**MindFlow  
Software Requirements Specification (SRS)**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to define the functional and non-functional requirements for a macOS-based diagram software that allows users to create flowcharts, app flows, ER diagrams, and mind maps.

**1.2 Scope**

This software provides an infinite 2D canvas for creating structured diagrams. It supports adding main topics, subtopics, and relationships between them using intuitive keyboard and mouse interactions. The application enables users to zoom, pan, edit, and export diagrams efficiently.

**1.3 Intended Audience**

* Software developers
* UI/UX designers
* Business analysts
* Educators & students

**1.4 References**

* macOS AppKit & SwiftUI Documentation
* UX/UI Best Practices for Diagramming Tools

**2. Overall Description**

**2.1 Product Features**

**2.1.1 Infinite 2D Canvas**

* Users can scroll infinitely in all directions.
* Zoom in and out with pinch gestures or keyboard shortcuts.
* Pan across the canvas using mouse drag.

**2.1.2 Topic & Subtopic System**

* Press **Return** to add a new **main topic**.
* Press **Tab** to add a **subtopic** under the selected topic.
* Topics and subtopics appear as rectangles.
* Default names: "Main Topic 1", "Main Topic N", "Subtopic 1", "Subtopic N".

**2.1.3 Relationship & Connections**

* **Control + Drag** from one topic to another to create a relationship.
* Automatic curved connectors between elements.
* Support for different line styles (dotted, solid, arrows).

**2.1.4 Editing & Customization**

* **Press Spacebar** to edit text inside a topic.
* Change shape colors, borders, and font styles.
* Resize and move topics/subtopics while maintaining connections.

**2.1.5 Smart Grid & Auto-Layout**

* Auto-align elements in a structured manner.
* Snap to grid functionality.
* Auto-arrange subtopics under main topics.

**2.1.6 Export & Import**

* Save diagrams in JSON format for future editing.
* Export as **PNG, SVG, or PDF**.

**2.1.7 Undo/Redo Support**

* Maintain a history of changes.
* **Cmd + Z** to undo, **Cmd + Shift + Z** to redo.

**2.2 User Characteristics**

* Users should have basic familiarity with macOS UI.
* No programming knowledge is required.

**2.3 Constraints**

* Runs only on **macOS**.
* No cloud collaboration in the initial version.
* Performance optimized for up to **1000 elements per diagram**.

**3. Functional Requirements**

|  |  |
| --- | --- |
| **ID** | **Requirement Description** |
| FR-1 | The system shall provide an infinite scrollable and zoomable 2D canvas. |
| FR-2 | The system shall allow users to add main topics by pressing the **Return** key. |
| FR-3 | The system shall allow users to add subtopics by pressing the **Tab** key. |
| FR-4 | The system shall allow users to create relationships by **Control + Dragging** between topics. |
| FR-5 | The system shall allow users to edit text inside a shape by pressing the **Spacebar**. |
| FR-6 | The system shall allow users to export diagrams in PNG, SVG, and PDF formats. |
| FR-7 | The system shall provide undo and redo functionality. |
| FR-8 | The system shall auto-arrange elements in a structured layout. |

**4. Non-Functional Requirements**

|  |  |
| --- | --- |
| ID | Requirement Description |
| NFR-1 | The software shall provide a **smooth and responsive** user experience. |
| NFR-2 | The system shall handle up to **1000 elements** in a diagram efficiently. |
| NFR-3 | The UI shall be intuitive and follow macOS **HIG (Human Interface Guidelines)**. |
| NFR-4 | The system shall persist data locally for offline use. |

**5. System Models**

**5.1 Use Case Diagram**

* User opens the app → Creates a new diagram → Adds topics & subtopics → Connects them → Edits text → Saves or Exports the file.

**5.2 Data Model**

* **Topic (id, name, position, color, parentId)**
* **Subtopic (id, name, position, parentId)**
* **Relationship (sourceId, targetId, type)**

**6. Recommended Tech Stack**

**6.1 Language & UI Framework**

* **Swift + SwiftUI** for UI components.
* **AppKit (NSView)** for custom rendering of the infinite canvas.

**6.2 Graphics & Rendering**

* **Core Graphics (CGContext)** for custom shape drawing.
* **Core Animation (CAShapeLayer)** for smooth interactions.
* **Metal (Optional)** for GPU-accelerated rendering if needed.

**6.3 Infinite Scrolling & Zooming**

* **NSScrollView + NSClipView** for infinite scrolling.
* **Magnification Gesture** for pinch-to-zoom.
* **Pan Gesture (NSPanGestureRecognizer)** for smooth panning.

**6.4 Data Handling**

* **Tree Data Structure** to manage topics and subtopics.
* **Codable (JSON)** for saving and loading diagrams.
* **Core Data (Optional)** for persistent storage.

**6.5 Keyboard & Mouse Interactions**

* **NSResponder** for keyboard shortcuts (Return, Tab, Space, Control + Drag).
* **Mouse Events (NSGestureRecognizer)** for dragging, selecting, and creating relationships.

**7. Future Enhancements**

* Cloud-based real-time collaboration.
* AI-powered auto-layout and diagram suggestions.
* Integration with **iPad for Apple Pencil support**.

**8. Appendix**

* Shortcuts: **Return (Add Topic), Tab (Add Subtopic), Control + Drag (Create Relationship), Space (Edit Text)**.
* Supported Formats: JSON, PNG, SVG, PDF.