

Buriza Wallet: UI/UX Design and Front-End Development

Clark Alesna¹ Stan Kiefer Gallego² Caitlin Lindsay³

2025-09-15

clark@saib.dev
stan.gallego@saib.dev
caitlin.lindsay@saib.dev
SAIB Inc.

Table of Contents

I. Introduction	3
II. Buriza Front-End	5
ii.1 Design	5
ii.1.1 Material Design 3	6
ii.2 Development	8
ii.2.1 Project Architecture	8
ii.2.2 Platform Implementations	8
ii.2.3 Front-End Implementation	9
III. Implementation	11
iii.1 Mobile	11
iii.2 Browser	11
iii.3 Desktop	11
IV. User Interface	13
V. Conclusion	14
VI. References	15
VII. Links	16
VIII. References	17

I. Introduction

Buriza is a comprehensive Cardano wallet software suite spanning mobile, desktop, and browser platforms. The name Buriza is inspired by the Japanese interpretation of “blizzard,” symbolizing power, resilience, and control. Reflecting this spirit, Buriza empowers users with full sovereignty over their wallet infrastructure through open-source development and complete decentralization.

Funded under Project Catalyst Fund 13, the initiative is structured across three complementary proposals:

- [Buriza.Mobile](#) — focused on delivering a flexible and responsive mobile application,
- [Buriza.Browser](#) — the browser extension interface, and
- [Buriza.Desktop](#) — a desktop implementation featuring full-node capabilities.

Since Milestone 1, Buriza’s design has undergone significant evolution. The team adopted Material Design 3 (MD3) principles to create a more refined, accessible, and cohesive user experience. A major visual shift includes the introduction of Buriza Frostling, a bold, yeti-inspired mascot that replaces the earlier soft, gem-like designs. Frostling embodies Buriza’s core values—security, trust, and power—and appears throughout the interface to reinforce brand consistency.

The updated design also features enhanced iconography, a standardized color system based on Buriza’s three original core colors, and new accent shades that provide stronger visual contrast. Clean mockups and improved UI patterns further highlight the wallet’s sleek, modern aesthetic.

On the technical side, Buriza is built on a modular .NET 9 architecture, emphasizing shared components and platform independence. At the core of the system is the Buriza.UI shared component library, enabling a single codebase to serve multiple targets—browser extension, progressive web app (PWA), and cross-platform MAUI applications. This approach reduces redundant development and ensures a consistent user experience across all platforms.

The front-end implementation utilizes:

- Blazor for application logic and rendering,
- MudBlazor for MD3-compliant UI components, and
- Tailwind CSS for utility-first styling.

The browser extension is powered by Blazor WebAssembly and follows Manifest V3 standards. The PWA achieves near-native performance through WebAssembly, while the MAUI app leverages BlazorWebView to combine native device capabilities with a unified web-based UI.

With these design and architectural foundations in place, this document outlines the progress and deliverables completed for Milestone 2. It details the evolution of the user interface and system architecture since Milestone 1, explains the project setup and tools used, and documents the front-end implementation methodology. Screenshots, videos, and working examples are provided to demonstrate Buriza’s evolving functionality. The report also highlights the challenges encountered and the solutions developed by the SAIB Inc. team, reflecting its ongoing commitment to building a secure, performant, and user-centered Cardano wallet.

This report has been converted to Word format for Project Catalyst submission. The original version may be found in the Buriza project's GitHub repository: <https://github.com/SAIB-Inc/Buriza/tree/main/src/Buriza.Docs>

II. Buriza Front-End

This section covers the branding and UI/UX design of the Buriza wallet, describing the updates made since the first submission. Additionally, it tackles the implementation of the design - detailing the project set up and various tools utilized.

ii.1 Design

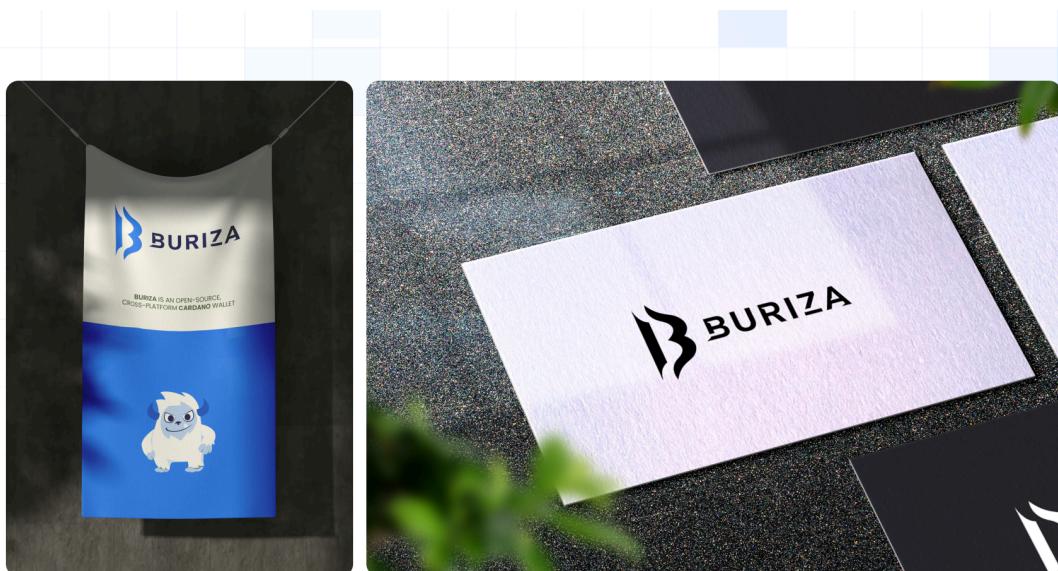
This section outlines Buriza's design evolution in alignment with the principles and best practices of Google's Material Design 3.

Buriza's design has significantly evolved since the submission of Project Catalyst's Milestone 1, with the initial branding and UI/UX designs detailed in the following [report](#).

A key transformation in Buriza's branding is the introduction of its new mascot, Buriza Frostling. Inspired by the yeti, Frostling's energetic expressions and bold, resilient personality stand in contrast to the earlier delicate, pastel mascots. This strong character is present throughout the application's design, reinforcing Buriza's core values of security, trust, and power.

The updated branding report also features enhanced iconography, an expanded color palette, and clean mockups that showcase Buriza's modern and sleek aesthetic. These comprehensive changes can be viewed in the revised [branding report](#), which includes detailed descriptions and illustrations of the logo and typography.

Figure 1 – Buriza Mockup: featuring the mascot, colors, and logo



Title:
Logo

Brand
Guidelines

Produced by
SAIB

Page
09

These developments are reflected in the overall UI and UX of the wallet, resulting in a more refined and visually cohesive application. The improvements stem from the opportunity the SAIB Inc. design team had to explore Material Design 3, using its guidelines to further elevate the project's look and feel. These refinements can be seen in detail in the following [Figma file](#).

ii.1.1 Material Design 3

Material Design is a design system created by Google designers and developers that provides UX foundations, guidance, and UI components across Android, Flutter, and the Web. Its most recent update, Material Design 3, highlights emotional impact. Placing great significance on expressiveness, dynamism, and accessibility, Material Design 3 expands the system with additional shapes, typography guidelines, intuitive motions, and more ([Material Design 3](#), 2025).

SAIB's design team studied the foundations of this update, noting and implementing changes based on Material Design 3's guidelines on color, elevation, icons, shape, motion, and typography. The impact of this study can be seen in the most recent design. Buriza's colors have been evaluated for contrast, and the initial color scheme has been updated to a more vibrant palette with accents and a greater variety of shades established for uniformity. Elevation is utilized to provide visual hierarchy and interaction guides for users, and labels have been added to the primary menu icons. Similar considerations have been applied to the wallet's shapes, with Material Design 3 button group designs added to many main pages and expressive animations applied in the main header. While Buriza's initial typography has not changed, revisions have been made to consider type roles within pages.

Figure 2 – Buriza Mobile UI - Home: design evolution from first to third iteration (left to right)

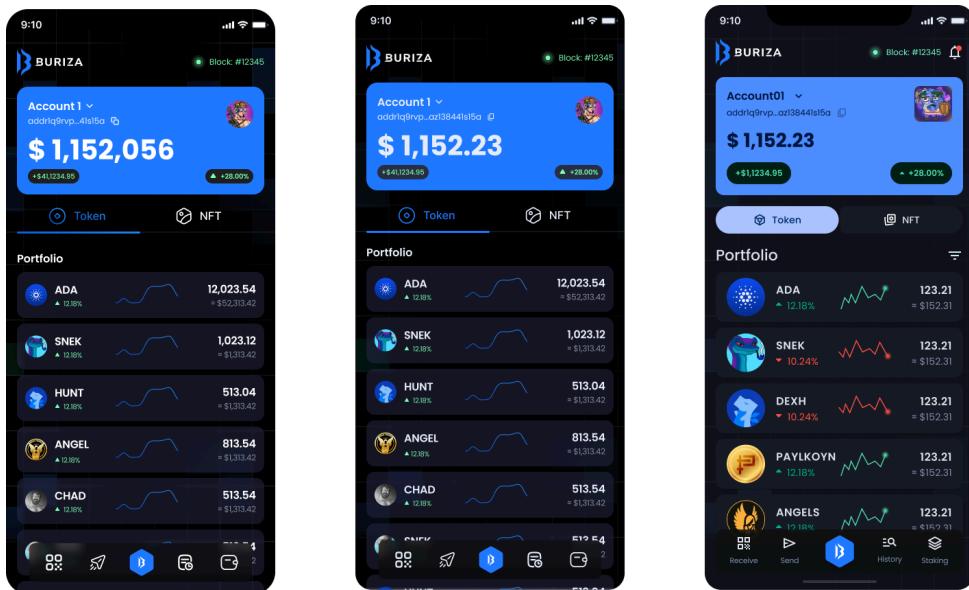


Figure 3 – Buriza Mobile UI - History: design evolution from first to third iteration (left to right)

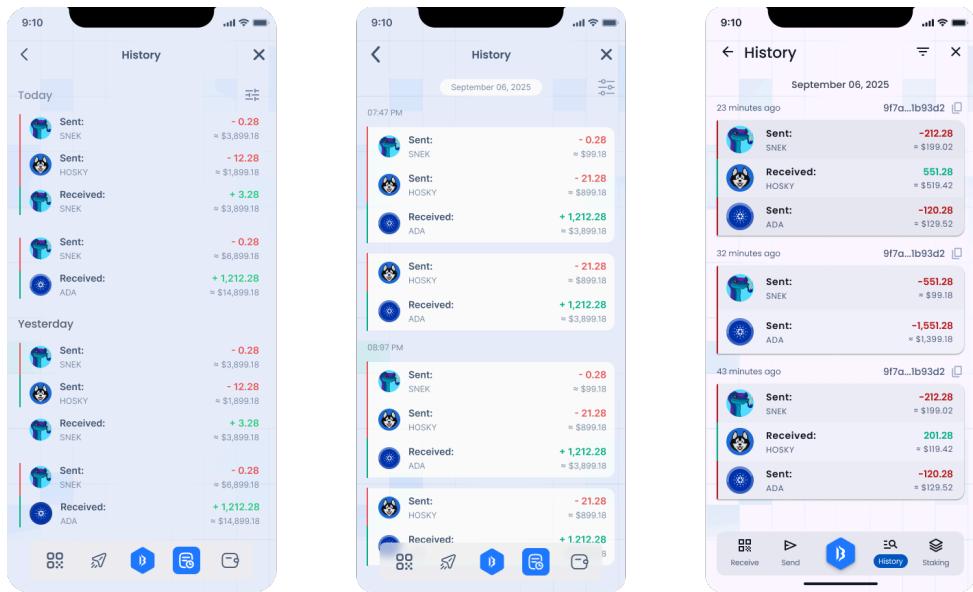
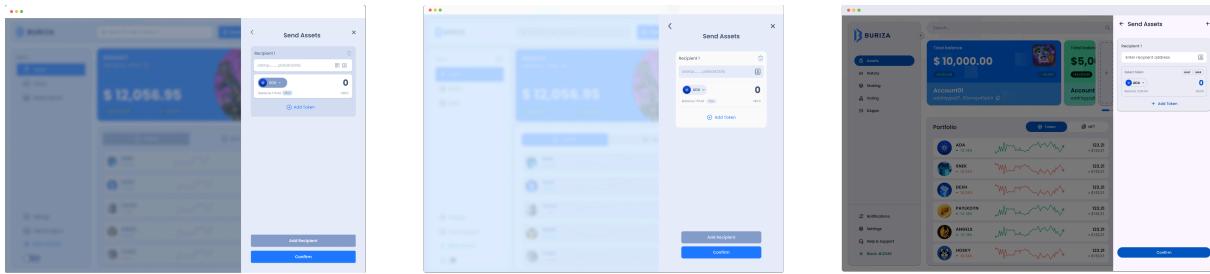


Figure 4 – Buriza Desktop UI - Send Assets: design evolution from first to third iteration (left to right)



Buriza's latest design update embodies SAIB's desire to create an impactful application that continues to evolve and improve. The team strives for continuous design iteration and improvement. With the future release of prototypes or mockups to the public, SAIB will continue to evolve through community feedback, validating users' needs and experiences while augmenting improvements with design solutions backed by research and best practices.

ii.2 Development

This section covers the technical implementation of Buriza's wallet suite, including architecture decisions, platform implementations, and front-end technologies.

ii.2.1 Project Architecture

Buriza's architecture is built around component sharing and platform independence, enabling a single codebase to serve multiple deployment targets.

ii.2.1.1 Solution Structure

Buriza uses a modular architecture with 5 .NET 9 projects designed for scalability and maintainability:

- **Buriza.UI** - Shared Blazor component library containing all user interface elements
- **Buriza.Data** - Core data models and services for wallet functionality
- **Buriza.Extension** - Browser extension for seamless dApp interaction
- **Buriza.Web** - Progressive web app for universal access
- **Buriza.App** - Cross-platform MAUI app for mobile and desktop

This architecture eliminates the need to implement separate UI layers for each platform. Instead of building distinct interfaces for the browser extension, web app, and mobile app, all platforms consume the same **Buriza.UI** components. This approach ensures consistent user experience while dramatically reducing development effort and maintenance overhead.

ii.2.1.2 Shared Component Library (**Buriza.UI**)

The component library provides a comprehensive set of reusable Blazor components organized into a hierarchical structure:

- **Common** - Foundational UI primitives like buttons, text fields, and navigation tabs
- **Controls** - Advanced interactive elements including asset cards and search functionality
- **Layout** - Application structure components for sidebars and main content areas
- **Pages** - Complete page/screen implementations for wallet operations like assets, transaction history, dapp access, send, and receive

The library leverages MudBlazor's Material Design 3 implementation alongside Tailwind CSS for utility-first styling and responsive design patterns. This combination ensures both visual consistency and flexible customization across all platform implementations.

ii.2.2 Platform Implementations

Each platform implementation leverages the shared **Buriza.UI** components while providing platform-specific features and deployment mechanisms.

ii.2.2.1 Browser Extension

The browser extension represents Buriza's most integrated approach to decentralized web interaction, embedding wallet functionality directly into the user's browsing experience. Built with Blazor WebAssembly and the Blazor.BrowserExtension framework, the extension follows Manifest V3 standards to ensure compatibility with modern browser security requirements and enhanced performance through service workers.

ii.2.2.2 Progressive Web App

The progressive web application serves as Buriza's most accessible deployment target, requiring only a modern web browser to provide full wallet functionality. Built with Blazor WebAssembly, the PWA delivers near-native performance by compiling C# code to WebAssembly bytecode that executes directly in the browser's runtime environment.

The PWA architecture enables installation directly from the browser without requiring app store distribution, creating a native-like experience while maintaining web-based deployment advantages.

ii.2.2.3 Cross-Platform App

The [MAUI](#) application targets iOS, Android, macOS, and Windows through a hybrid architecture that combines native platform capabilities with web-based UI components. At the core of this implementation is BlazorWebView, a native control that hosts Blazor content within a webview container while providing seamless integration with platform-specific APIs.

BlazorWebView acts as a bridge between the native MAUI shell and the Blazor UI layer, allowing the same **Buriza.UI** components to render within a native application context. This approach eliminates the need to rebuild the entire user interface using platform-specific controls like XAML, while still providing access to native device features such as secure storage, biometric authentication, and push notifications.

The hybrid model significantly reduces development complexity by leveraging existing web technologies and shared component libraries. Rather than maintaining separate native codebases for each platform, the application shares a single UI implementation while the MAUI framework handles platform-specific compilation and native API integration automatically.

ii.2.3 Front-End Implementation

The front-end implementation combines modern web technologies with component-driven architecture to deliver consistent user experiences across all platform deployments.

ii.2.3.1 Blazor Components

[Blazor](#) is Microsoft's web framework that serves as the foundational UI technology for Buriza, enabling C# development for web interfaces through WebAssembly compilation. This Microsoft-developed framework allows the entire Buriza application stack to use a single programming language, eliminating context switching between backend and frontend development.

The component architecture follows a hierarchical structure where complex UI elements are composed of smaller, reusable primitives. Blazor's two-way data binding simplifies form interactions and real-time updates, particularly important for wallet operations that require immediate visual feedback on transaction states and balance changes.

ii.2.3.2 MudBlazor Design System

[MudBlazor](#) v8.11.0 provides the Material Design 3 foundation for Buriza's visual language, delivering pre-built components with accessibility standards and smooth animations. The theming system integrates seamlessly with CSS custom properties, allowing Buriza to maintain brand identity while leveraging Material Design's proven usability patterns.

Buriza extends this foundation with Buriza-styled components:

Common Components:

- **BurizaButton** - Standardized button with consistent styling and interaction states
- **BurizaTextField** - Custom text input with validation and wallet-specific formatting
- **BurizaTabs** - Navigation tabs with custom styling for asset type switching
- **BurizaSelect** - Dropdown selection with wallet account and asset filtering
- **BurizaHeader** - Consistent header layout across all pages

Control Components:

- **BurizaAssetCard** - Asset display with balance, price changes, and interactive actions
- **BurizaSearchBar** - Universal search functionality for assets and transactions
- **BurizaDappCard** - dApp connection interface with authorization controls

ii.2.3.3 Tailwind CSS Integration

[Tailwind CSS](#) v4 complements MudBlazor by providing utility-first styling capabilities for custom layouts and responsive behavior. The integration uses a Bun build pipeline for rapid CSS compilation and automatic purging of unused styles, ensuring optimal bundle sizes.

The utility-first approach enables precise control over spacing, positioning, and responsive breakpoints, particularly valuable for wallet interfaces that require exact alignment for transaction details and balance displays. Custom CSS variables bridge the gap between Tailwind utilities and MudBlazor's theming system.

ii.2.3.4 Responsive Design Patterns

Buriza implements a mobile-first responsive design strategy that adapts to various screen sizes and interaction methods. The design system accommodates everything from mobile browser extensions to desktop applications while maintaining usability and visual hierarchy.

The responsive approach extends beyond screen size to consider platform-specific interaction patterns. Touch targets are appropriately sized for mobile interfaces, while desktop versions support keyboard navigation and hover states, ensuring consistent functionality regardless of how users access Buriza.

III. Implementation

This section outlines the implementation progress and highlights challenges encountered by the development team.

Buriza's unique infrastructure significantly streamlined the development process. Rather than maintaining separate codebases for each platform and operating system, the unified Buriza.UI codebase ensures compatibility across mobile, desktop, and web platforms—spanning Windows, macOS, Linux, Android, and iOS. As a result, platform-specific development primarily focused on using Tailwind's responsive directives to adapt the user interface across varying screen sizes.

To ensure both code quality and visual fidelity, the development process established had code peer-reviewed prior to being pushed to the repository. The team maintained a strong focus on building a beautiful, high-quality application through continuous collaboration and refinement.

iii.1 Mobile

Development began with a mobile-first approach, with the goal of closely following the design system established by the design team. The focus was on replicating the intended user experience, layout behavior, and visual responsiveness on mobile devices.

During this process, the team encountered a challenge related to how image assets were handled. Since the Buriza.App project (used to compile the mobile application) did not have direct access to image files stored in the shared Buriza.UI library, assets such as icons and illustrations could not be referenced at runtime.

To solve this, the team developed an internal utility that enables asset embedding directly into the application code. This ensured that all necessary visual elements remained accessible regardless of platform-specific limitations—eliminating the need for runtime file system access and enabling smooth, consistent rendering across platforms.

iii.2 Browser

Browser extension development leveraged the shared codebase, with only minor layout adjustments needed to fit the extension's popup dimensions. When fully expanded, the extension mirrors the desktop interface.

Similar to the mobile application, the extension faced challenges with accessing shared image assets once deployed in the browser environment. Since browser extensions operate in isolated environments with limited file system access, the same asset management solution used in the mobile app was also applied here. This unified approach allowed the team to maintain consistency in asset rendering without duplicating resources or code.

iii.3 Desktop

The desktop implementation also leveraged Blazor's flexible layout system. Developers were able to implement the drawer-based interface with minimal overhead, using dynamic layouts that adjusted based on page context.

Additionally, core features such as Send and Receive reused mobile UI patterns within drawers, allowing the team to focus on small adjustments rather than re-implementing entire views.

IV. User Interface

V. Conclusion

VI. References

- Google. (2025). *Material Design 3*. Material Design. <https://m3.material.io/>
- Google. (2023). *Manifest V3 - Chrome Extensions*. Chrome Developers. <https://developer.chrome.com/docs/extensions/mv3/>
- Microsoft. (2024). *ASP.NET Core Blazor*. Microsoft Learn. <https://learn.microsoft.com/en-us/aspnet/core/blazor/>
- Microsoft. (2024). *.NET Multi-platform App UI (.NET MAUI)*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/maui/>
- Microsoft. (2024). *BlazorWebView for .NET MAUI*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/maui/user-interface/controls/blazorwebview>
- Mozilla. (2023). *Progressive web apps (PWAs)*. MDN Web Docs. https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps
- MudBlazor Team. (2024). *MudBlazor - Blazor component library*. MudBlazor. <https://mudblazor.com/>
- Tailwind Labs. (2024). *Tailwind CSS - A utility-first CSS framework*. Tailwind CSS. <https://tailwindcss.com/>
- WebAssembly Community Group. (2023). *WebAssembly*. WebAssembly. <https://webassembly.org/>

VII. Links

VIII. References

Material Design. (2025). *Material Design 3*. Google. <https://m3.material.io/>