

Product Requirements Document (PRD) — Video Game Catalog App

1. Overview

- **Product:** iOS app for cataloging personal video game collections.
- **Platform:** iPhone only, iOS 17+.
- **Orientation:** Portrait only.
- **Scope:** Offline-only, personal use.

2. Objectives

- Provide users with a structured, lightweight way to catalog and manage their games.
- Allow flexible tagging of properties and platforms, while preventing duplicates.
- Enable simple searching, sorting, and filtering.
- Allow JSON-based import/export for backup and migration.

3. Key User Stories (v1)

1. As a user, I want to add a new game with a title and release date so it's stored in my catalog.
2. As a user, I want to view my entire library as a searchable, sortable list.
3. As a user, I want to filter my list by properties like developer, genre, or system.
4. As a user, I want to edit game details and manage its properties/platforms.
5. As a user, I want to prevent duplicate entries based on title+release.
6. As a user, I want to import/export my data as JSON through the Files app.
7. As a user, I want to manage properties (rename/delete) centrally.
8. As a user, I want inline error validation and clear feedback when input is invalid.

4. Requirements

4.1 Data Model

- **Game:**
 - `title: String` (required)
 - `release: Int` (YYYYMMDD, required)
 - `status: Enum [library, wishlist]`
 - `properties: { series[], developers[], publishers[], genres[], modes[], platforms[] }`
- **Composite Key:** lowercase alphanumeric title + YYYYMMDD.
- **Properties:**
 - Series, Developers, Publishers, Genres → free text, case-preserved, canonical lowercase for uniqueness.
 - Modes, Systems, Formats → predefined enums.
 - Platforms = unique (system + format) pairs.

4.2 Validation

- Trim inputs, block empty/whitespace.
- Composite key uniqueness enforced.
- Enum validation on JSON import.
- Inline validation messages under fields.

4.3 Game List

- Default entry screen: list of all games.
- Search bar: titles only, live search.
- Sort: Title (A→Z), Release (asc/desc).
 - Secondary sort: always ascending by other field.

- Filters: multi-select, AND-only. Reset on leaving list.
- Toolbar: Search, Sort, Filter, “+” Add Game. Import/Export in menu.

4.4 Game Detail & Edit

- Shows title, release (formatted), properties, and platforms.
- Platform UI: group by system; formats listed with SF Symbols (digital/physical).
- Edit screen: grouped property summaries, “+ Add” buttons → navigation screens.
- Cancel: discard all changes; if dirty, confirm prompt.

4.5 Property Management

- Combined “Manage Properties” screen with sections for each type.
- Navigation flows into list/detail screens.
- Rename/delete supported.
- Deletion rules:
 - Modes: remove from all games.
 - Systems/Formats: delete associated Platforms and remove from all games.
 - Orphaned properties auto-removed if unused.
- Confirm prompt before destructive actions.

4.6 Import/Export

- JSON via iOS Files (UIDocumentPicker).
- Schema:
 - Required: title, release, status, properties.
 - Properties must include: series, developers, publishers, modes, platforms.
 - Platforms: { system, format }.
- Top-level `schemaVersion`.
- Validation:

- Reject duplicate composite keys in file.
- Reject enum mismatches.
- Report all errors.
- Options:
 - **Add:** add all; overwrite conflicts by composite key.
 - **Overwrite:** purge DB and replace; destructive confirmation.

5. Non-Goals (v1)

- Boxart/images.
- Notes or custom free-text fields.
- A–Z scroller/letter section headers.
- Metrics screen.
- Saved/smart filters.
- About screen.
- iCloud sync.
- Dark Mode, Dynamic Type, haptics, localization.

6. Future Considerations (v2+)

- iCloud sync.
- Dark Mode and accessibility enhancements.
- Metrics screen with statistics.
- Notifications/reminders for release dates.
- A–Z index scroller.
- Notes per game.
- Tap-through property navigation.

- Saved filter views.
- Multi-language support.

7. Risks

- Complexity of property/platform deletion cascades.
- JSON schema validation needs robust error handling.
- Large libraries (up to 1000 games) must remain performant.