

Project Analysis & Design Document: Song Sharing Application (Camino Rojo)

Version: 1.13 Status: Draft Date: December 24, 2025

Changelog

Version	Date	Description of Changes
1.0	Oct 26, 2023	Initial Document Creation. Defined User Personas, Core Requirements, Domain Model, and Tech Stack.
1.1	Oct 26, 2023	Renamed document title to "Song Sharing Application".
1.2	Oct 26, 2023	Added Melody feature requirements. Specified ABC Notation as the format for rendering and playback.
1.3	Oct 26, 2023	Added Screen Inventory (Section 9) detailing the 4 core MVP screens.
1.4	Oct 26, 2023	Scope Adjustment: Simplified Phase 1 (MVP) to single-version songs only. Deferred multi-version logic to Phase 2.
1.5	Oct 26, 2023	Roadmap Restructure: Introduced Phase 2 (MLP). Moved Transposition, Melody, and Tags to Phase 2. Phase 1 reduced to essential upload/view.
1.6	Oct 26, 2023	Expanded Data Dictionary to include full schema for Categories (Tags) and Setlists. Added "Group by Category" feature to PDF Export.
1.7	Oct 26, 2023	Added detailed Technical Specifications (Section 8) for ABC Notation rendering logic and the Transposition Algorithm.
1.8	Oct 26, 2023	Document Reorganization: Moved Roadmap to Section 4. Shifted Domain Model, Architecture, and Tech Specs to follow.
1.9	Oct 26, 2023	Refined Screen Inventory (Section 9). Created subsections for Phases 1-4. Detailed MVP screens, left placeholders for future phases.
1.10	Oct 26, 2023	Added reference to visual mockup for Screen 1 (Home).

1.11	Oct 26, 2023	Corrected broken image link in Screen 1 Inventory; replaced with reference to generated HTML/CSS mockup file.
1.12	Oct 26, 2023	Added Changelog table to document history.
1.13	Dec 24, 2025	Updated document date.

1. Introduction

1.1 Purpose

The purpose of this application is to centralize, preserve, and share the medicine songs (Icaros, ceremonial songs) of the Camino Rojo community. It serves as a digital songbook that bridges the gap between oral tradition and accessibility, allowing community members to learn, practice, and perform these songs with accurate lyrics, chords, and melodies.

1.2 Scope

The application is a Progressive Web App (PWA).

- **It IS:** A collaborative library for lyrics and chords, a setlist management tool, and a reference for learning melodies.
- **It IS NOT:** A music streaming service (like Spotify) or a general-purpose social network.

1.3 Key Definitions

- **ChordPro:** A text-based format for writing music where chords are embedded in lyrics (e.g., Amazing [D]Grace).
- **ABC Notation:** A text-based shorthand for musical notation (e.g., C D E F | G A B c) used to render sheet music and play melodies.
- **Song:** For the MVP, a Song is a single entity containing lyrics, chords, and metadata. (In future phases, this will evolve into "Compositions" with multiple "Versions").

2. User Analysis

2.1 User Personas

1. **The Guitarero (Musician):** Needs accurate chords, the ability to transpose keys on the fly, and organized setlists for ceremonies.
2. **The Singer/Participant:** Needs clear, large lyrics to sing along during ceremonies. Needs to listen to the melody to learn.

3. **The Guardian (Admin/Elder):** Wants to ensure the songs are transcribed correctly and respectful of their origins.

2.2 User Roles & Permissions

- **Guest (Unregistered):** Can browse the library, search songs, view chords/lyrics, listen to audio, and use the transposition tool.
- **Member (Registered):** All Guest features + create favorites, build setlists, submit new songs, and export PDFs.
- **Admin/Moderator:** Can approve new song submissions and manage global tags.

3. Functional Requirements

3.1 Song Library & Discovery

- **Overview:** A paginated, lazy-loaded list of all Songs in the system.
- **Deep Search:** Search functionality queries both **Titles** and **Full Lyrics** (stripping chord brackets for accurate text matching).
- **(Phase 2 MLP) Advanced Filtering:** Filtering by Category (Water, Fire, etc.), Language, Key, and Author is deferred to the Lovable Product phase.

3.2 The Song Viewer (Player)

- **ChordPro Rendering:** Parses text like [Am]Hello into a visual format where chords appear above the corresponding syllable.
- **Audio Reference:** Embedded players or links (YouTube/SoundCloud/Spotify) if defined in the file metadata.
- **(Phase 2 MLP) Melody Display & Playback:** Rendering ABC Notation and playing synth notes.
- **(Phase 2 MLP) Transposition Engine:** The tool to shift keys (e.g., +2 semitones).
- **(Phase 3) Multi-Version Toggle:** Switching between arrangements.

3.3 Account & Community Features

- **Authentication:** Sign up/Login via Email/Password (Admin only for MVP).
- **(Phase 3) Personal Library:** "Heart/Star" songs.
- **(Phase 4) Setlist Management & PDF Export:** Create lists and export them.

3.4 Offline Support

- **PWA (Progressive Web App):** The app must be installable on mobile devices.

- **Caching:** Browsed songs cached for access in nature/offline environments.

4. Roadmap

Phase 1: The Bare Essentials (MVP)

- **Core Value:** Get the songs online and viewable.
- **Admin Upload:** Admin can upload a raw .cho (ChordPro) file. The system parses Title/Author from directives (e.g., {title: Song Name}).
- **Admin Delete:** Ability to remove a song.
- **Public Library:** Simple list view and Text Search.
- **Song Viewer:** Static ChordPro rendering (No transpose, no melody).

Phase 2: Minimal Lovable Product (MLP)

- **Core Value:** Make the app useful for musicians and learners.
- **Transposition:** Add the engine to shift keys.
- **Melody:** Render and Play ABC notation.
- **Rich Editing:** Admin UI to edit lyrics/chords in a text area (instead of just file upload).
- **Taxonomy:** Add Categories/Tags (Water, Fire, etc.) and Filtering.

Phase 3: Community & Evolution

- **Core Value:** Engagement and Crowd-sourcing.
- **User Accounts:** Public sign-up, Favorites.
- **Multiple Versions:** Users submit "Alternative Chords".
- **Voting:** Community votes on versions.

Phase 4: The Professional Toolkit

- **Core Value:** Utility for power users.
- **Setlists:** Create and manage ceremony lists.
- **PDF Generation:** * Generate a clean, printable PDF of a setlist.
 - **Smart Grouping:** Feature to optionally group songs in the PDF by Category (e.g., all "Water" songs together, then "Fire" songs).
- **Advanced Offline Mode.**

5. Domain & Data Model

5.1 Conceptual Model

Note: To support the future roadmap, the database schema will retain the Parent-Child structure (Composition -> Version), but for the MVP UI, we will effectively treat them as a 1-to-1 relationship.

5.2 Data Dictionary (Target Schema)

A. User

- id : UUID (Primary Key)
- email : String
- role : Enum (admin, moderator, user)
- created_at : Timestamp

B. Composition (The Song Identity)

- id : UUID
- title : String
- original_author : String
- primary_language : String
- created_at : Timestamp

C. Song_Version (The Content)

- For MVP, only one "Default" version exists per Composition.
- id : UUID
- composition_id : FK -> Composition
- content_chordpro : Text
- melody_notation : Text (ABC Notation)
- key : String (e.g., Am)
- capo : Integer
- audio_url : String
- contributor_id : FK -> User
- version_name : String (Default: "Standard")

D. Category (Tags)

- id : UUID (PK)

- name : String (e.g., "Medicine", "Icaro", "Water", "Fire")
- type : Enum (Theme, Rhythm, Origin)
- created_at : Timestamp

E. Composition_Category (Join Table)

- *Many-to-Many link between Songs and Tags*
- composition_id : FK -> Composition
- category_id : FK -> Category

F. Setlist

- id : UUID (PK)
- owner_id : FK -> User
- title : String
- description : String
- is_public : Boolean
- created_at : Timestamp

G. Setlist_Item

- id : UUID (PK)
- setlist_id : FK -> Setlist
- song_version_id : FK -> Song_Version
- order_index : Integer (Defines the sequence: 1, 2, 3...)
- transposition_override : Integer (Optional: Saves a specific key for this specific ceremony)

6. System Architecture

6.1 Technology Stack

- Frontend: Next.js (React), Tailwind CSS.
- Music Rendering: react-chord-pro (or custom parser), abcjs (Sheet music - Phase 2).
- Backend: Supabase (PostgreSQL + Auth).
- State: TanStack Query (Offline caching).

6.2 Offline Strategy

- **Service Workers:** Cache app shell (UI).
- **Data Persistence:** Store song JSON in `localStorage` via TanStack Query.

7. UI/UX Design Guidelines

7.1 Design Philosophy

- **Ceremony Mode:** Dark backgrounds, high contrast text.
- **Simplicity:** MVP Interface should feel like a simple songbook, not a complex database editor.

8. Technical Specifications

8.1 Parsing Logic

- **ChordPro:** Extracts chords `[Am]` and wraps them in spans for styling. The parser must distinguish between directives (lines starting with `{`) and content.
- **ABC Notation (Phase 2):**
 - **Library:** Utilizes `abcjs` for client-side rendering.
 - **Visual Rendering:** The `melody_notation` string is passed to `abcjs.renderAbc('target-div', abcString, { responsive: 'resize', scale: 1.2 })`. This ensures the sheet music SVG scales responsively on mobile devices.
 - **Audio Synthesis:** Uses `abcjs.synth` to generate an audio buffer on the fly. This provides a "Play Melody" button that highlights notes as they play (cursor tracking) using the `visualObj.setTiming()` listener.

8.2 Transposition Algorithm (Phase 2)

The system utilizes a Circular Chromatic Indexing approach to handle key shifts.

- **Reference Array:** `const CHROMATIC_SCALE = ['C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#', 'A', 'A#', 'B']`.
- **Normalization Map:** Maps flats to sharps for calculation (e.g., `{ 'Db': 'C#', 'Eb': 'D#', ... }`).
- **The Algorithm:**
 1. **Tokenization:** Regex extracts the Root note from the chord string (handling slash chords like `Am/G` by extracting both `A` and `G`).
 2. **Index Lookup:** Convert Root to standard Sharp format and find its index (0-11).
 3. **Calculation:** `newIndex = (currentIndex + delta) % 12`.

- **Handling Negatives:** If `newIndex < 0`, add 12 (e.g., -1 becomes 11/B).
- 4. **Reconstruction:** Retrieve note at `newIndex` and re-append chord quality (m, 7, sus4) and slash bass.
- **ABC Transposition:** The `abcjs` library includes a native logic `visualObj.transpose(int, true)`. This must be synchronized with the textual transposition state so both Lyrics/Chords and Sheet Music shift together.

9. Screen Inventory

9.1 Phase 1: MVP Screens (The Essentials)

Screen 1: The "Home" (Library) Screen

The entry point for all users.

- **Visual Reference:** [See generated file: `screen_1_home.html`]
- **Header:** App Logo (Left) & "Login" link (Right - for Admins only).
- **Search Bar:** Prominent input field at the top: "*Search title or lyrics...*".
- **Song List:** A simple vertical list of cards.
 - **Content:** Song Title (Bold), Author Name (Grey/Small).
 - **Admin Mode:** If logged in, each card displays a "Delete" (trash can) icon.
- **Empty State:** "No songs found" message if search yields no results.

Screen 2: The "Song Detail" Screen

The core viewing experience.

- **Visual Reference:** [See generated file: `screen_2_detail.html`]
- **Top Bar:** "Back" Arrow (Left), Song Title (Center).
- **Metadata Area:** Author Name, Original Key (e.g., "Key: Am").
- **Lyrics Container:**
 - The main scrolling area.
 - Displays lyrics in large, readable font.
 - Chords displayed above the relevant syllables in a contrasting color (e.g., Gold or Cyan).
- **External Media:** A YouTube/SoundCloud embed at the very bottom (only if a link exists in the file).

Screen 3: The "Admin Login" Screen

- **Visual Reference:** [See generated file: `screen_3_login.html`]
- **Layout:** Centered modal or page.
- **Form:** Email input, Password input.
- **Action:** "Sign In" button.

Screen 4: The "Upload Song" Screen (Admin Only)

- **Visual Reference:** [See generated file: `screen_4_upload.html`]
- **Header:** "Upload New Song".
- **Input Area:** A File Picker button that accepts `.cho`, `.txt`, `.pro` files.
- **Instructions:** Brief text: "*Upload a ChordPro file. Title and Author will be detected automatically.*"
- **Actions:**
 - "Upload" (Primary).
 - "Cancel" (Secondary).
- **Feedback:** Success message or Error toast upon completion.

9.2 Phase 2: MLP Screens (Lovable Product)

(Screens to be implemented in Phase 2)

1. **Song Detail (Enhanced):** Adds Transpose (+/-) buttons and Melody Player (Synth).
2. **Filter Modal:** UI for selecting Categories (Water, Fire) and Keys.
3. **Admin Song Editor:** Rich text area for manual editing of ChordPro and ABC notation.

9.3 Phase 3: Community Screens

(Screens to be implemented in Phase 3)

1. **User Signup/Login:** Public access.
2. **User Profile:** Shows "Favorites" list.
3. **Add Version Modal:** Form to submit an alternative version of an existing song.
4. **Version Comparison:** UI to toggle between different user versions.

9.4 Phase 4: Professional Screens

(Screens to be implemented in Phase 4)

1. **Setlist Manager:** Drag-and-drop interface for ordering songs.