LAYOUTISH

Development Roadmap & Feature Specification

Version 1.0 | Created: January 30, 2026 | Appish

Executive Summary

Layoutish is a macOS menu bar application that enables users to save and restore window layouts across single or multiple monitors. Unlike competitors that focus on window snapping, Layoutish specializes in **complete workspace restoration**—remembering which apps were open, their exact positions, and restoring everything with a single click or hotkey.

Key Differentiators

- **App Launching:** Opens apps that aren't running when restoring layouts
- **Smart Monitor Handling:** Graceful fallback when displays change (dock/undock)
- **Simpler than Moom:** Focused UX without overwhelming complexity
- **Visual Layout Editor:** See your layout before applying (planned v1.2)
- **Consistent Appish Design:** Matches Lockish/Soundish family aesthetic

Market Positioning

Competitor	Price	Layout Saving	Our Advantage
Moom	$10	Excellent	Simpler UX, better multi-monitor
Rectangle Pro	$10	Good	App launching, visual editor
Magnet	$8	None	We have layouts (their gap)
BetterSnapTool	$3	Basic	More polished, auto-restore
Layoutish	**$12.99**	**Excellent**	—

Version 1.0 — MVP Release

Target: Q1 2026

Layout Saving & Restoration

- **Save current layout**: Capture all window positions, sizes, and open apps [Essential]
- **Name layouts**: User-friendly names (Coding, Design, Meetings) [Essential]
- **Restore layout**: Re-open apps and position windows exactly as saved [Essential]
- **Per-display configuration**: Save which monitor each window belongs to [Essential]
- **Menu bar quick access**: Click layout name to instantly apply [Essential]

Layout Management

- **Unlimited layouts**: No artificial cap on saved presets [Essential]
- **Edit layout name**: Rename existing layouts [Essential]
- **Delete layout**: Remove unwanted presets [Essential]
- **Update layout**: Re-save current positions to existing layout [Essential]

Hotkey Activation

- **Global hotkeys**: Assign keyboard shortcuts to each layout [Essential]
- **Quick switcher**: ■+Shift+L opens layout picker overlay [High]
- **Customizable shortcuts**: User sets their own key combos [High]

Infrastructure

- **Accessibility permission flow**: Onboarding with clear instructions (Lockish-style) [Essential]
- **License validation**: Lemon Squeezy integration [Essential]
- **Launch at login**: SMAppService integration [High]
- **Sparkle updates**: Auto-update framework [High]

Version 1.1 — Enhanced Experience

Target: Q2 2026

Enhanced Features

- **Layout thumbnails**: Small preview showing window arrangement [High]
- **iCloud sync**: Sync layouts across multiple Macs [High]
- **Monitor profile detection**: Auto-detect when displays change [High]
- **Import/Export**: Share layouts as JSON files [Medium]

Quality of Life

- **Layout search**: Quick search in switcher overlay [Medium]
- **Recent layouts**: Show recently used at top [Medium]
- **Duplicate layout**: Clone existing layout as starting point [Medium]

Version 1.2 — Power User Features

Target: Q3 2026

Advanced Features

- **Visual layout editor**: Drag-and-drop preview of window positions [High]
- **Auto-restore on app launch**: Remember positions per-app automatically [High]
- **Workspace contexts**: Group layouts by project/mode with colored icons [Medium]
- **Layout triggers**: Auto-apply when specific monitors connected [Medium]

Multi-Window Handling

- **Per-window identification**: Track multiple windows from same app [High]
- **Window title matching**: Position specific windows by title [Medium]
- **Tab group support**: Safari/Chrome tab group awareness [Medium]

Version 2.0 — Intelligence & Community

Target: Q4 2026

Smart Features

- **AI layout suggestions**: Learn user patterns and suggest layouts [Medium]
- **Time-based layouts**: Auto-switch based on time of day [Medium]
- **App-aware restoration**: Smart handling of app-specific quirks [Medium]

Community

- **Layout library**: Browse and import community layouts [Low]
- **Layout templates**: Pre-built layouts for common workflows [Low]

Technical Architecture

Required Permissions

Permission	Required	Purpose
Accessibility	Yes	Window manipulation via AXUIElement APIs
Screen Recording	Optional	Layout thumbnails/previews

Key macOS APIs

- **CGWindowListCopyWindowInfo** — Get all windows on screen
- **AXUIElementSetAttributeValue** — Set window position/size
- **NSWorkspace.shared.runningApplications** — Check running apps
- **NSWorkspace.shared.open** — Launch applications
- **NSScreen.screens** — Monitor detection and configuration

Data Model (JSON)

```
{
  "layouts": [{
    "id": "uuid",
    "name": "Coding",
    "icon": "laptop.computer",
    "hotkey": "■■1",
```

```
    "monitorConfig": "MacBook Pro + Dell U2720Q",
    "windows": [{
      "appBundleId": "com.apple.Safari",
      "frame": {"x": 0, "y": 0, "width": 960, "height": 1080},
      "displayId": 1
    }]
  }]
}
```

UI Design Guidelines

Brand Colors (Matching Lockish)

Color	Hex	RGB	Usage
Brand Purple	#9A8AE6	(0.6, 0.55, 0.9)	Primary accent, buttons
Brand Purple Dark	#404066	(0.25, 0.25, 0.4)	Button backgrounds
Success Green	#4DC760	(0.3, 0.75, 0.4)	Active/success states

Menu Bar Dropdown Structure

The menu bar dropdown follows Lockish's design patterns:

• **Header:** App name + status indicator (● Active)
• **Warning Banners:** Permission/license warnings (red/orange background)
• **Main Content:** Scrollable list of saved layouts with expand/collapse
• **Action Row:** "+ Save Current Layout" button (brand purple)
• **Footer:** Settings gear (left) + Quit text (right)

Layout Card Component

• 35x35 icon (SF Symbol or emoji) with rounded corners
• Layout name (bold, 10pt) with hotkey badge
• Expand chevron to reveal: edit name, update positions, delete, hotkey picker
• Apply button with brand purple styling
• Hover state with subtle background highlight