Arcane Library

System Design Document

#### **Table of Contents**

Page

## 1. Introduction

* 1. Purpose of This Document
  2. References

1. System Architecture

2.1 Architectural Design

2.2 Decomposition Description

3. Persistent Data Design

5.1 Database Descriptions

5.2 File Descriptions

1. Requirements Matrix

Appendix A – Agreement Between Customer and Contractor

Appendix B – Peer Review Sign-off

Appendix C – Document Contributions

1. **Introduction**

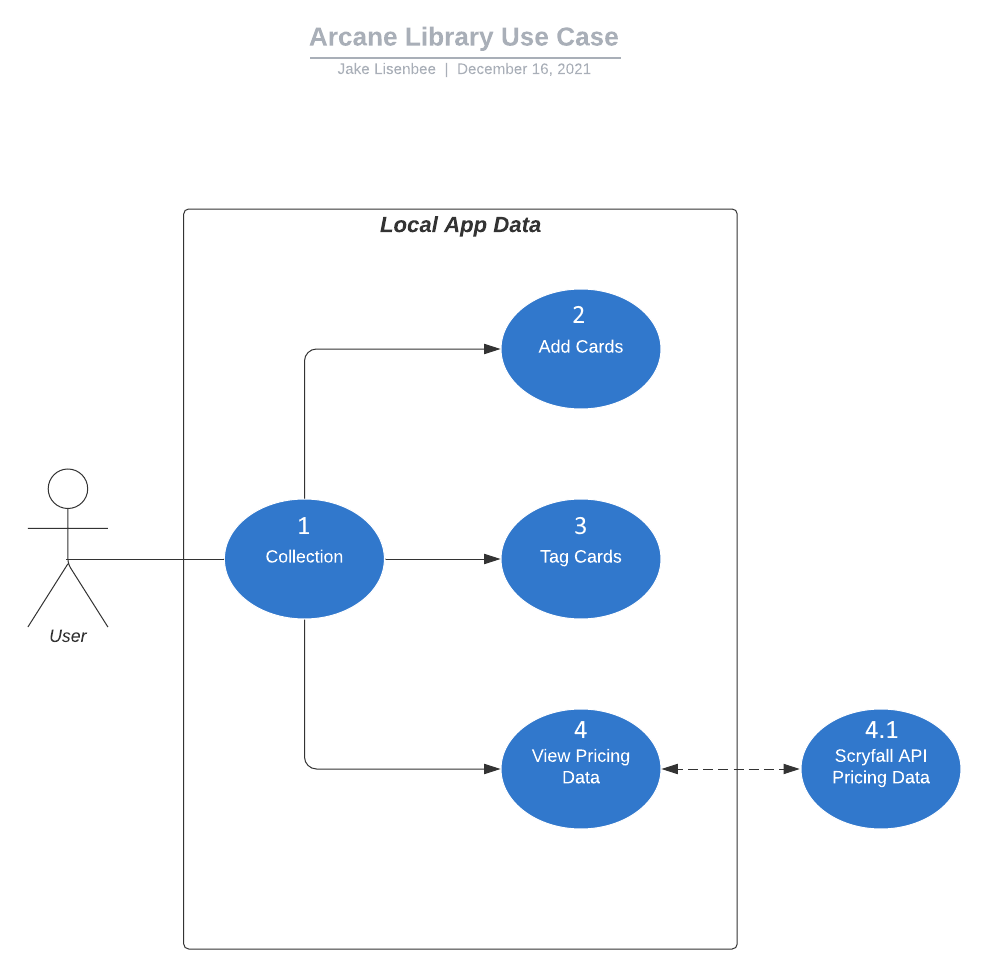
1.1 Purpose of This Document

This document outlines to future developers and the customer the intended flow of the Arcane Library trading card organizer by the users of the app. Outlined is the system architecture and flow of data throughout the system, as well as the organizational structures for maintaining that data.

1.2 References

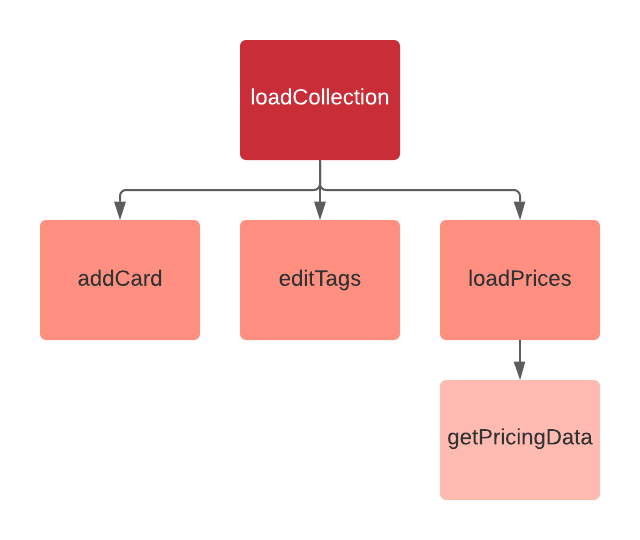
Lisenbee, J. R. (2021). (tech.). *Arcane Library System Requirements Specification* (pp. 1–9). Catonsville, Maryland: University of Maryland, Baltimore County.

1. **System Architecture**

2.1 Architectural Design

Above is the Flow of interaction for the Arcane Library card sorting app. The user first interacts with the software by loading in their collection. Then they can add cards to that collection, tag cards with custom tags, and load pricing data for your collection.

Hardware required is just an Android or IOS enabled device, and software required is React Native for flexible development on multiple platforms.

2.2 Decomposition Description

This chart describes the entirety of the program. As shown, only a few functions are needed as this is barely more than a pretty spreadsheet editor. When a user opens the app, loadCollection pulls the collection in from the saved file and displays it. Depending on how the user then interacts with the collection, and until the user exits, addCard, editTags, or loadPrices are called when a user Adds a Card, Wants to add, remove, or alter any tags on a card, or Get pricing data when the user saves the Collection.

1. **Persistent Data Design**
   1. Database Descriptions

There are no databases used for this software except the external one of the Scryfall API, which is not edited in any functionality of this software.

* 1. File Descriptions

There are a total of two files used for this entire program, the rest is drawn from existing style elements in the OS. The first is the file containing the data of the Collection, which is saved to on user request or app exit. The second is the .json holding the most recently requested pricing data from Scryfall for cards in the users collection, and is updated upon user request. Both are saved locally on the device.

1. **Requirements Matrix**

| **Requirement** | **How Satisfied** |
| --- | --- |
| #1: View Collection | The program constructs the collection saved locally on a .json file and displays it as the first screen the user sees. |
| #2: Add Cards | There is an add card button and search bar that allows the user to search for existing cards and add them to their collection. A quantity and set of tags are attached to each card listing. |
| #3: Tag Cards | By long-pressing on a card listing, the user can edit the tags on a card or add new ones through a popup menu. |
| #4: Request Pricing | When the listing is manually saved by the user, and if the user has an internet connection, the Scryfall API is called and pricing data is retrieved and applied to each card listing. |

**Appendix A – Agreement Between Customer and Contractor**

By signing this document, the customer agrees that this document is to be used as a description of the control of flow by potential users. They also agree that this document is subject to change at any time by request of either the AL development team or the customer based on the specifications below

Changes made to this document will require another round of sign-offs by all party members, the proposed schedule will be adjusted by an adequate amount as determined by the AL development team, and compensation renegotiated up to within 20% of the previously agreed amount. Changes to this document will not be reflected in the project planning until a maximum of 7 days after re-signing has occurred, during which period the changes may be negotiated without another signing.

| **Signer** | **Signature** |
| --- | --- |
| AL Development Team |  |
| Customer |  |

**Appendix B – Team Review Sign-off**

The AL Development Team has reviewed this document and found it sufficient to communicate to the customer the needs and requirements of the development team and customer.

| **Signer** | **Signature** |
| --- | --- |
| AL Development Team |  |
| Customer |  |