## 

**Note-taking application**

NoteR

New Uzbekistan University

Software Engineering (JSE5, “DRAM”)

Diyorbek S., Rahmatulloh B., Aydos K., Mukhammadjon G.

2.01.2024

**Functional and non-functional requirements:**

* + Functional Requirements:
    - The application shall allow users to create a new note.
    - The application shall allow users to edit an existing note.
    - The application shall allow users to delete a note.
  + Non-Functional Requirements:
    - The application shall be responsive and provide a consistent user experience across different devices.
    - The application shall load notes quickly, with a maximum loading time of 2 seconds.

**Requirements:**

* + Create New Note
  + Edit Existing Note
  + Delete Note
  + Responsive Design
  + Fast Note Loading

**Requirement descriptions:**

* + The application should provide a screen with input fields for the user to enter a new note's title and content. Upon submission, a new note is created.
  + Users should be able to access an edit screen for an existing note, where they can modify the title and content.
  + The application must confirm with the user before deleting a note.

**UML Activity Diagram**

**The diagram shows activities for the whole application:**

**Mobile Application Development Lifecycle**

* + **Planning:**
    - Define project scope and objectives.
    - Identify features and functionalities.
  + **Design:**
    - Create wireframes and UI design.
    - Define data models for notes.
  + **Implementation:**
    - Develop note creation, editing, and deletion functionalities.
    - Implement UI screens and navigation.
  + **Testing:**
    - Conduct unit tests for note functionalities.
    - Perform integration tests for UI components.
  + **Deployment:**
    - Release the application on app stores.

**Planning:**

* + **What to Produce:** Detailed project plan outlining tasks, milestones, and timelines.
  + **Justification:** According to [Source], effective planning is crucial for project success. It helps identify potential risks, allocate resources efficiently, and set clear expectations.

**Analysis:**

* + **What to Produce:** Requirements document detailing both functional and non-functional requirements.
  + **Justification:** As per [Source], a comprehensive requirements analysis ensures that the development team understands the project scope, leading to more accurate estimates and reduced rework.

**Design:**

* + **What to Produce:** UI/UX design prototypes, database schema, and system architecture.
  + **Justification:** Referring to [Source], a well-designed architecture enhances system performance, and a user-friendly UI improves the overall user experience.

**Implementation:**

* + **What to Produce:** Actual code, unit tests, and code documentation.
  + **Justification:** [Source] emphasizes that thorough implementation, including testing and documentation, is essential for maintainability and future scalability.

**Testing:**

* + **What to Produce:** Test cases, black-box testing results, and user acceptance testing documentation.
  + **Justification:** According to [Source], rigorous testing ensures the reliability and functionality of the application, reducing the likelihood of post-release issues.

**Deployment:**

* + **What to Produce:** Deployed application on app stores, release notes.
  + **Justification:** [Source] highlights the significance of a smooth deployment process for delivering a positive user experience and garnering positive reviews.

**Maintenance:**

* + **What to Produce:** Bug fixes, updates, and user support channels.
  + **Justification:** [Source] suggests that proactive maintenance is crucial for addressing issues promptly, ensuring user satisfaction, and keeping the application up-to-date.

**Application Portability**

* + Mobile application portability refers to the ability of an application to run seamlessly across various platforms and devices, maintaining consistent functionality and user experience.
  + To achieve portability, the application will utilize responsive design principles, support libraries, and adapt to different screen layouts on Android and iOS. For example, the use of Flutter framework enables a single codebase for both platforms.

**Some specific solutions are present using external sources:**

**Responsive UI using Flutter:**

* + **Solution:** Utilize Flutter's responsive design principles to create a UI that adapts to various screen sizes and orientations.
  + **Reference:** Flutter: Building Adaptive UIs

**Support Libraries for Android:**

* + **Solution:** Implement Android Support Libraries to ensure compatibility with a wide range of Android devices and versions.
  + **Reference:** AndroidX Overview

**iOS Layouts for Different Devices:**

* + **Solution:** Use iOS Auto Layout to create flexible layouts that work well on different iOS devices.
  + **Reference:** Auto Layout Guide

**Cross-Platform Testing Framework:**

* + **Solution:** Employ a cross-platform testing framework like Appium or Detox to ensure consistent behavior across multiple devices and platforms.
  + **Reference:** [Appium: Cross-platform Mobile Application Automation](http://appium.io/)

**Minimize Device-Specific Dependencies:**

* + **Solution:** Reduce dependencies on device-specific features and hardware to enhance portability.
  + **Reference:** Building Portable Mobile Apps

**System and Hardware Requirements**

**All the system requirements to run the mobile application are mentioned:**

* + System Requirements:
    - Android: Min SDK Version 21
    - Min iOS Version: 12.0
  + Hardware Requirements:
    - Internet connectivity
    - 2GB RAM

**Black Box Testing**

* **Black Box Testing Table:**

| **Test Description** | **Expected Results** | **Actual Results** | **Comments** |
| --- | --- | --- | --- |
| Create New Note Test | New note should be successfully created | New note created successfully | The creation functionality is working as expected. |
| Edit Existing Note Test | Changes to the note should be saved | Changes saved successfully | Editing an existing note works correctly. |
| Delete Note Test | Note should be removed from the list | Note deleted successfully | The delete functionality is functioning as intended. |
| Responsive Design Test | UI elements should adapt to screen size | UI elements adapt as expected | The application UI is responsive on different devices. |
| Note Loading Time Test | Notes should load within 2 seconds | Notes loaded within the expected time | The application meets the specified loading time. |