Software Requirements Specification (SRS)

for Digital Doodle

Version: 1.0 **Date:** March 20, 2025

Prepared by: Abd Al Mohsin Siraj (2254901009)

Index

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for the **Digital Doodle** mobile application. This application allows users to create digital sketches on their mobile devices (Android) with advanced tools such as infinite zooming, layer management (drafts), and state history (undo/redo). This SRS details the functional and non-functional requirements for developers, stakeholders, and quality assurance testers.

1.2 Scope

Digital Doodle is a standalone mobile application developed using the **Flutter** framework.

- **Core Functionality:** Users can draw freehand strokes on a digital canvas.
- **State Management:** The app maintains a history stack to allow users to undo and redo actions.
- **Persistence:** Users can save unfinished drawings as "Drafts" to local storage and resume them later.
- **Export:** Completed artwork can be exported as high-resolution PNG images to the device's gallery.
- **Navigation:** The canvas supports an "infinite" workspace via pan and zoom gestures.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS:** Software Requirements Specification
- **CRUD:** Create, Read, Update, Delete
- **MVP:** Minimum Viable Product
- **UI/UX:** User Interface / User Experience
- **JSON:** JavaScript Object Notation (used for data serialization)
- **Stroke:** A continuous set of points drawn by the user from finger-down to finger-up.

2. Overall Description

2.1 Product Perspective

Digital Doodle is a self-contained product. It relies on the mobile device's native file system for storing drafts and the native media gallery for exporting final images. It does not require a backend server or internet connection to function.

2.2 Product Functions

The major functions of the system are:

1. **Freehand Drawing:** Rendering vector-based strokes with customizable properties (color, size, opacity).
2. **Canvas Manipulation:** Panning and zooming to view different areas of the workspace.
3. **Session Management:** Saving the current state of the canvas to a local database (Shared Preferences).
4. **History Control:** Traversing backward and forward through the drawing steps.
5. **Media Export:** Converting the canvas view into a static image file.

2.3 User Characteristics

- **Target Audience:** Students, hobbyists, and casual artists who need a quick and lightweight sketching tool.
- **Technical Proficiency:** Low to Medium. The interface is designed to be intuitive, requiring no prior knowledge of digital art software.

2.4 Constraints

- **Hardware:** Must run on smartphones with touch capability.
- **OS:** Compatible with Android 10 and above.
- **Storage:** Minimum 16 GB internal storage with 2 GB of RAM to run it properly.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 Drawing Engine

- **FR-01:** The system shall allow the user to draw continuous lines on the canvas using touch input.
- **FR-02:** The system shall provide a tool to customize the brush **Color** via a color picker.
- **FR-03:** The system shall provide a tool to customize the brush **Size**.
- **FR-04:** The system shall provide a tool to customize the brush **Opacity**.
- **FR-05:** The system shall allow the user to change the **Canvas Background Color**.

3.1.2 Navigation (Zoom & Pan)

- **FR-06:** The system shall provide a specific "Pan Mode" distinct from "Drawing Mode."

- **FR-07:** In Pan Mode, the system shall support multi-touch gestures to **Zoom In** and **Zoom Out** .
- **FR-08:** In Pan Mode, the system shall support dragging to **Pan** the visible area of the canvas.

3.1.3 History Management (Undo/Redo)

- **FR-09:** The system shall maintain a stack of executed strokes.
- **FR-10:** The system shall provide an **Undo** button that removes the most recent stroke from the canvas.
- **FR-11:** The system shall provide a **Redo** button that restores the most recently undone stroke.
- **FR-12:** The Redo history shall be cleared if a new stroke is drawn after an Undo action.

3.1.4 Data Persistence (Drafts)

- **FR-13:** The system shall allow the user to **save** the current drawing as a draft locally.
- **FR-14:** The system shall persist draft data including all strokes, colors, and canvas settings using JSON serialization.
- **FR-15:** The system shall provide a **Drafts List** screen displaying saved files with their last modification timestamp.
- **FR-16:** The system shall allow the user to **Load** a draft, restoring the canvas to its saved state.
- **FR-17:** The system shall allow the user to **Delete** a saved draft permanently.

3.1.5 Export

- **FR-18:** The system shall request write permission to the device's external storage/gallery.
- **FR-19:** The system shall export the current canvas view as a **PNG image** file to the device's main photo gallery.

3.2 Non-Functional Requirements

3.2.1 Performance

- **NFR-01:** The drawing latency (time between touch and line appearance) shall be less than 50ms.
- **NFR-02:** The application shall maintain a frame rate of at least 60 FPS during drawing and zooming operations.

3.2.2 Reliability

- **NFR-03:** The application shall not crash if the user attempts to load a corrupted draft file; it should handle the error gracefully.
- **NFR-04:** The application shall retain the list of saved drafts even after the app is closed and restarted.

3.2.3 Usability

- **NFR-05:** The UI shall visually distinguish between "Drawing Mode" and "Pan Mode" to prevent user errors.
- **NFR-06:** All icons shall follow standard Material Design guidelines for familiarity.