the expected target market and potential revenue streams, such as in-app purchases or ads.

Software Requirements Specification (SRS):

The SRS outlines the detailed specifications and features of the Java compiler app. Here's a simplified version of the SRS:

1. **Introduction:**

- Provide an overview of the Java compiler app and its purpose.

2. **Functional Requirements:**

- Detail the functional requirements identified during the requirements analysis phase.

3. **Non-Functional Requirements:**

- Specify the non-functional requirements, including performance, usability, security, and compatibility aspects.

4. **System Architecture:**

- Describe the high-level system architecture of the app, including the major components and their interactions.

5. ****User Interface Design:****

- Present wireframes, mockups, or visual representations of the app's user interface design.

6. ****Data Management:****

- Explain how the app will handle file operations, such as creating and deleting Java files.

7. ****Error Handling and Reporting:****

- Describe how the app will handle errors, exceptions, and provide meaningful error messages to users.

8. ****Testing Requirements:****

- Define the testing approach and requirements for ensuring the app's functionality, performance, and reliability.

9. ****Deployment and Distribution:****

- Outline the steps and considerations for deploying the Java compiler app, including APK creation and distribution channels.