

Software Design Document

# Magic: The Gathering™ - Unofficial Collector’s Compendium

Icon

Description automatically generated with medium confidence

Version: 1.1.0

Written: 05/26/2023

[GitHub Repository](https://github.com/CSmith1998/MagicTheGathering-UnofficialCollectorsCompendium)

Pittsburgh Technical College

REVISIONS

Within this document, being the first in its series, no revisions to previous documents or plans are present. Instead, this version (v) of the Software Design Document (SDD) for the project of the working title “Magic: The Gathering™ - Unofficial Collector’s Compendium” shall set forth the first iteration of this documentation to act as a basis for any future changes or revisions.

# Revision Chart

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Primary Author(s) | Description of Version | Date Completed |
| Draft | Caine Smith | Initial draft. Created for distribution and comment review. | 04/27/2023 |
| 1.1.0 | Caine Smith | First Revision / Update. Revision of Traceability Matrix. | 05/26/2023 |

TABLE OF CONTENTS

# Index

1. Introduction 3
   1. Design Overview 3
   2. Requirements Traceability Matrix 3
2. System Architectural Design 4
   1. Chosen System Architecture 4
   2. System Interface Description 4
3. Detailed Description of Components 5
   1. Component-n 5
4. User Interface Design 6
   1. Description of User Interface 6
      1. Screen Images 6
      2. Objects and Actions 6
5. Additional Material 7

INTRODUCTION

# Design Overview

This document will cover the overall design of relevant components and the user interface. Proceeding through this documentation, the description of the systems and subsystems to be represented will begin with a higher level overview, narrowing to finer details in further sections. As the first version of this documentation, details for some or all components or systems may still be vague, and all details are subject to possible alterations in future revisions.

# Requirements Traceability Matrix

| ID | Assc. ID | Requirements Description | Business Need | Specification | Test Cases |
| --- | --- | --- | --- | --- | --- |
| 1 | 1.1 | Landing Page | Engages user, hooks their attention. | Not Started | N/A |
| 2 | 1.2.2 | Email Confirmation | Ensures only valid emails are used. | SendGrid Email Verification | N/A |
| 3 | 1.3.1 | Login Page | Users need way to access user content. | Input for Email & Password. Captcha challenge. Login button. | N/A |
| 3 | 1.3.2 | Forgotten Password Link | Helpful utility. Reduces need for support team. | Not Started | N/A |
| 3 | 1.3.3 | Captcha | Discourages robots/programmatic access | Google reCAPTCHA Implementation | A.1.1 |
| 6 | 1.6.2.1 | Track Logins | Allows user engagement research | Not Started | N/A |
| 6 | 1.6.3 | Usage Statistics | Track use time/trends | Not Started | N/A |
| 6 | 1.6.3.1 | Average User Access Amount | Statistics for marketing research | Not Started | N/A |
| 6 | 1.6.3.2 | Average User Access Length | Statistics for marketing research | Not Started | N/A |

SYSTEM ARCHITECTURAL DESIGN

# Chosen System Architecture

[Solid lines represent confirmed connections – Dotted represent possible/undecided connections.]

A picture containing diagram

Description automatically generated

# System Interface Description

The system interface for this application is deceptively simple. At the very ground level is the data repository, our Azure database. Here, pertinent information such as user details, roles, cards, and collections are stored in a safe, secure manner to allow data to be accessed efficiently anytime and anywhere, provided the access is authorized.

The next level of the system is the APIs. A custom API coded specifically for this project handles all access to the database outside of direct access from authorized personnel for update and maintenance purposes. Additionally, the custom API for this project handles most, if not all, interactions with a 3rd party API that serves as a source of up-to-date card information. Card information gathered from the 3rd party API will be securely stored in the project’s database when a card is added to a user’s collection, provided a copy of that card does not already exist within the stored data.

Finally, the top-level of the system is the user interface, or web application. All commands, apart from those issued for testing or maintenance purposes by administrators, will flow from interaction with the web app. By manipulating the interface, users can search through the available repository and database, and perform CRUD operations on their personal collection to fulfill the purpose of the application.

DETAILED DESCRIPTION OF COMPONENTS

# Azure Database

This component, as the name suggests, is a database repository for storing dynamic data so that users can reliably access persistent data during each use of this project’s primary product. The database must be capable of accepting a steady flow of complex data manipulation commands in a swift and orderly fashion and should be capable of storing a wide range of data in a secure manner.

It is important that care is taken to ensure that end-users do not gain access to unauthorized portions of the database for any reason, so that the integrity of the data contained within can be maintained.

# Custom Web Service

As the component of this project that handles the brunt of the work, the custom web service exists as the middle-man for communication between the user interface, 3rd party services, and the database. All commands that interact with the central database flows through the approved methods and checks of the web service, which allows for sensitive data to be encapsulated and limits the possibility of unauthorized access to the database. Additionally, through use of a web service as a middle man, only approved actions can be taken against the underlying database, aiding in ensuring that no user can cause a catastrophic data breach through ordinary means.

# 3rd Party Repository Service

This component, being an external resource to this project, exists solely as a means for attained up-to-date card information. The web application and/or web service will call to this service to ensure the accuracy of card data and allow for new cards to be easily added to the central database and, by proxy, users’ collections.

# Website (User Interface)

Finally, the web interface is the face of the project, being the direct interface with which users interact with the product as a whole. As users navigate the site, the web app will transmit and receive commands and date to and from the web services to create, read, update, and delete information as the user requires. Design is important for this component, as users should be able to understand what they are looking at and how to make use of the tools and features presented to them. Additionally, design should be enticing to keep the user feeling satisfied with the visual, and technical, experience.

USER INTERFACE DESIGN

# Description of the User Interface

The interface for this project is a web application that enables users to interact with API services to create, read, update, and delete data within a central database. By accessing the site, users can navigate the UI through the standard methods of links, buttons, or urls to accomplish a variety of tasks.

From the landing page, users have the option to peruse the home screen to familiarize themselves with the site, login as an existing user, or register as a new user. Once a user is logged in, depending upon their role and access level, the user may either edit their card collection, search a card database, or manage administrative functions. Following the wise examples set forth by Steve Krug, our users will experience a simple, yet robust interface that excels at its intended functions while remaining visually interesting and easy to use.

## Screen Images

Images of the user interface are not yet available. To be provided in future revisions.

## Objects and Actions

To be provided in future revisions.

ADDITIONAL MATERIAL