How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: bdaoust

MTG Manager

Description

MTG Manager is an easy to use Magic the Gathering (http://magic.wizards.com/en) deck manager app that allows you to create and manage various MTG decks, add or remove cards and specify the quantity of a given card.

With MTG Manager you can:

- Keep track of your existing MTG decks, or use it to easily brainstorm new deck ideas.
- Access your decks anywhere, even while offline
- Search through a wide collection of MTG cards to add to your decks
- Synchronize your MTG decks across your various devices

Intended User

This app is intended for people who play MTG using physical* cards and who want an easy way to keep track of their various decks**.

* People who play the digital version of the game do not need a standalone app to keep track of their decks.
** A deck is a collection of MTG cards (normally around 60 cards), that are chosen by the player. A MTG deck is then used to compete against other MTG players.

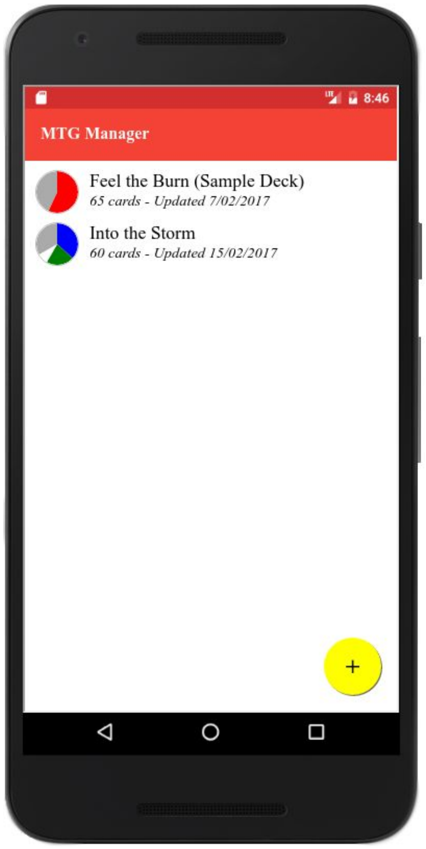
Features

- Create MTG decks
- Search for MTG cards from a collection of thousands of unique cards
- Add cards to decks / Remove cards from decks
- Synchronize decks with the user's other devices
- View a list of your MTG decks in a widget

User Interface Mocks

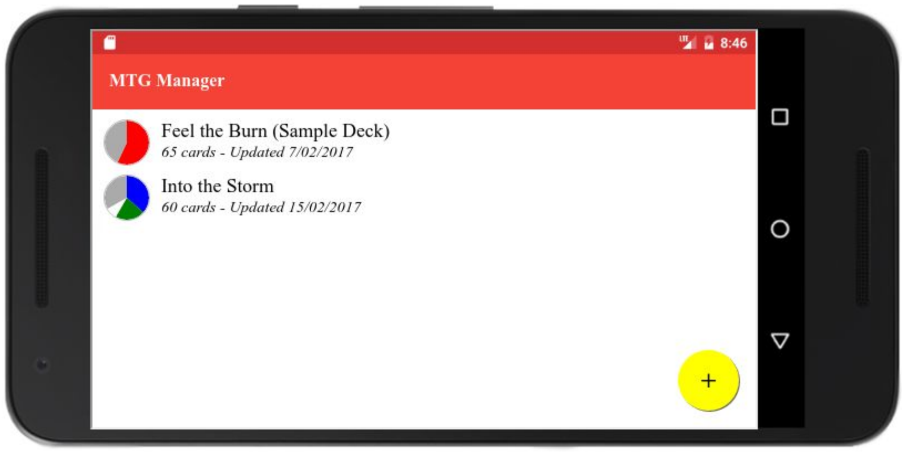
The first screen that will be shown will be a login screen. The FirebaseUI library will be used to implement the interface (screen not shown). Once logged in, the user will be presented with a list of decks (a sample deck called "Feel the Burn" will be provided) on the phone, and a list of decks + a detail view of the sample deck on the tablet.

Screen 1 - Deck List (Phone - Portrait)


8:46
MTG Manager
Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017
Into the Storm
60 cards - Updated 15/02/2017
+

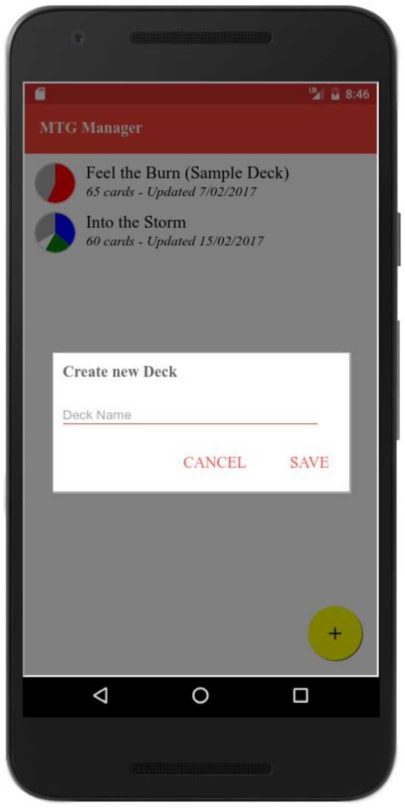
This screen shows the list of decks that the user has. Each deck item includes a deck name, number of cards, and last updated date. There is also a custom view (i.e. pie chart) which shows the breakdown of the different card colors used in the deck. For example, the Feel the Burn deck contains about 57% red cards and 43% colorless cards. The user can click on the FAB to create a new deck.
Note: The default install of the app would only contain 1 sample deck.

Screen 1 - Deck List (Phone - Landscape)


8:46
MTG Manager
Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017
Into the Storm
60 cards - Updated 15/02/2017
+

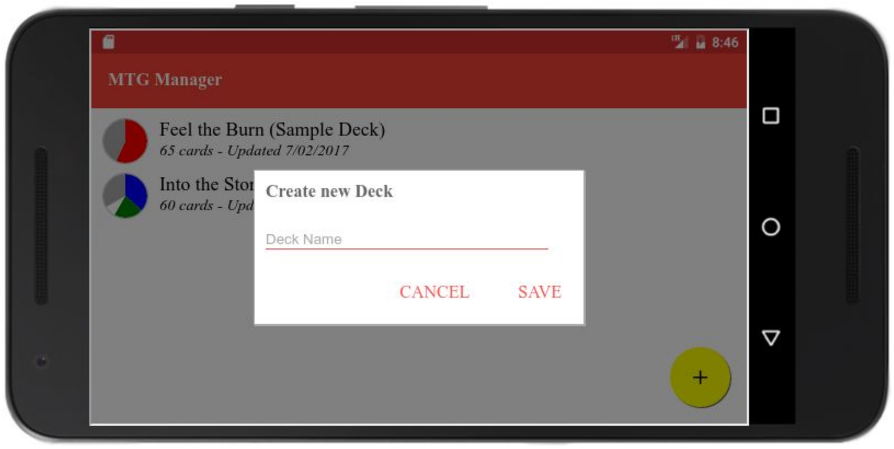
Same as the previous screen, but in landscape mode

Screen 2 - Deck List with Create new Deck dialog (Phone - Portrait)


8:46
MTG Manager
Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017
Into the Storm
60 cards - Updated 15/02/2017
Create new Deck
Deck Name
CANCEL
SAVE
+

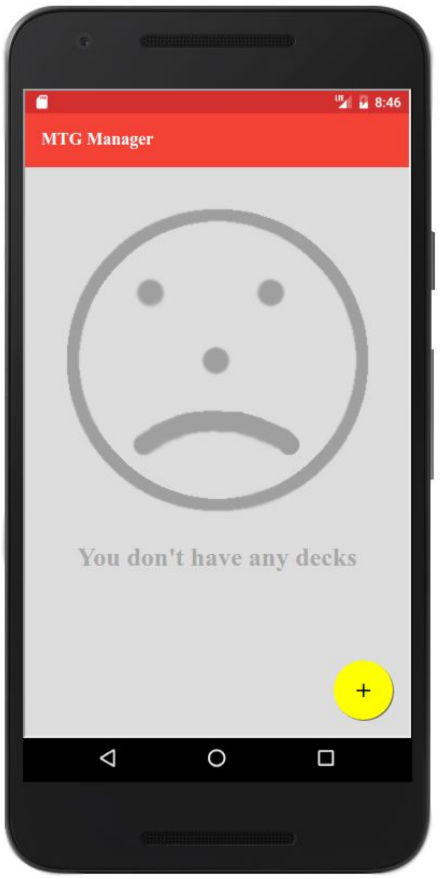
If the user clicks on the FAB (+ sign), a dialog will pop up allowing the user to create a new deck.

Screen 2 - Deck List with Create new Deck dialog (Phone - Landscape)



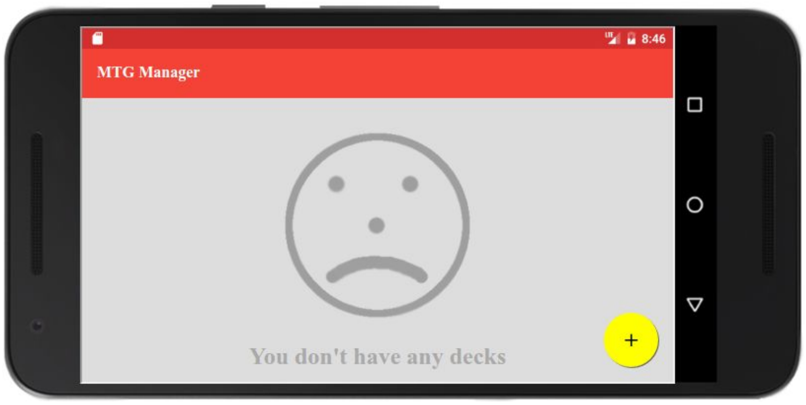
Same as the previous screen, but in landscape mode

Screen 3 - Empty Deck List (Phone - Portrait)



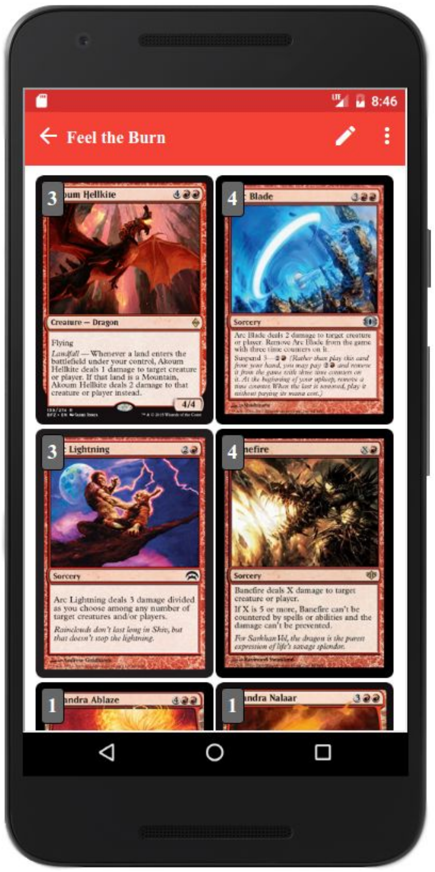
In the event that a user has deleted all their decks (including the sample deck) they will be presented with this screen.

Screen 3 - Empty Deck List (Phone - Landscape)



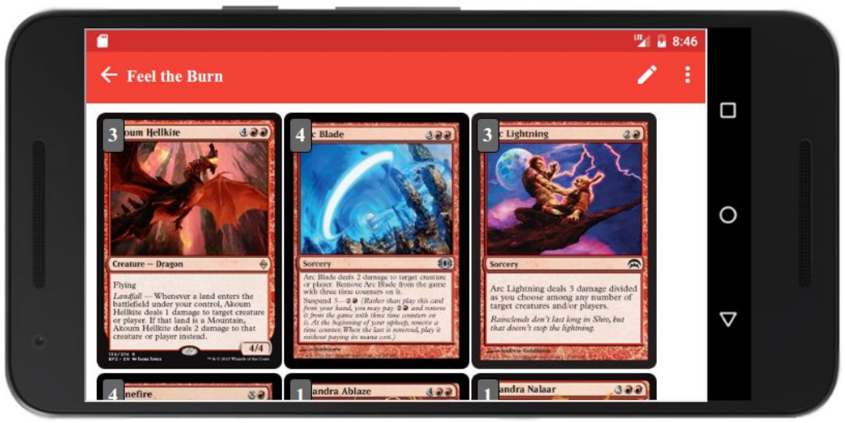
Same as the previous screen, but in landscape mode

Screen 4 - Deck Details (Phone - Portrait)



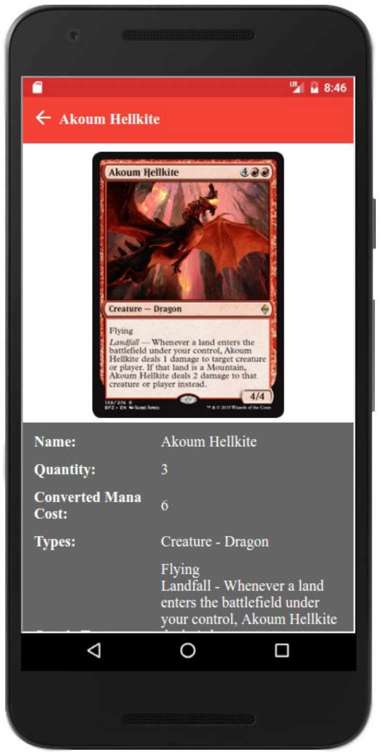
This screen shows a collection of cards contained in the deck, arranged in alphabetical order. Each card has a counter positioned at the top left corner of that card to indicate how many copies of that card are in the deck. This screen also has an up arrow to return to the deck list screen, the deck name, a crayon icon to edit the deck, and a more options icon which will allow the user to delete the deck from his/her collection.

Screen 4 - Deck Details (Phone - Landscape)



Similar to the previous screen, but in landscape mode. In this mode the screen width is larger so we have room for an extra column of cards.

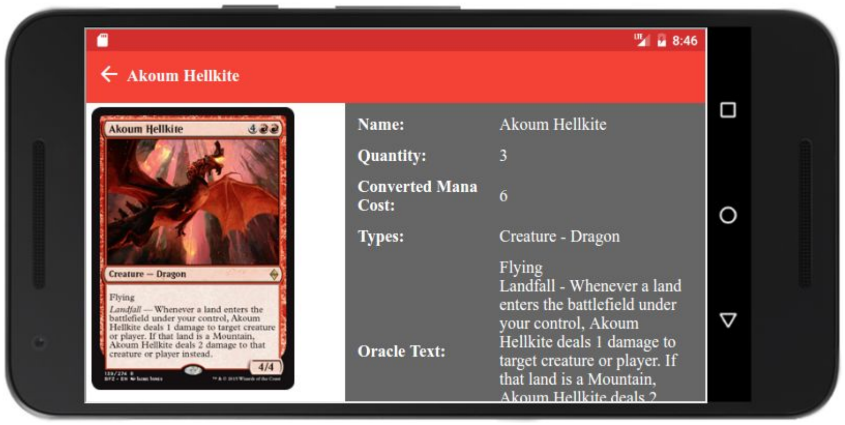
Screen 5 - Card Details (Phone - Portrait)



Name:	Akoum Hellkite
Quantity:	3
Converted Mana Cost:	6
Types:	Creature - Dragon
	Flying Landfall - Whenever a land enters the battlefield under your control, Akoum Hellkite

If a user clicks on a card they will be presented with a card detail screen. This screen contains an up arrow to bring the user back to the deck details screen, the card name as well as some information about the card (e.x. quantity contained in the deck, the cards converted mana cost, the card types, the name of the artist, etc.) which might otherwise be hard to read from just looking at the card image.

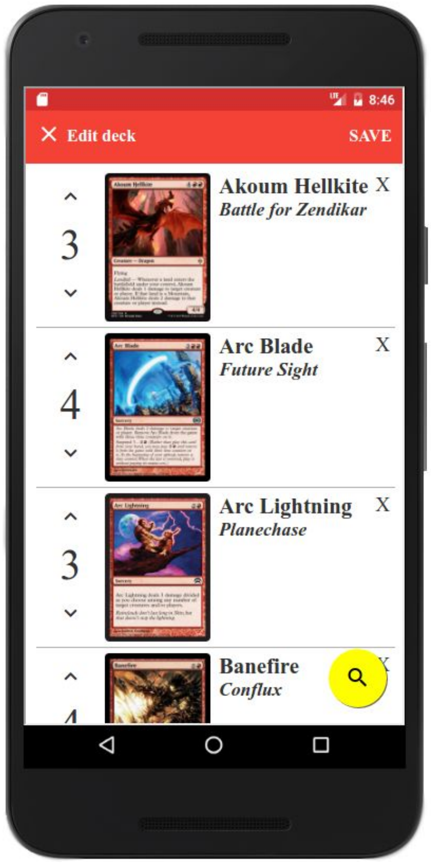
Screen 5 - Card Details (Phone - Landscape)



Name:	Akoum Hellkite
Quantity:	3
Converted Mana Cost:	6
Types:	Creature - Dragon
Oracle Text:	Flying Landfall - Whenever a land enters the battlefield under your control, Akoum Hellkite deals 1 damage to target creature or player. If that land is a Mountain, Akoum Hellkite deals 2

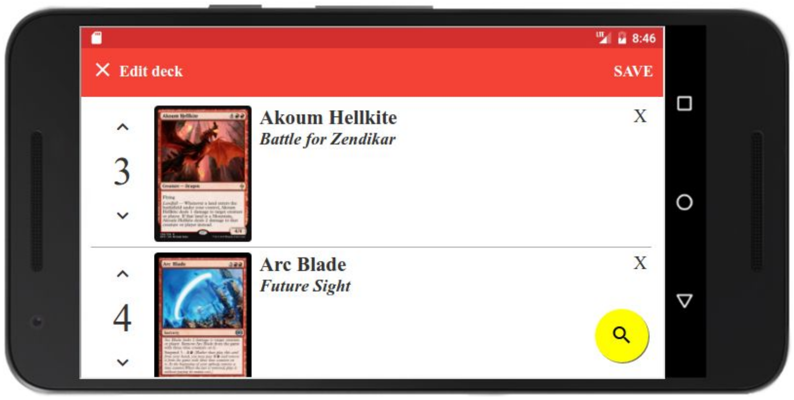
Similar to the previous screen, but in landscape mode. In this mode we instead display the card image on the left and the card info on the right.

Screen 6 - Edit Deck (Phone - Portrait)



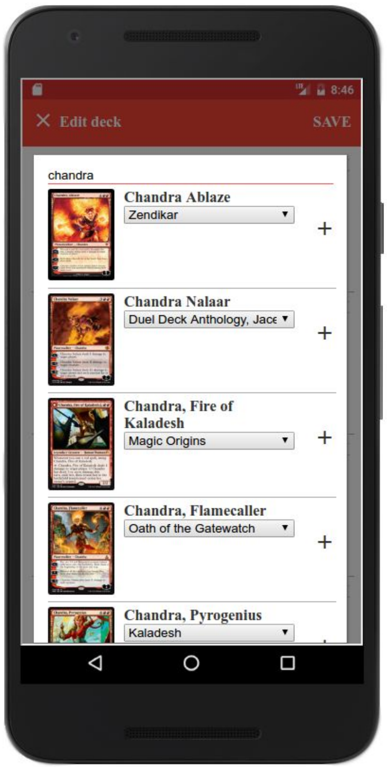
If the user clicks on the crayon icon to edit the deck they are presented with this screen. From this screen, the user can click on the X in the toolbar to cancel any changes that were made, click on SAVE to save any changes that were made, or click on the FAB to pop up the card search dialog. Additionally, for each card in the deck, the user is presented with an interface to remove all copies of a given card from the deck (X), increment the number of copies of a given card (up arrow), or decrement the number of copies of a given card (down arrow)

Screen 6 - Edit Deck (Phone - Landscape)



Same as the previous screen, but in landscape mode

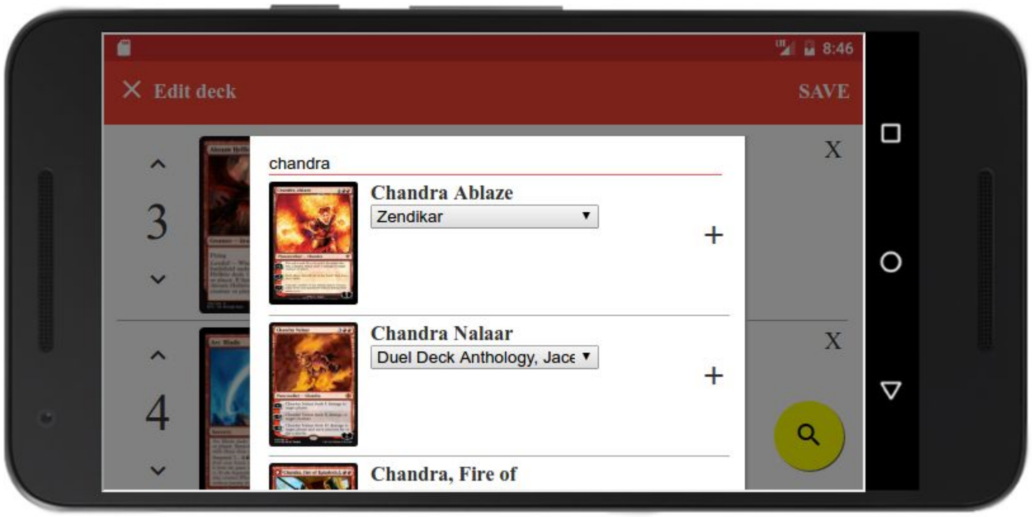
Screen 7 - Edit Deck with Search Card dialog (Phone - Portrait)



If the user clicks on the FAB (magnifying glass) they will be presented with this dialog. This card search dialog has an input field to allow the user to enter a card name (or partial card name) to search for. Each time the value of the input changes, a new search is made to find cards that match (unless the user is typing really fast, then we will wait for the input to "stabilize", for example waiting 500ms after the input changes before executing the search). Additionally, for each card found, a spinner is presented to the user to allow them to choose a different expansion for a given card (for example, the user might be interested in adding the card named

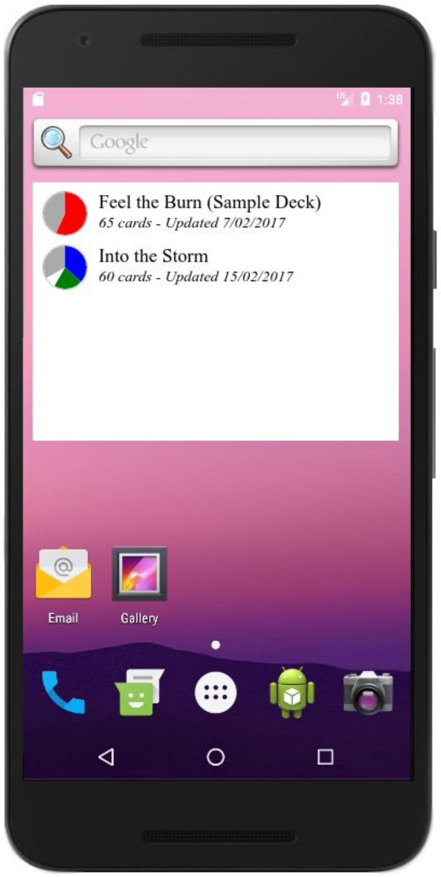
"Chandra Nalaar" to their deck but they may want to select the version from the "Magic 2011" expansion as opposed to the version from the "Duel Deck Anthology, Jace vs Chandra" expansion), and a + sign is shown to allow the user to add that card to their deck.

Screen 7 - Edit Deck with Search Card dialog (Phone - Landscape)



Same as the previous screen, but in landscape mode

Screen 8 - Deck List Widget (Phone)

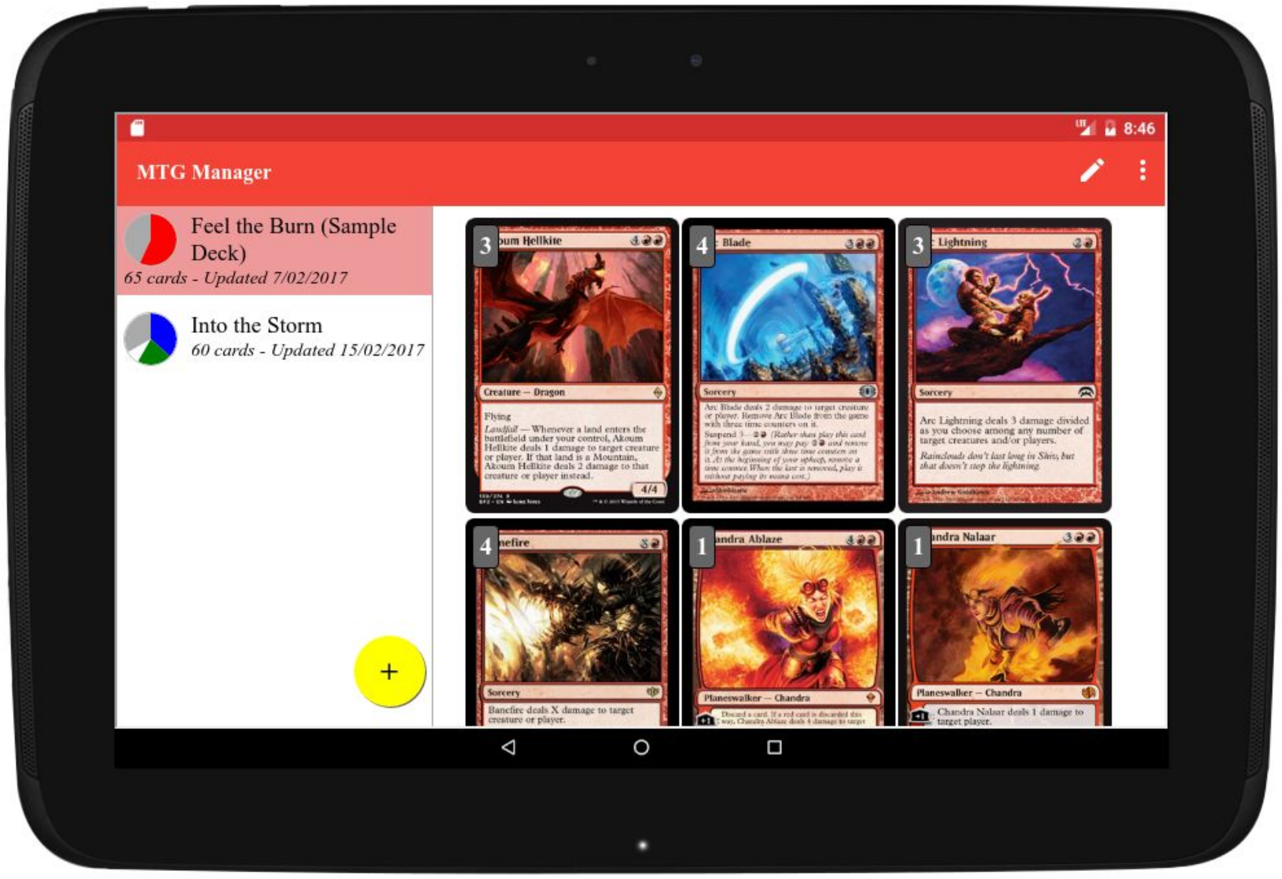


Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017

Into the Storm
60 cards - Updated 15/02/2017

This screen shows the Deck List Widget.

Screen 1 - Master/Detail (Tablet - Landscape)

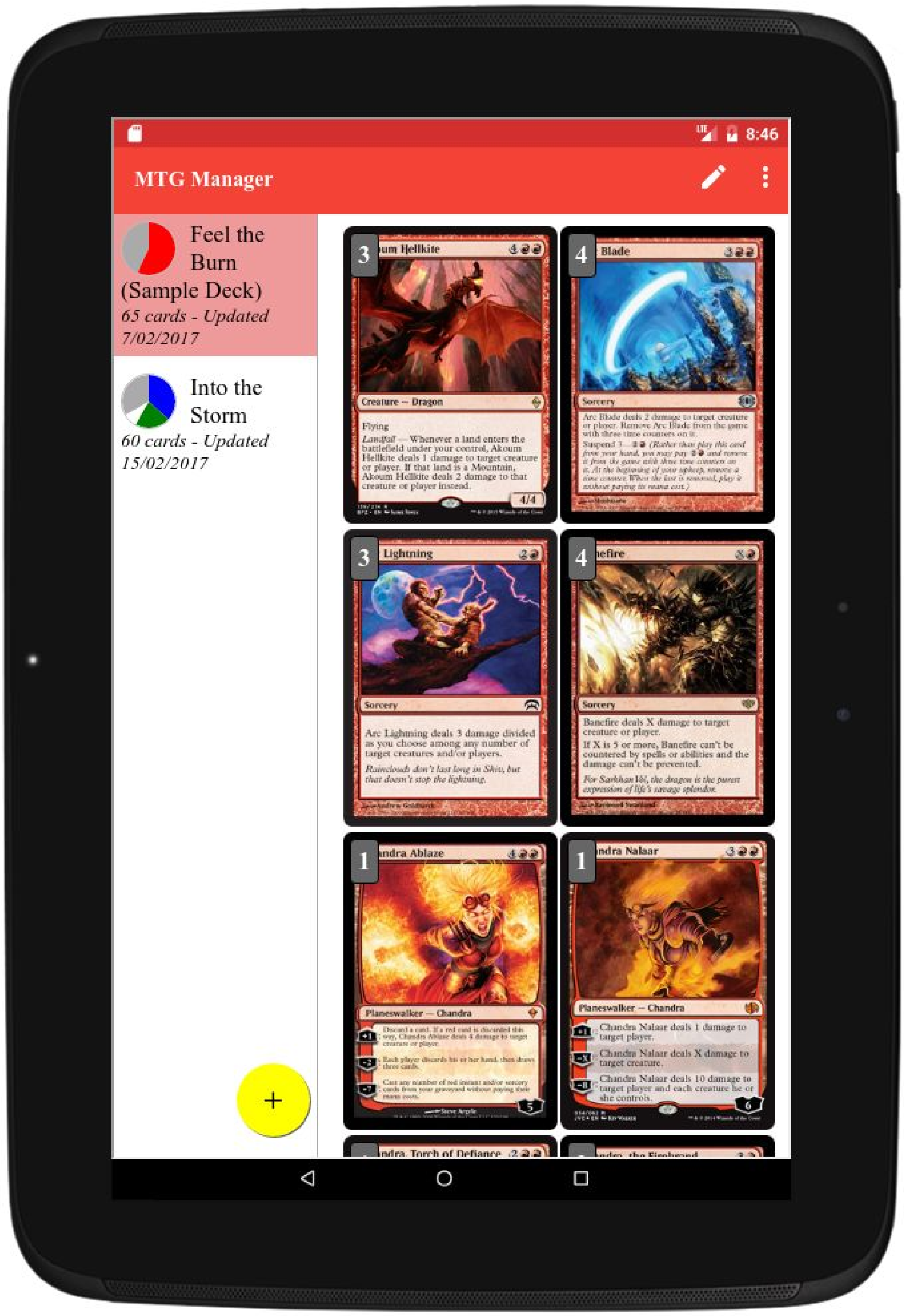


Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017

Into the Storm
60 cards - Updated 15/02/2017

This screen combines the Decks List screen and the Deck Details screen from the phone UI in a master/detail pattern since we have more space on a tablet.

Screen 1 - Master/Detail (Tablet - Portrait)

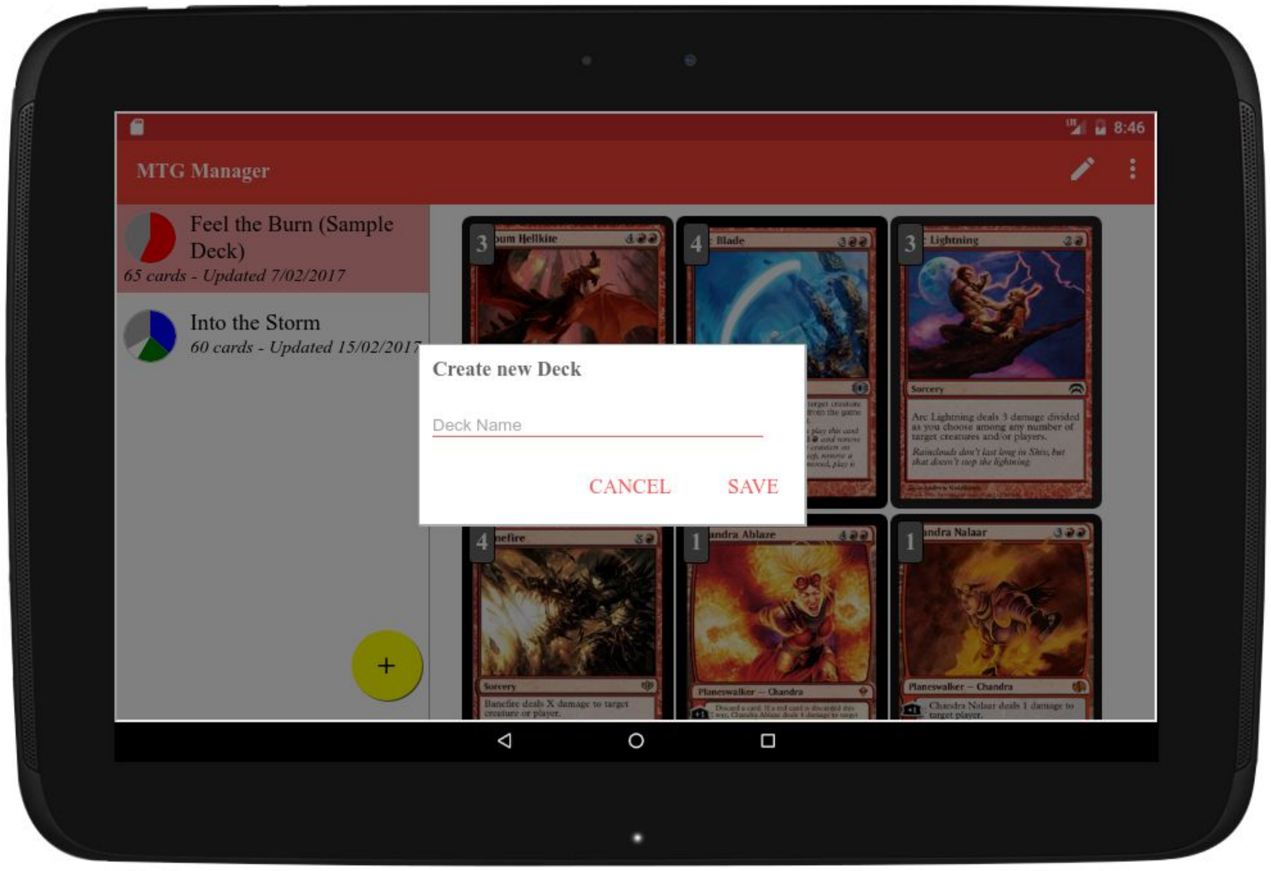


Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017

Into the Storm
60 cards - Updated 15/02/2017

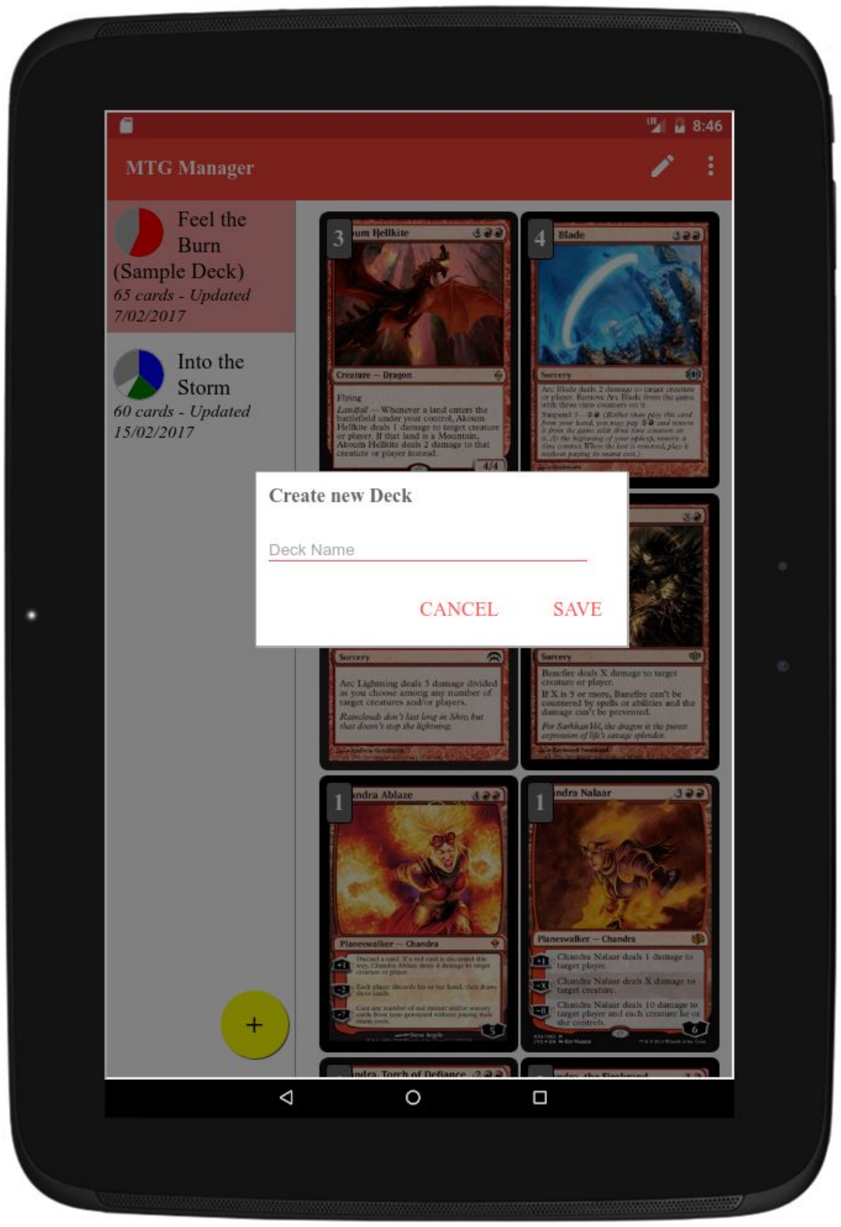
Similar to the previous screen, but in portrait mode. In this mode since we have less horizontal space, we can shrink the space given to the deck list and show one fewer column of cards.

Screen 2 - Master/Detail with Create new Deck dialog (Tablet - Landscape)


8:46
MTG Manager
Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017
Into the Storm
60 cards - Updated 15/02/2017
Create new Deck
Deck Name
CANCEL
SAVE
+

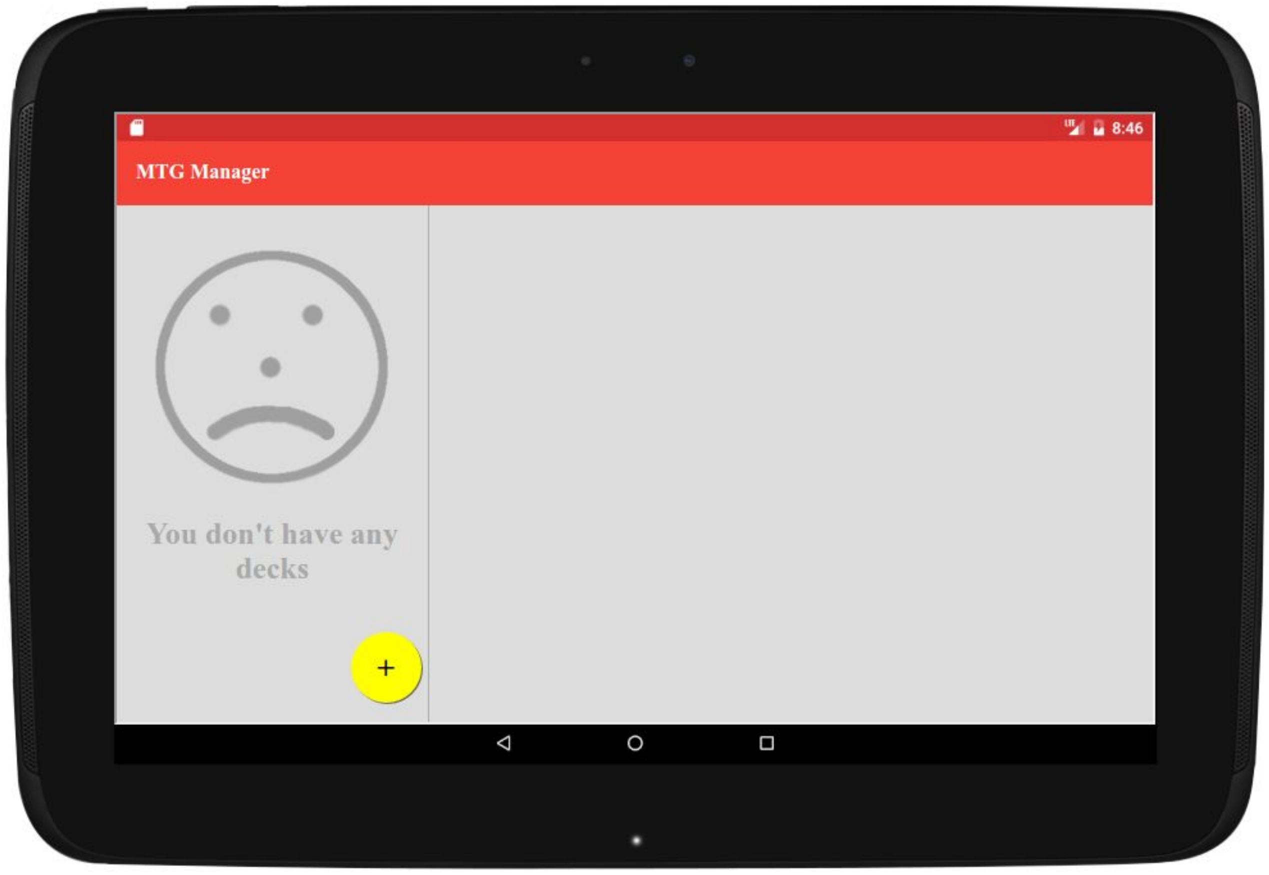
When a user clicks on the FAB, a dialog will pop up to allow the user to create a new deck, similar to the phone interface.

Screen 2 - Master/Detail with Create new Deck dialog (Tablet - Portrait)


8:46
MTG Manager
Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017
Into the Storm
60 cards - Updated 15/02/2017
Create new Deck
Deck Name
CANCEL
SAVE

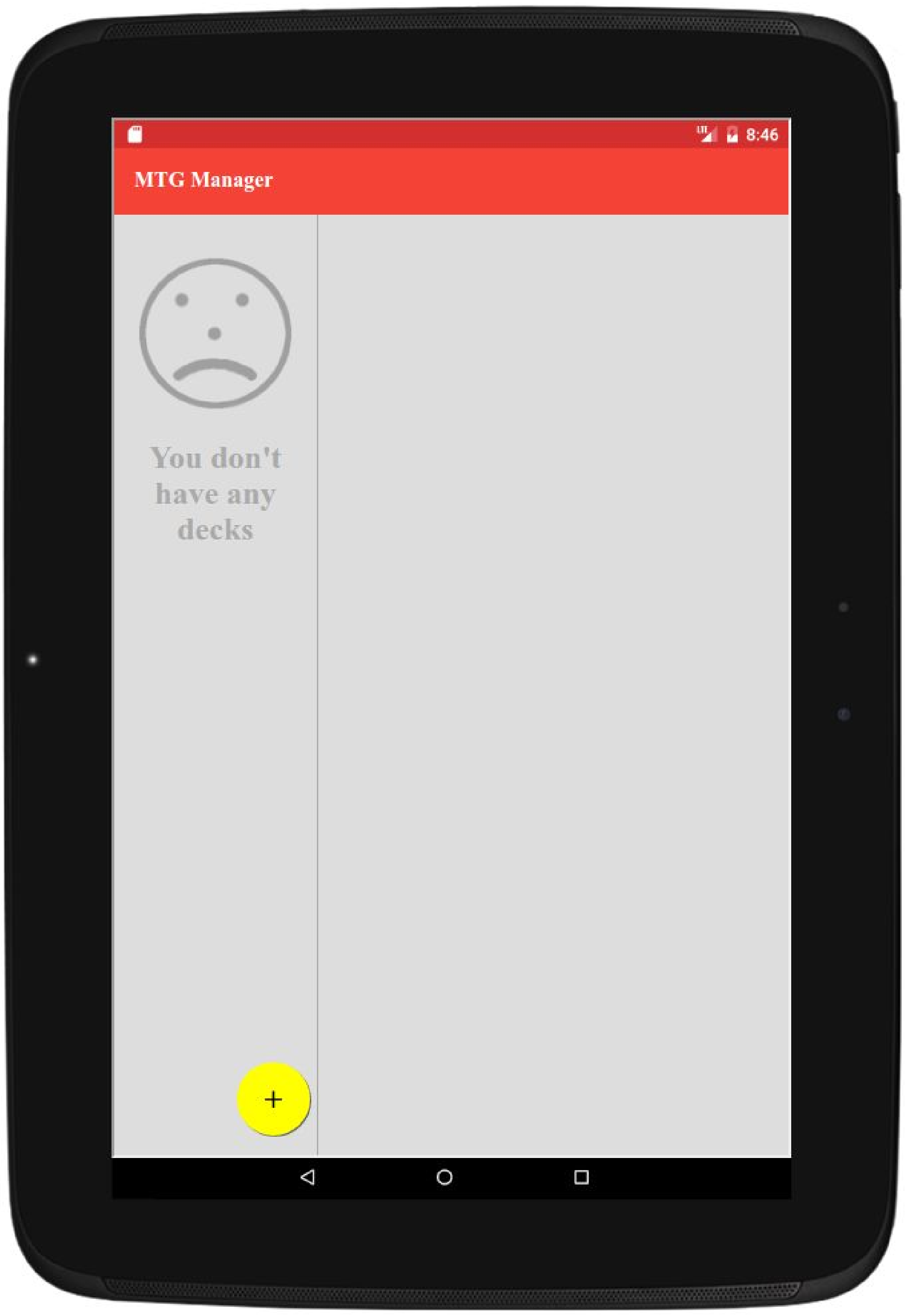
Same as the previous screen, but in portrait mode

Screen 3 - Empty Deck List (Tablet - Landscape)



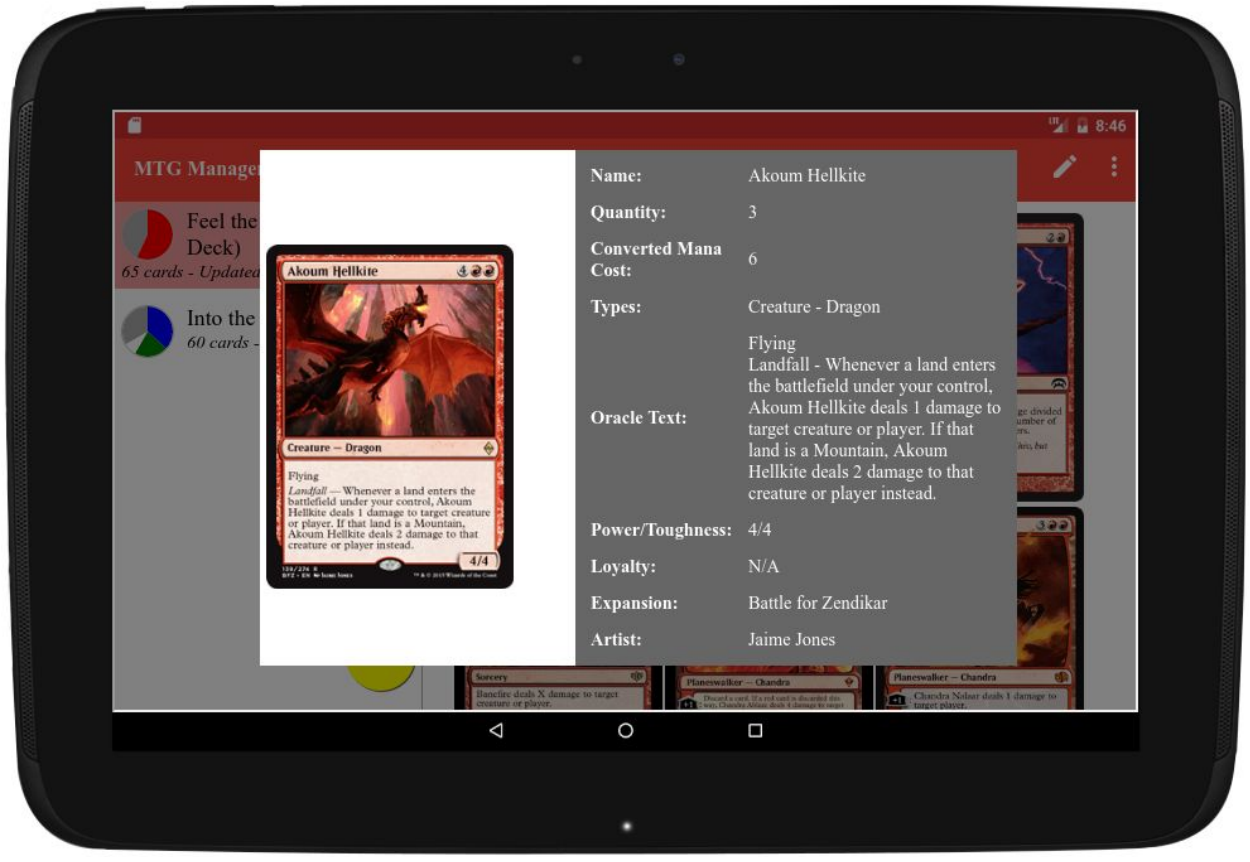
This screen is shown to the user if they deleted all their decks. One thing to notice is that since there are no decks, the icon to edit a deck and the option to delete a deck are hidden.

Screen 3 - Empty Deck List (Tablet - Portrait)



Same as the previous screen, but in portrait mode

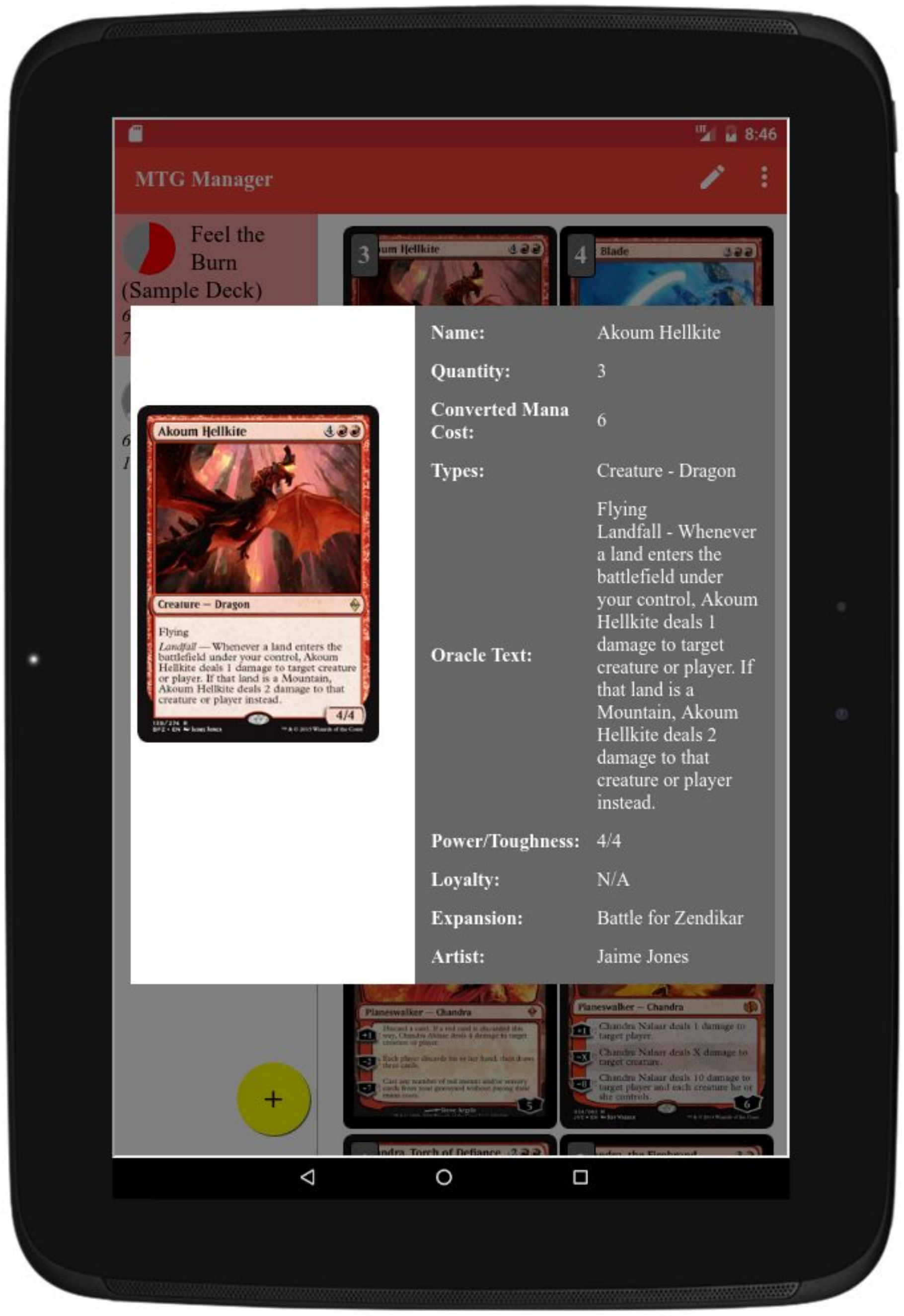
Screen 4 - Card Details (Tablet - Landscape)



Name:	Akoum Hellkite
Quantity:	3
Converted Mana Cost:	6
Types:	Creature - Dragon
Oracle Text:	Flying Landfall - Whenever a land enters the battlefield under your control, Akoum Hellkite deals 1 damage to target creature or player. If that land is a Mountain, Akoum Hellkite deals 2 damage to that creature or player instead.
Power/Toughness:	4/4
Loyalty:	N/A
Expansion:	Battle for Zendikar
Artist:	Jaime Jones

This screen is similar to the card details screen on the phone except that the card details are shown in a dialog on top of the master/details screen instead of being shown in a stand alone screen.

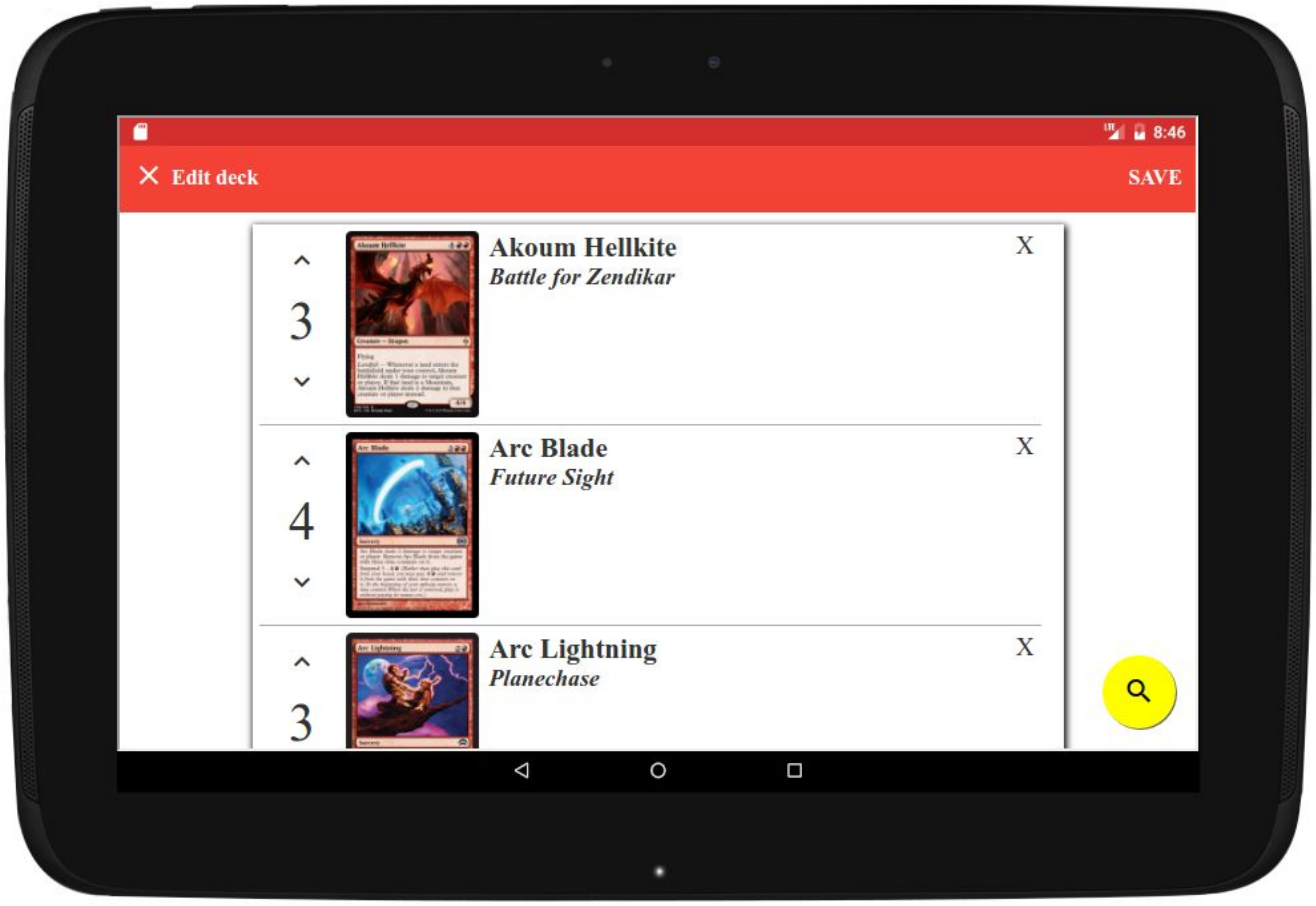
Screen 4 - Card Details (Tablet - Portrait)



Name:	Akoum Hellkite
Quantity:	3
Converted Mana Cost:	6
Types:	Creature - Dragon
Oracle Text:	Flying Landfall - Whenever a land enters the battlefield under your control, Akoum Hellkite deals 1 damage to target creature or player. If that land is a Mountain, Akoum Hellkite deals 2 damage to that creature or player instead.
Power/Toughness:	4/4
Loyalty:	N/A
Expansion:	Battle for Zendikar
Artist:	Jaime Jones

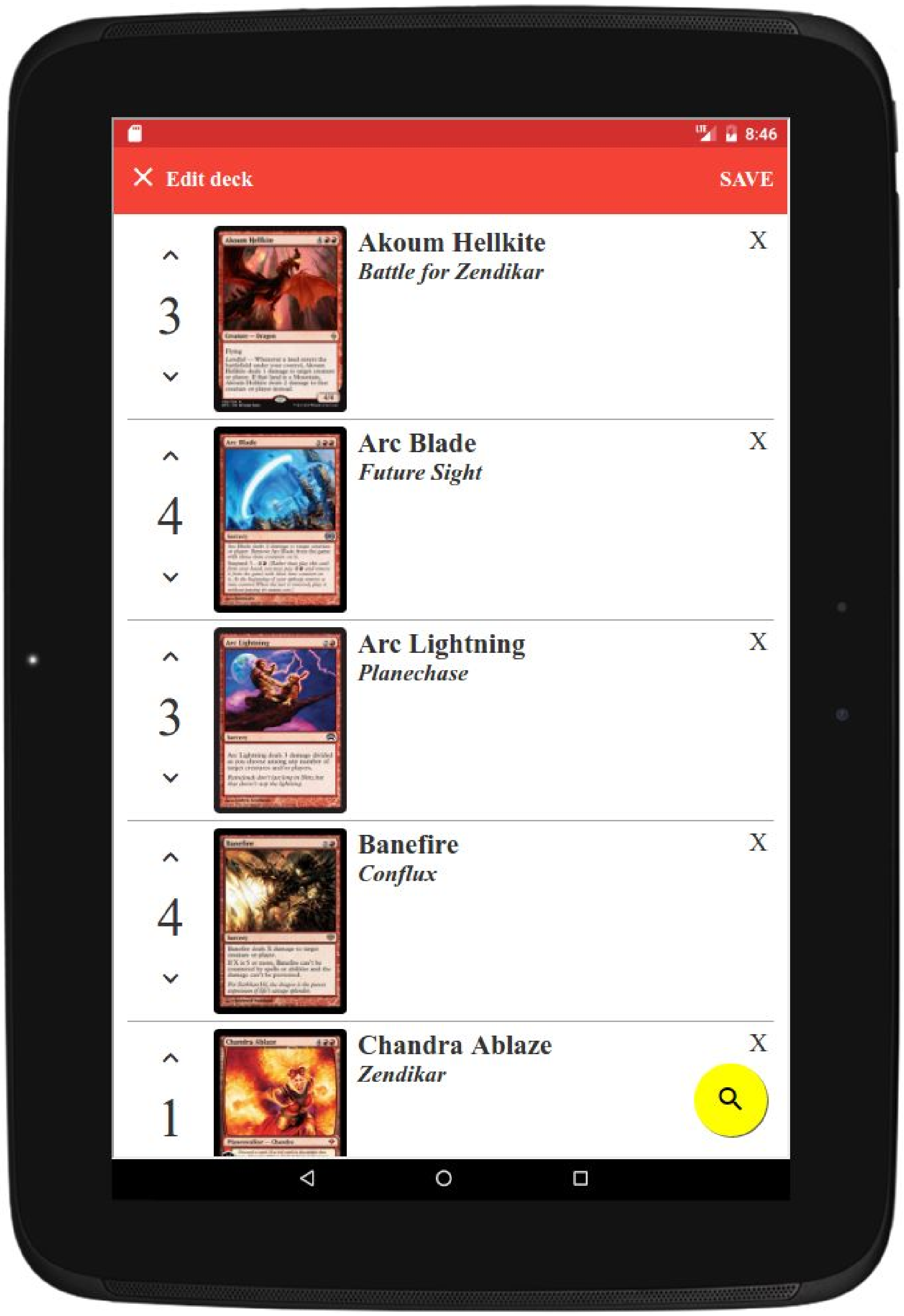
Same as the previous screen, but in portrait mode

Screen 5 - Edit Deck (Tablet - Landscape)



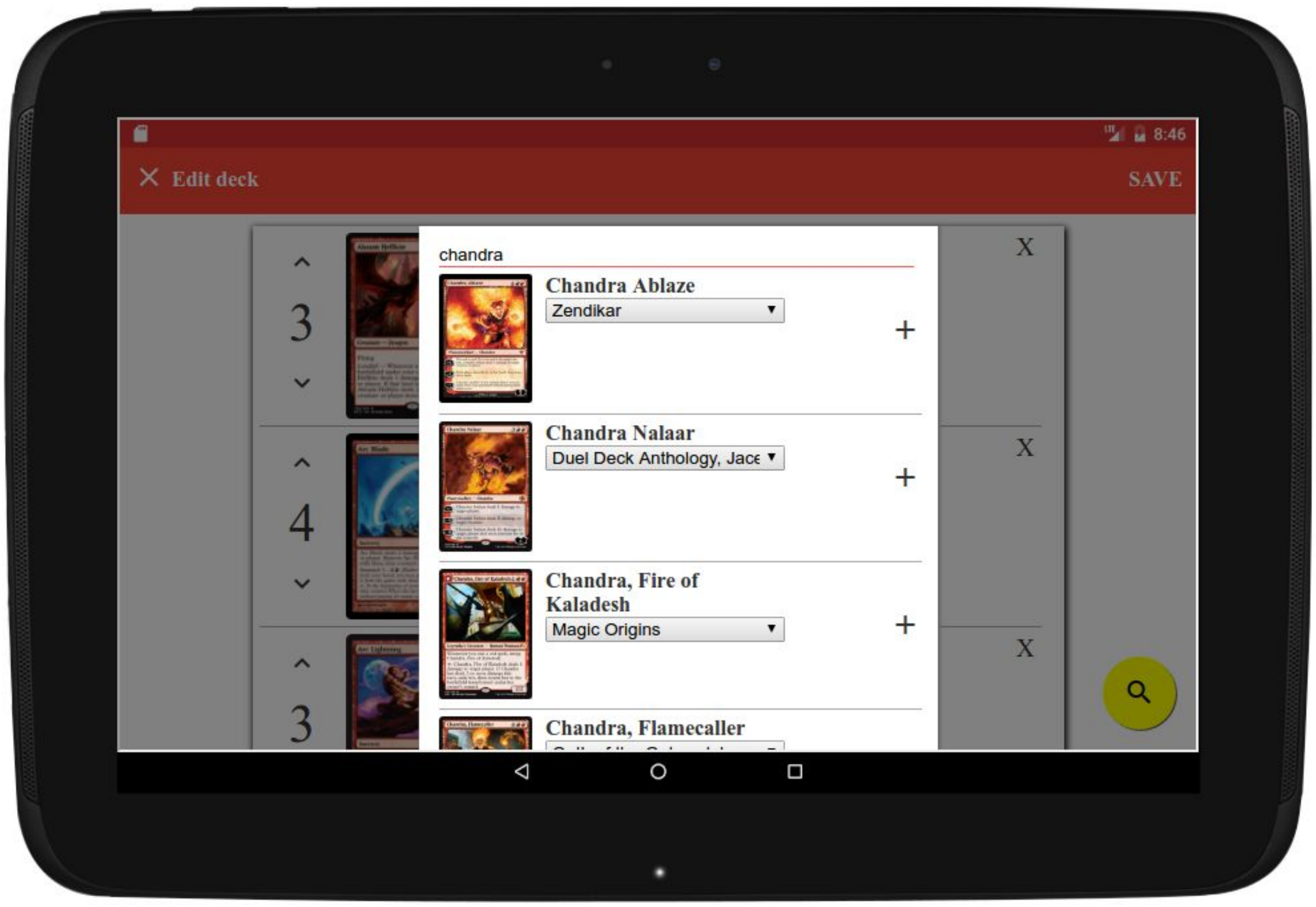
This screen is similar to the Edit Deck screen on the phone except that the card list is shown on its own surface, and the margins are bigger so that we don't end up with a horizontally stretched out UI.

Screen 5 - Edit Deck (Tablet - Portrait)



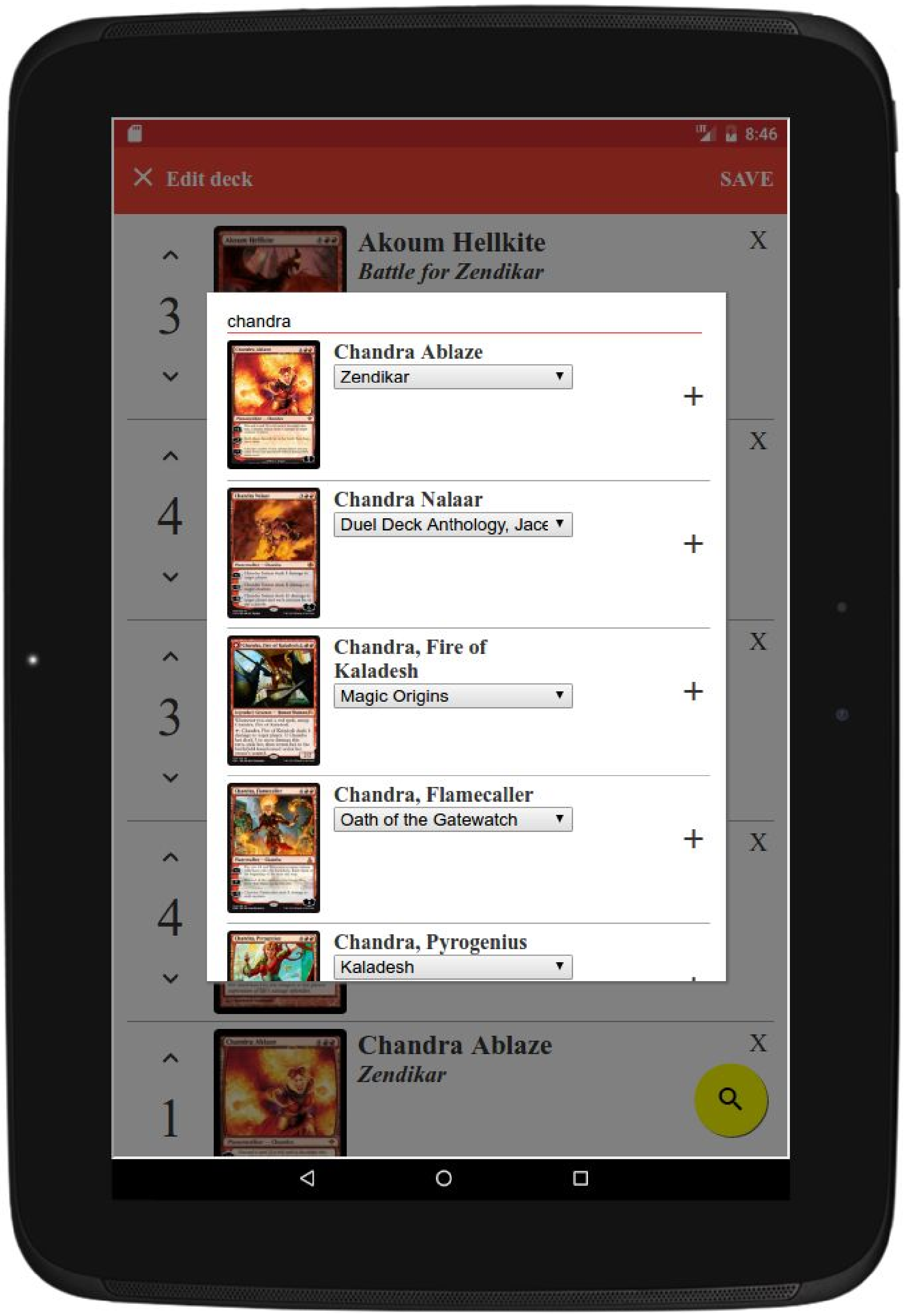
Similar to the previous screen, but in portrait mode. In this mode since we have less horizontal space, we don't need the card list to be on its own surface and we can have smaller margins.

Screen 6 - Edit Deck with Search Card dialog (Tablet - Landscape)



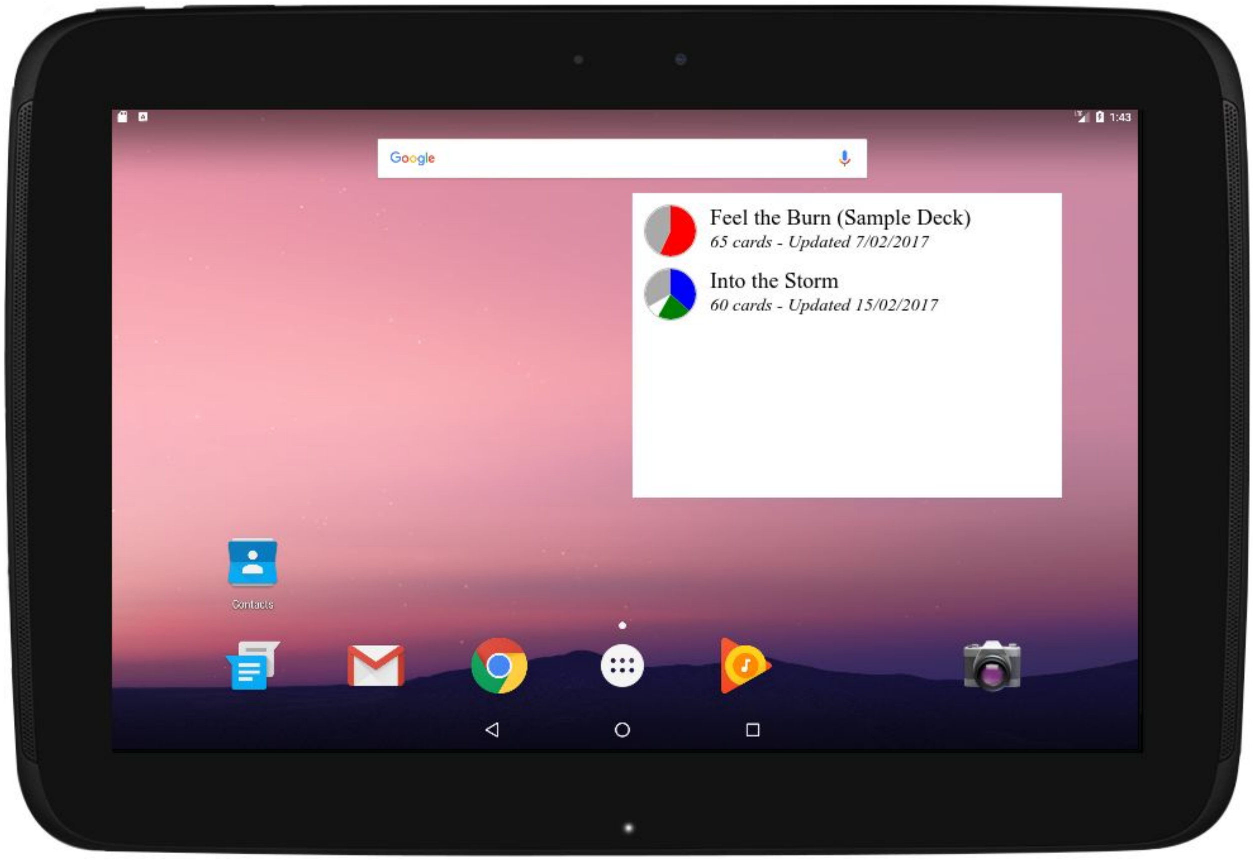
The search dialog on this screen is similar to the one on the phone.

Screen 6 - Edit Deck with Search Card dialog (Tablet - Portrait)



Same as the previous screen, but in portrait mode

Screen 7 - Deck List Widget (Tablet - Landscape)

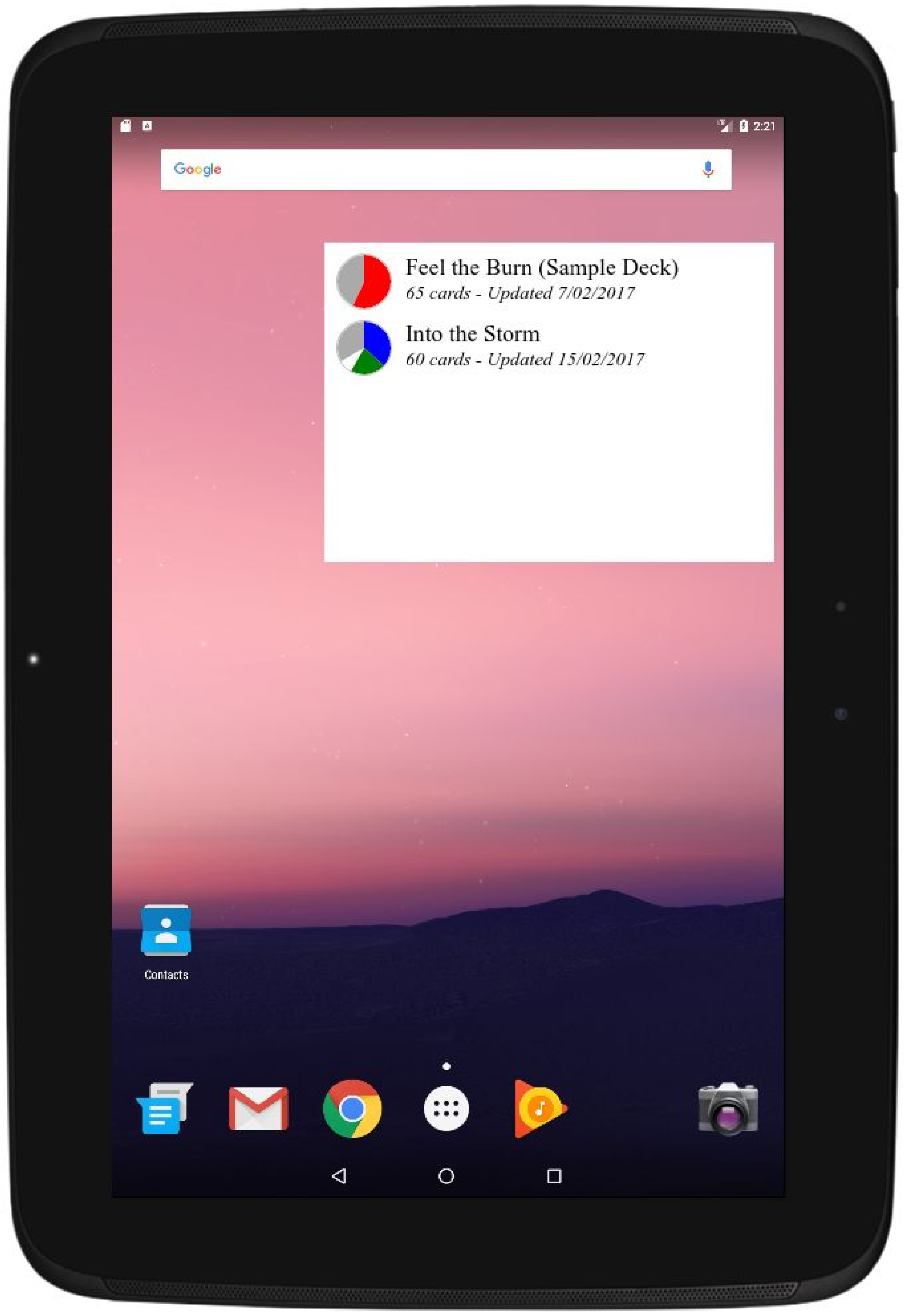


Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017

Into the Storm
60 cards - Updated 15/02/2017

This screen shows the Deck List Widget.

Screen 7 - Deck List Widget (Tablet - Portrait)



Feel the Burn (Sample Deck)
65 cards - Updated 7/02/2017

Into the Storm
60 cards - Updated 15/02/2017

Same as the previous screen, but in portrait mode

Key Considerations

How will your app handle data persistence?

For data persistence I intend to use a Content Provider, more specifically the FirebaseInitProver, as I plan to store the user created data in the cloud with the help of Firebase.

Describe any corner cases in the UX.

This app has 4 important corner cases in the UX

- If the user deletes all decks then the main screen would display information to the user to let them know that they don't have any decks, therefore making it clearer what needs to be done in order to see something on the main screen.
- If the user tries to create a deck with an invalid name (e.x. a name that already exists or an empty name) then a Toast would alert the user that the given name cannot be empty or that it already exists.
- If the user loses Internet access and they try to search for a card, then the UI would alert the user (possibly through the use of a Toast) that search is not available because of the lack of Internet access.
- If the user searches for a card name that does not return any results then the UI would alert the user that no results were found either via a Toast or by displaying a String in the results View to indicate that no results were found.

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

Magic: The Gathering SDK - Java (https://github.com/MagicTheGathering/mtg-sdk-java) to allow the ability to search for various MTG cards. This library communicates with the http://magicthegathering.io/ APIs.

Glide (https://github.com/bumptech/glide) to handle the loading and caching of images.

FirebaseUI (https://github.com/firebase/FirebaseUI-Android) to handle the authentication for Firebase

Describe how you will implement Google Play Services.

Google Play Services (AdMob) will be used to display an interstitial ad in the "free" version of the app.

Google Play Services (Firebase) will be used to store the user's decks so that they can be synchronised across devices.

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

Create a new Android Studio project and in your build.gradle file add the dependencies for the following libraries:

- Magic the Gathering library (https://github.com/MagicTheGathering/mtg-sdk-java)
- Glide image loading library (https://github.com/bumptech/glide)
- Firebase (https://firebase.google.com/docs/android/setup)
- Firebase UI authentication (https://firebase.google.com/docs/auth/android/google-signin)

In your build.gradle file, configure the productFlavors section to create a "free" and a "paid" version. Once that is done, you can AdMob as a freeCompile dependency (https://firebase.google.com/docs/admob/android/quick-start)

Task 2: Create Sample Deck

Create a class (for example Deck.class) to represent a set of Cards* contained in a deck. The Deck class should contain a deck name, set of Cards, and last updated date.
* The type Card is the one that is generated from the Magic the Gathering library

Using the Deck class, you can create the sample deck "Feel the Burn". This will be useful to help build the UI and to help the user understand how the app works.

The sample deck should contain the following cards:

- 3x Akoum Hellkite (multiverse id: 401805)
- 4x Arc Blade (multiverse id: 126193)
- 3x Arc Lightning (multiverse id: 205386)
- 4x Banefire (multiverse id: 186613)
- 1x Chandra Ablaze (multiverse id: 195402)
- 1x Chandra Nalaar (multiverse id: 393821)
- 1x Chandra, Torch of Defiance (multiverse id: 417683)
- 2x Chandra, the Firebrand (multiverse id: 259205)

- 2x Conflagrate (multiverse id: 114909)
- 4x Fireball (multiverse id: 191076)
- 2x Hammer of Bogardan (multiverse id: 3452)
- 4x Incinerate (multiverse id: 134751)
- 2x Increasing Vengeance (multiverse id: 262661)
- 4x Maze of Ith (multiverse id: 1824)
- 12x Mountain (multiverse id: 191401)
- 3x Nevinyrral's Disk (multiverse id: 420882)
- 4x Urza's Mine (multiverse id: 220948)
- 4x Urza's Power Plant (multiverse id: 220952)
- 4x Urza's Tower (multiverse id: 220956)

Note that the multiverse id, which is used on Wizards gatherer (e.x. http://gatherer.wizards.com/Pages/Card/Details.aspx?multiverseid=220948), is what should be used as the unique identifier for a card. Two cards can have the same name, but if they are from different sets they will have different multiverse ids.

Task 3: Implement UI for Each Activity and Fragment

Build UI for Deck List Fragment

- Build UI for Deck color breakdown pie chart
- Build UI for Deck List item
- Build UI for FAB
- Build UI for empty list of decks

Build UI for Deck Details Fragments

- Build UI for individual card
- Build UI to hold all the cards, using a GridLayout

Build UI for Master/Detail on tablet

- Build a UI that contains 2 Fragments (Deck List Fragment + Deck Details Fragment)

Build UI for Card Details

- Build UI for Card Details on phone (portrait)
- Build UI for Card Details on phone (landscape)
- Build UI for Card Details on tablet, showing as a dialog

Build UI for Edit Deck Activity

- Build UI for individual card
- Build UI for list of cards to edit (phone)
- Build UI for list of cards to edit (tablet - landscape)
- Build UI for FAB

Build UI for search

- Build UI for search card field
- Build UI for no cards found matching the given name
- Build UI for a found card

While implementing the UI, make sure to follow best practices for Material Design, and use the following colors from the Material Design guidelines:

- colorPrimary #F44336 (red - 500), used for the toolbar
- colorPrimaryDark #D32F2F (darker red - 700), used for the system bar
- colorAccent #FFFF00 (bright yellow - A200), used for FAB
- otherColorAccent #FF5252 (light red - A200), used for highlighting form fields and button text, since all the nice looking yellows didn't offer enough contrast for text in combination with the white background, which is not good for accessibility

Task 4: Create a Paid version and a Free version

In the "free" src directory of the app, create a Deck List Fragment which loads an interstitial ad before showing the Deck Details screen (or updating the Deck Details Fragment on tablets). The "paid" version of the Deck List Fragment should not show any ads before showing the Deck Details.

Task 5: Implement card search functionality

Implement the card search functionality by making a request to the magicthegathering.io API (using the included Magic the Gathering library specified earlier).

- Add the Internet permission and the Access Network State permission
- Monitor the input field for changes, and when the value does change (and isn't empty), send a request to the magicthegathering.io API
- Make sure the request to the API is executed on a different Thread (either through the use of an Async Task or an IntentService)
- Filter out any Cards that do not have a multiverse id (as we are not interested in them)
- Display the Cards in the list

As of March 1st 2017, a search by card name to the magicthegathering.io API (once filtered to remove cards without a multiverse id), when searching for the name "chandra" should contain at least the following cards:

Chandra Ablaze - Zendikar
Chandra Nalaar - Lorwyn
Chandra Nalaar - Magic 2010
Chandra Nalaar - Magic 2011
Chandra Nalaar - Duel Decks Anthology, Jace vs. Chandra
Chandra Nalaar - Duel Decks: Jace vs. Chandra

Chandra’s Fury - Magic Origins
Chandra’s Fury - Magic 2013
Chandra’s Ignition - Magic Origins
Chandra’s Outrage - Magic 2012
Chandra’s Outrage - Magic 2011
Chandra’s Outrage - Archenemy
Chandra’s Outrage - Magic 2014 Core Set
Chandra’s Phoenix - Magic 2014 Core Set
Chandra’s Phoenix - Magic 2012
Chandra’s Pyrohelix - Kaladesh
Chandra’s Revolution - Aether Revolt
Chandra’s Spitfire - Magic 2011
Chandra, Fire of Kaladesh - Magic Origins
Chandra, Flamecaller - Oath of the Gatewatch
Chandra, Pyrogenius - Kaladesh
Chandra, Pyromaster - Magic 2014 Core Set
Chandra, Pyromaster - Magic 2015 Core Set
Chandra, Roaring Flame - Magic Origins
Chandra, Torch of Defiance - Kaladesh
Chandra, the Firebrand - Magic 2013
Chandra, the Firebrand - Magic 2012
Oath of Chandra - Oath of the Gatewatch

Task 6: Add Firebase support

Go to the Firebase website to setup a Firebase project for your app (https://firebase.google.com/), a Google account is required. Additional instructions are found here -> https://firebase.google.com/docs/android/setup

Once your app has been linked to your Firebase project, follow the FirebaseUI for Android - Auth instructions to add authentication for your app. (https://github.com/firebase/FirebaseUI-Android/tree/master/auth)

Now that Firebase authentication has been added modify other parts of the app that require Firebase, including the following:

- When creating a new Deck, add it to Firebase
- When deleting a Deck, delete from Firebase
- Get the list of Decks from Firebase
- Get the list of Cards in a Deck from Firebase
- When adding, removing, or updating the number of copies of a given Card to a Deck, make the same changes to Firebase.
- Implement a ContentProvider that works with Firebase

Task 7: Implement List of Decks widget

Follow the Android App Widgets documentation to implement an app widget with collections to show the user's list of Decks.
(https://developer.android.com/guide/topics/appwidgets/index.html)

- Declare App Widget in the Manifest
- Add the AppWidgetProviderInfo Metadata
- Create the App Widget Layout
- Create a Class that extends AppWidgetProvider
- Implement an App Widget with Collections, by making use of the RemoteViewsService, and a RemoteViewsFactory

---

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"