

# Capstone Stage 1 - Project Requirements

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Search Results](#)

[Search Parameters](#)

[Card Details](#)

[Slide Navigation](#)

[Decks](#)

[Add/Edit Deck](#)

[Deck List](#)

[Collection](#)

[Collection Contents](#)

[Favorites Listing](#)

[Settings](#)

[Card Search - Tablet](#)

[Decks - Tablet](#)

[Card Details](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement Sqlite database](#)

[Task 3: Implement loading of card prices](#)

[Task 4: Implement loading of card images](#)

[Task 5: Build UI for MainActivity](#)

[Task 6: Build UI for Search Result Fragment](#)

[Task 7: Build UI for Search parameters Fragment](#)

[Task 8: Build UI for Card View Fragment](#)

[Task 9: Build UI for Deck List Fragment](#)

[Task 10: Build UI for Deck Content List Fragment](#)

[Task 11: Build UI for Add/Edit Deck Fragment](#)

[Task 12: Build UI for Collection List Fragment](#)

[Task 13: Build UI for Collection Content Fragment](#)

[Task 14: Build UI for Favorites Fragment](#)

[Task 15: Build UI for Slide out Navigation](#)

[Task 16: Build UI for Settings Activity](#)

[Task 17: Implement Google Services](#)

[Task 18: Implement Material Design](#)

[Task 19: Implement ShareActionProvider](#)

[Task 20: Implement Favorites Widget](#)

**GitHub Username:** Hankahn

# Magic the Managing

## Description

This app will give Magic the Gathering players a way to do some common tasks on the go. The tasks include searching of cards, building decks, managing their collections and retrieving up to date prices for cards.

## Intended User

Magic the Gathering players and collectors.

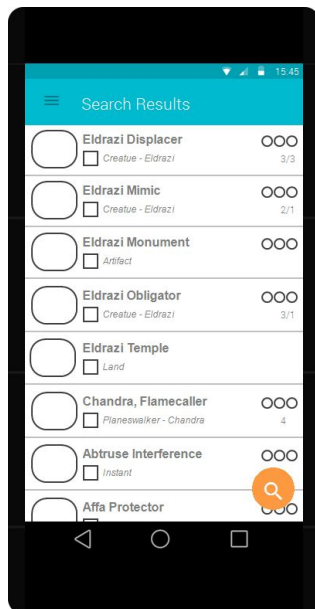
## Features

- Perform searches for cards.
- Build and manage decks.
- Manage a collection.
- Get up to date prices for card(s).
- Mark cards as favorites for easier finding.
- Allow multiple users to share that app on one device by using Google Identity.
- Share cards and decks.
- Widget that will list favorite cards on the home screen.

## User Interface Mocks

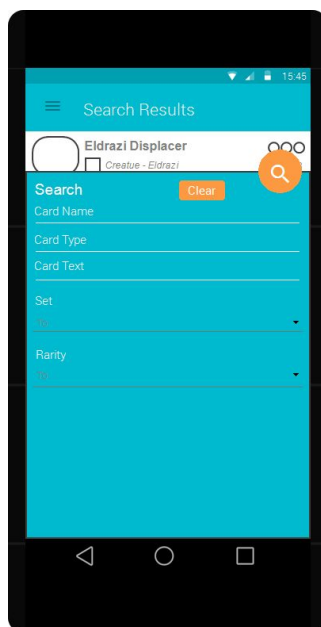
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

## Search Results



This is the initial screen when loading the app. It is a list of all the available cards. Each line represents a card (from left to right) with a cropped image of the card, the title, set/rarity icon, card type and mana cost. When clicking on a card it will load the card detail screen for it. Long pressing on a card will allow the user to add it as a favorite, add it to a deck or edit how many copies the user has in their collect. The floating action bar will bring up a bottom sheet of search parameters.

## Search Parameters



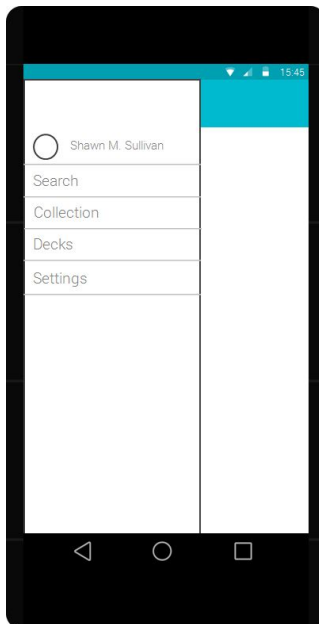
This screen will allow the user to adjust the search parameters to find cards based on name, type, text, set, rarity and other possible fields. The user can clear the parameters to reset. The user can swipe down or click outside the bottom sheet to dismiss.

## Card Details



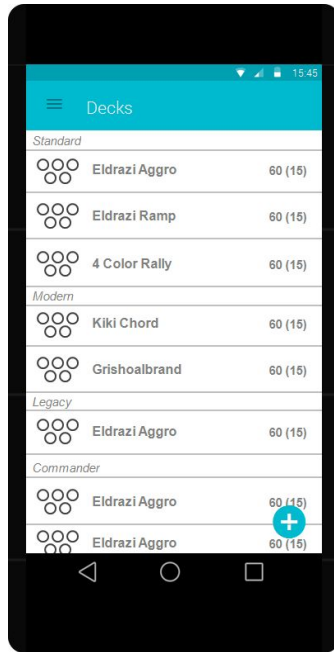
This screen will show an image of the card, prices, rulings and other printings. The rulings and printings are a ViewPager so they can click the tabs to view the respective data. This screen is also part of another ViewPager so the user can swipe between other cards in the search results.

## Slide Navigation



This is the slide navigation for the app. It will allow the user to quickly jump from the various parts of the app. It will also show a random cropped card image along the top as a background. When a user is signed in using Google Identity it will display their image and name.

## Decks



This screen will list the decks created by the user grouped by format. Each item represents (from left to right) a deck with the colors of the deck, the title and the number of cards in the deck and the sideboard. Long pressing on a deck will load the edit deck screen. Clicking the floating action bar will allow the user to create a new deck.

## Add/Edit Deck

This screen will allow the user to add or edit a deck. They can supply a name, format and description.

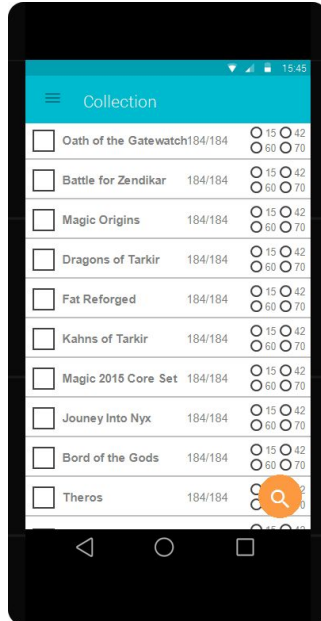
## Deck List

Card Type	Card Name	Rarity/Set Icon	Type	Mana Cost	Copies (Main Deck / Side Board)
Creatures (8/2)	Eldrazi Displacer	3/3	Creature - Eldrazi	4 0	
	Endless One	3/3	Creature - Eldrazi	2 0	
	Eldrazi Mimic	3/3	Creature - Eldrazi	2 2	
Instants (5/3)	Lightning Bolt		Instant	4 0	
	Hero's Downfall		Instant	1 3	
Enchantments (4/0)	Bonds of Morality		Enchantment	2 0	
	Containment Membrane			2 0	

This screen will list the contents of a deck grouped by card type. Each card item represents (from left to right) the cropped card image, card title, rarity/set icon, type, mana cost and number of copies in the main deck and in the side board. Long pressing a card will bring up a dialog to allow the user to modify how many copies of the card the user has in the main deck and side

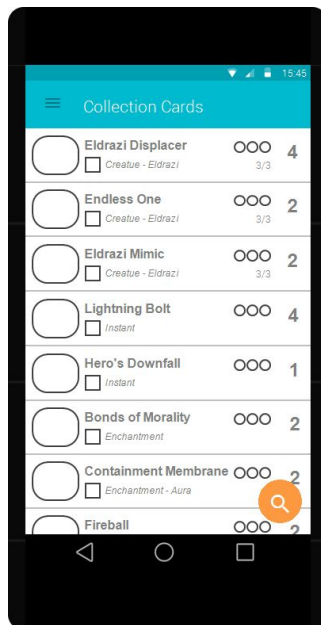
board. Clicking on the floating action button will bring up the search parameters bottom sheet to initiate a search so the user can find cards to add.

## Collection



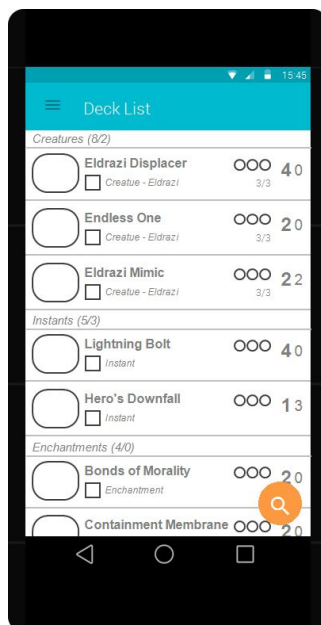
This screen will list all the sets and how many cards of each rarity from the set the has in their collection. Each item represents (from left to right) the set icon, set name, number of cards from the set the user owns and the number of cards at each rarity from the set the user owns. Clicking on the floating action button will bring up the search parameters bottom sheet to initiate a search so the user can find cards to add.

## Collection Contents



This screen lists the cards from each set and how many copies the user owns. Each item represents (from left to right) the cropped card image, card title, set/rarity icon, type, mana cost and number of copies owned. Clicking on the floating action button will bring up the search parameters bottom sheet to initiate a search so the user can find cards to add.

## Favorites Listing

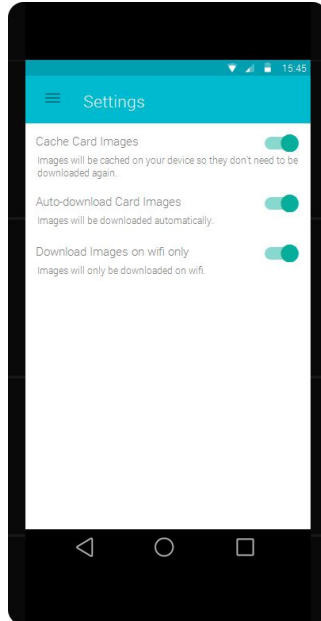


This is the slide navigation for the app. It will allow the user to quickly jump from the various parts of the app. It will also show a random cropped card image along the top as a background.



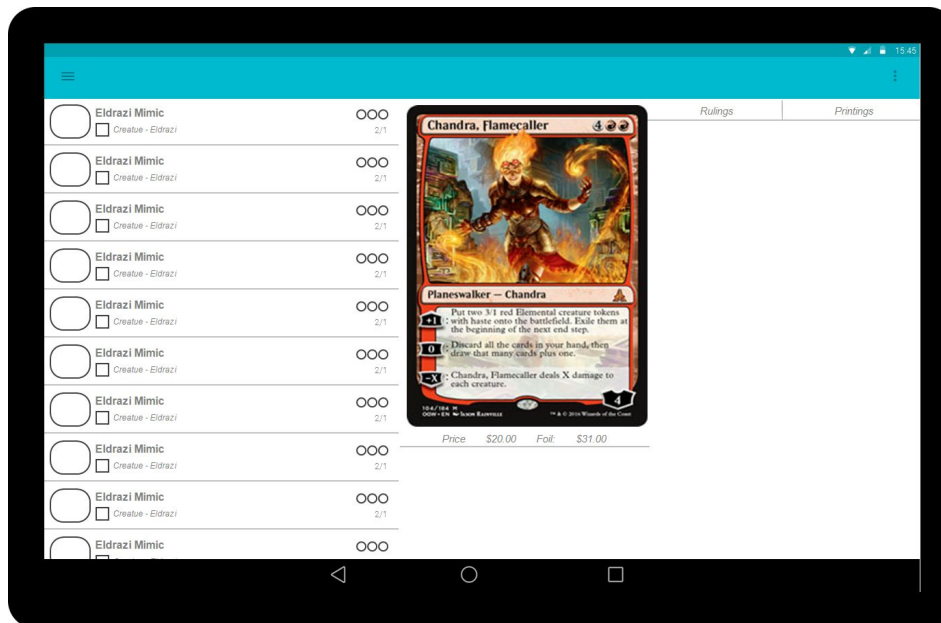
When a user is signed in using Google Identity it will display their image and name. Clicking on the floating action button will bring up the search parameters bottom sheet to initiate a search so the user can find cards to add.

## Settings



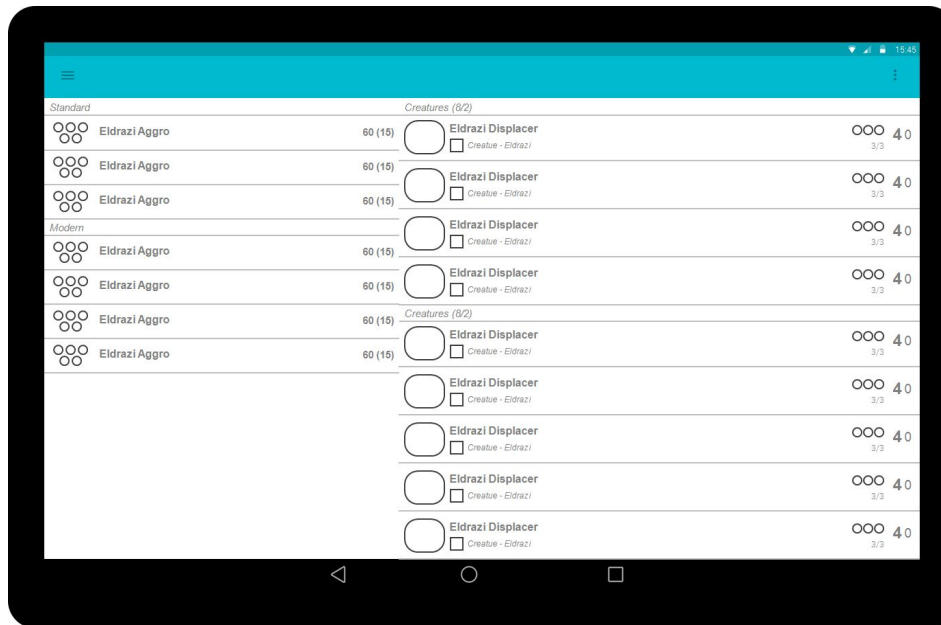
This screen will allow the user to manage the settings for the app.

## Card Search - Tablet



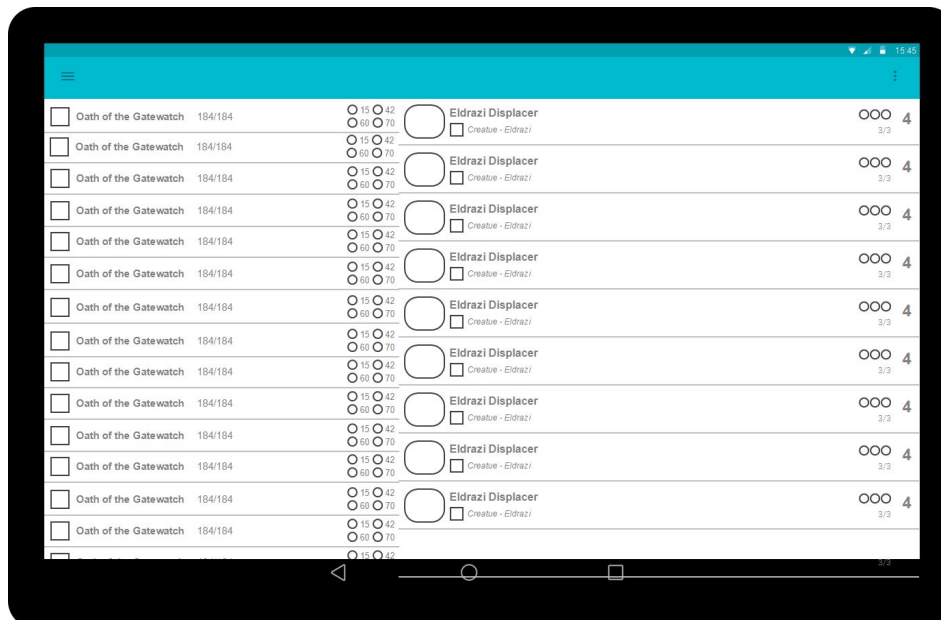
This screen combines the search results fragment and card detail view fragment to better use the available screen real estate provided by a tablet.

## Decks - Tablet



This screen combines the deck list and deck content list fragments to better use the extra screen real estate provided by a tablet.

## Card Details



This screen combines the collection list and collection content list fragments to better use the extra screen real estate provided by a tablet.

## Key Considerations

### How will your app handle data persistence?

The app will retrieve card data and store user created decks, collection and card prices in a SQLite database with a Content Provider. Possible settings for the app will be stored in SharedPreferences.

### Describe any corner cases in the UX.

Rotating the device while retrieving card images and card prices should not crash the app. If the device is not online it should not

### Describe any libraries you'll be using and share your reasoning for including them.

- Android-sqlite-asset-helper: For managing the installation of the pre existing sqlite database and handling possible database updates.
- Retrofit: For making calls to the card price web service.
- Picasso: For retrieving/caching card images from a remote server.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

### Task 1: Project Setup

- Add the libraries (android-sqlite-asset-helper, Retrofit and Picasso)
- Create a signed apk
- Setup free and paid releases

### Task 2: Implement Sqlite database

- Create external sqlite database file
- Place in assets folder
- Implement android-sqlite-asset-helper library to load the database from the assets folder
- Implement handling future database updates to preserve user created data while updating system data.
- Create methods for retrieving, inserting and updating database data.

- Implement ContentProvider.

### **Task 3: Implement loading of card prices**

- Acquire access to TCG Player pricing API.
- Implement IntentService to load card prices.
- Store retrieved prices into Sqlite database.

### **Task 4: Implement loading of card images**

- Implement AsyncTask for loading of card images using Picasso.
- Implement caching of retrieved images.

### **Task 5: Build UI for MainActivity**

- Create layout.
- Setup MainActivity so that only one fragment is shown on mobile devices and two fragments are shown on tablet devices to better use space.

### **Task 6: Build UI for Search Result Fragment**

- Add RecyclerView for displaying results.
- Create layout for card items.
- Implement event handlers to react to user input.

### **Task 7: Build UI for Search parameters Fragment**

- Create a layout for a bottom sheet that will hold the search parameters.
- Implement event handlers to react to user input.

### **Task 8: Build UI for Card View Fragment**

- Create a ViewPager to allow user to scroll through card data for all cards in search results.
- Create a second ViewPager to scroll between the card data tabs.

### **Task 9: Build UI for Deck List Fragment**

- Add RecyclerView for displaying results.
- Create layout for deck items.
- Implement event handlers to react to user input.

### **Task 10: Build UI for Deck Content List Fragment**

- Add RecyclerView for displaying results.
- Create layout for deck card items.
- Implement event handlers to react to user input.

### **Task 11: Build UI for Add/Edit Deck Fragment**

- Build layout by adding necessary controls.
- Implement event handlers to react to user input.

### **Task 12: Build UI for Collection List Fragment**

- Add RecyclerView for displaying results.
- Create layout for collection set items.
- Implement event handlers to react to user input.

### **Task 13: Build UI for Collection Content Fragment**

- Add RecyclerView for displaying results.
- Create layout for set collection card items.
- Implement event handlers to react to user input.

### **Task 14: Build UI for Favorites Fragment**

- Add RecyclerView for displaying results.
- Create layout for favorite card items.
- Implement event handlers to react to user input.

### **Task 15: Build UI for Slide out Navigation**

- Create a layout for the slide out navigation.
- Implement event handlers to react to user input.

### **Task 16: Build UI for Settings Activity**

- Create a layout for managing the available settings for the app.
- Implement event handlers to react to user input.

### **Task 17: Implement Google Services**

- Implement Google Identity for differentiating users of the device so they can have their own content (decks, collections and favorites).
- Implement AdMob ads for a free release build variant.

### **Task 18: Implement Material Design**

- Implement shared element transitions
- Implement parallax scrolling.

### **Task 19: Implement ShareActionProvider**

- Implement ability to share card and deck information.

### **Task 20: Implement Favorites Widget**

- Create layout for widget.
- Implement loading of users favorites.
- Implement event handlers to react to user input.