# MTG Inventory Manager Requirements

—

By: Dewayne Scrivner, Jesse Pickard, Jordan Weaver, and Mason Allen

# Abstract (The Problem)

Magic: The Gathering (MTG) has over 27,000 unique cards

The volume of cards can make inventory management difficult

This leads players to struggle to find what cards they own and what cards are in decks

Poor tracking can lead to inefficient decks or overlooked strategic options when making decks

# Abstract (The Solution)

This project would be a web based Inventory Manager for anyone to use

With:

- Organized digital tracking
- Integrated deck builder
- Searchable card database

# Tools and Technology

**Fronted**

- React
- NextJS

**Backend & Database**

- Supabase

**Deployment**

- Vercel

**Card Data Source**

- Scryfall API

**Development**

- Visual Studio Code
- GitHub

# Login

Users originally navigate the site as a guest.

A login is required for certain site functions.

- Cannot access Deck, Inventory, or Profile without being signed in.
- Profile is not displayed unless the user is signed in.

A login button will be visible when the user is not signed in.

- Profile avatar otherwise.

Login is initiated depending on site navigation.

- Trying to access Deck or Inventory will redirect to sign-in.
- Upon success, the user will be redirected to the appropriate page.
- On fail, the user is prompted to try again.
- On cancel, the user is taken back to the homepage.

Supabase will be used to allow account creation with a Google login.

- Handles all authentication for us.

# Home/Search Page

**Includes Navigation Bar**

- (See Section 3 for Navigation Bar details)

**Displays cards**

- Displayed as images.
- Initially sorted alphabetically
- Displayed in sections of 50
- Users can scroll through displayed cards

**Selecting a card opens the Card Information Page**

- (See Section 8 for Card Information Page details)

**Filtering options available**

- (See Search Function for details)

**Loading behavior:**

- Displays loading indicator while retrieving data

**Failure behavior:**

- If the Scryfall API request fails, an error message is displayed

# Navigation Bar

**Search Bar**

- Input box for searching
- (See Search Function for details)

**Profile Icon (User Avatar)**

- Opens dropdown menu
- Profile Settings → Opens Profile Page
- Log Out → Confirmation pop-up
    - “No” cancels
    - “Yes” logs out → Returns to Homepage

**Deck Button**

- Navigates to Deck Page

**Inventory Button**

- Navigates to Inventory Page

**Home Button**

- Navigates to the Homepage.

**If User is Logged out**

- Clicking any button will bring them to the login page.

# Search Function

The search function is core to our project and exists in three main forms depending on where you access it from.

**Search Page**

- Most advanced, contains several filter options
- Search function is on a collapsable menu
- Several Cards are displayed at once

**Card Info Page**

- Search function contained in the navbar
- Allows for the searching of a card by name only

**Deck Display Page**

- Search is contained in the navbar
- Searches by name
- Quickly adds the card to the deck but not the inventory.

# Inventory

The utilizes the search page for its operation. The key difference is the source of the data, by default the search page accesses the entire database but the inventory will only display what the user has added to their collection. Decks are displayed here as well, but the deck page itself is a subsection of the inventory utilizing another restriction.

The user can access their inventory from the navbar.

# Deck Page

This page will allow users to view their decks as stacks of cards sorted by category

- By default the category is the same as their first card type
- Users can assign a card to a different pre labelled category such as, draw, removal or stax.

This page will also provide a small summary of the decks contents.

# Card Information Page

Displays high-resolution card image

- Displays user ownership data:
  - Quantity owned
  - Quantity currently used in decks

**Card Details Displayed**

- Card name
- Mana cost (symbol images)
- Type, subtypes, and supertypes (when applicable)
- Set name, rarity, and collector number (when applicable)
- Full rules text
- Artist name
- Power/Toughness (when applicable)
- Flavor text (when applicable)

# Card Information Page

**Format Legality**

- Displays legality for supported formats:
    - Commander
    - Standard
- Status shown as **“Legal”** or **“Not Legal”**

**Deck Integration**

- **Add to Deck** button
- Opens deck selection menu
- User confirms or cancels selection
- On confirmation, card is added to selected deck(s)

**Inventory Integration**

- **Add to Inventory** button
    - Increases owned quantity by 1
- **Remove from Inventory** button
    - Decreases owned quantity by 1
    - Hidden when owned quantity equals 0

# Database

## User Data

- Email
- Unique user ID

## Card Data

- Name
- Color(s)
- Cost
- Type
- Ect.

## Inventory Data

- Name and number of each card in user's inventory
- Will update number when card is added or removed in any way

## Deck Data

- Unique deck ID
- Deck information
  - Name
  - Color(s)

# Risky Requirement 1

## Risk:

Scryfall API failure/slow response time

## Impact:

- Search delays
- Incomplete data

## Mitigation:

- Error messages
- Loading indicators
- Caching Data

# Risky Requirement 2

Risk:

Dynamic UI behavior could cause bugs

Impact:

- UI Design
- Program not working as intended

Mitigations:

- Have default values
- Consistent Layout
- Testing all cases

# Risky Requirement 3

Risk:

Real Time Deck and Inventory Sync

Impact:

- Accidental Duplicate
- Accidental Negative Card Amounts
- Inconsistent Count of Cards

Mitigations:

- Adding constraints to card amount
- Validation Rules