
GitHub Username: gundamboy

Collectable Card Game Cube Draft Simulator

Description

This app lets you simulate a Cube Draft from a very popular collectable card game, Magic The Gathering. A cube draft is a casual Magic: The Gathering format where players create a cube, a large pool of cards selected for the purposes of playing a limited game. To prepare for this format, a player (or, if you prefer, your entire play group) prepares a "cube" — a specifically selected set of at least 360 different cards. Once the cube has been built, you can use it for any draft format. The most popular option is to build makeshift "booster packs" out of 15 randomly selected cards from the cube and then run a regular Booster Draft.
** information from https://mtg.gamepedia.com/Cube_Draft **

Other stuff:

- The app will keep all strings in the strings.xml file. While I am at it, all the dimens and other stuff will be in their files too because i'm sure if I don't put that you will fail this thing again.
- The app will be written in Java\.
- The enables RTL layout. Yeah, thats a given.

- The app includes support for accessibility including content descriptions, d-pad navigation, and non-audio versions of audio cues. The app won't have audio.
- "If it needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), the app uses an IntentService to do so." When the user searches for a card, the api is called. When the user wants to keep the card, its stored.
- "If it performs a short duration, on-demand requests(such as search), the app uses an AsyncTask." yes, it will do this..
- "If Room is used then LiveData and ViewModel are used when required and no unnecessary calls to the database are made." I already said Room would be used and you have to use the LiveData and ViewModel down below. Obviously you have to use LiveData and ViewModel with that.

Intended User

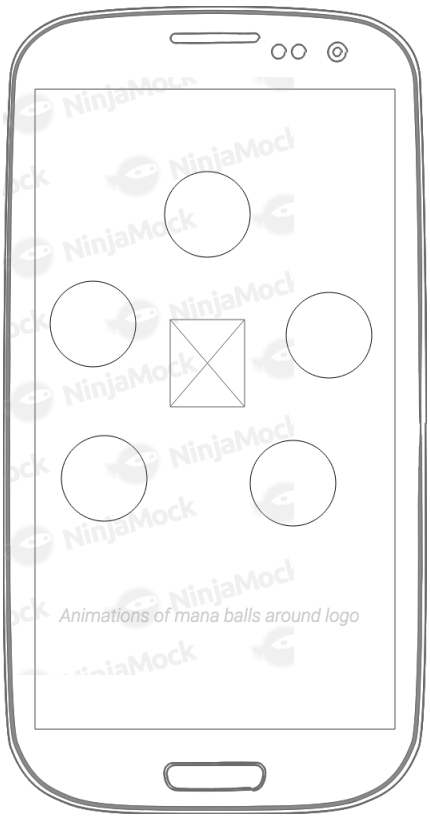
Anyone that plays Magic The Gathering and is interested in Cube Drafting

Features

List the main features of your app:

- App simulates a cube draft (one cube is a selection of 360 cards or more).
- App allows user to import a cube?
- Allows user to store card cube on your device or in the cloud (firebase cloud fire store).
- Generate booster packs from your cube for later draft format (not in this app). One 360 card cube supports 8 players, this is 3 booster packs per player. One booster pack is 15 cards.
- Review card cube.
- View each generated booster pack

User Interface Mocks



splash screen

login activity

registration activity

my cubes activity. If the user has cubes, they can review them here.

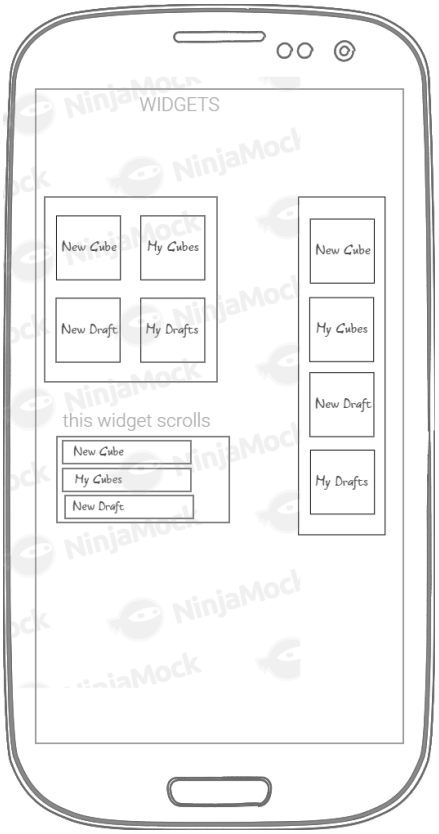
my drafts. it the use has drafted before, they can review them here.

new cube activity. start a new cube. the rectangles are cards. the circles indicate either selected or not selected. the user has filter options. this screen is also used to review the individual cards in an existing cube and existing draft cards (it’s the same layout).

new draft screen. the rectangles are the card "packs". each draft creates 3 packs of 15 cards. the user "opens" the pack to see the cards.

the way a draft works is you have 3 packs, and multiple players. player 1 picks a card from pack one, then that pack is passed around to all the players that each pick a card. next they do the same with pack 2, then pack 3 until there are no more cards. the player then makes a 40 card deck of cards.

reviewing the packs is the same screen the new cube layout



App widgets.

Key Considerations

How will your app handle data persistence?

The app will allow the user to register an account. The user will have the option to store their card lists locally, or in the cloud (firestore), or both. Accounts will be stored on firestore. Shared Preferences for some user preferences/settings.

Describe any edge or corner cases in the UX.

User will have access to a back navigation at all times.

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

*all libraries will be the newest versions. They will not be alpha or beta with the exception of Google libraries like Androidx, which is the approved way of doing things now by Google. The library versions here are subject to change because the developers can do an update at any time. *

Picasso (2.71828) will handle the loading and caching of images.
Retrofit (v2.5.0) will be used for api calls. Gson is used in conjunction with this.
Butterknife (v10.1.0) will be used for view binding.
Moshi (v1.8.0) will be used for creating POJOs from the json.
Timber (v4.7.1) will be used for logging.
Other libraries are a possibility, i won't know until I get started really.

Describe how you will implement Google Play Services or other external services.
Firestore will be used for accounts and cube lists.
Auth for allowing account sign up using your google account.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Set app up for use with androidx
- Configure libraries / gradle
- Convert styles.xml to use material design

Task 2: Implement UI for Each Activity and Fragment

- Build UI for SplashActivity
- Build UI for RegistrationActivity
- Build UI for MainActivity
- Build UI for NewCubeActivity
- Build UI for MyCubesActivity
- Build UI for DraftActivity
- Build additional UIs as needed

Task 3: Databases

I will setup the databases for use

- Set up the Room Database
- Add dummy info to database
- Setup Firestore Database
- Add dummy data to Firestore

Task 4: MainActivity

- List of user functions to allow navigation to various other activities

Task 5: New user registration and login

- Java class for registrations
- Setup basic registration process
- Setup Auth service for google account registration method
- Setup login functionality
- Unit test registration success and login

Task 6: New Cube

- Setup retorifit for calling the cards api
- Unit test the retrofit api call
- Display the data
- Setup basic functions for saving card, naming the cube, saving finished cube, etc

Task 7: My Cubes

Display a list of user created cubes

- Show the list
- Set up cube editing

Task 8: Draft Activity

- Setup new draft
- Generate booster packs from chosen cube
- Setup booster pack review list

Task 9: Splash Screen

- Setup splash screen activity / animations

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"