Design and implementation of a deck-building application for Magic: The Gathering

Brewer's Servant

Bachelor Degree in Informatics Engineering
Software Engineering Specialization

Author: David Garcia De Mercado
Director: Silvia Llorente Viejo

April 19, 2022

Abstract

The motivation behind the development of this application is that *Magic: The Gathering* has a growing player base that is starting to play the game, and construct decks to play the game. The game is growing and becoming more complex, which makes resource applications a must-have to design decks to play with.

This project consists of a market analysis and a design and implementation of a deck building application for *Magic: The Gathering.* The application allows users to search for cards with multiple filtering options, and card combos as well. Decks can be built so the user has an all-in-one application that fits their needs.

An *Android* application has been developed using *Kotlin* and integrating multiple external libraries, as well as using an external API. The different layers of the application have been separated to facilitate the expansion of the project in the future.

Keywords *Magic: The Gathering*, *Android*, Application, Cards, Decks, Design.

Resumen

La motivación detrás del desarrollo de esta aplicación es que *Magic: The Gathering* tiene una base de jugadores creciente que está empezando a jugar, y construyen barajas para jugar al juego. El juego está creciendo en complejidad, lo que convierte las aplicaciones de recursos en imprescindibles para diseñar barajas con las que jugar.

Este proyecto consiste en un análisis de mercado, y un diseño e implementación de una aplicación de construcción de barajas para *Magic: The Gathering.* La aplicación permite a los usuarios buscar cartas usando múltiples filtros, y combos de cartas también. Se pueden construir barajas, y de esta forma el usuario tiene una aplicación todo-en-uno que se ajusta a sus necesidades.

Una aplicación de *Android* ha sido desarrollada en *Kotlin* e integra librerías externas, además de usar una API externa. Las distintas capas de la aplicación se han separado para facilitar la expansión del proyecto en el futuro.

Palabras clave *Magic: The Gathering*, *Android*, Aplicación, Cartas, Barajas, Diseño.

Resum

La motivació darrere el desenvolupament de l'aplicació és que *Magic: The Gathering* té una base de jugadors creixent que comença a jugar, i construeixen baralles per a jugar al joc. El joc creix en complexitat, el que converteix les aplicacions de recursos en imprescindibles per a dissenyar baralles amb les quals jugar.

Aquest projecte consisteix en una anàlisi de mercat, i un disseny i implementació d'una aplicació de construcció de baralles per a *Magic: The Gathering.* L'aplicació permet als usuaris cercar cartes utilitzant múltiples filtres, i combos de cartes també. Es poden construir baralles, i d'aquesta manera l'usuari té una aplicació tot-en-un que s'ajusta a les seves necessitats.

Una aplicació d'*Android* ha estat desenvolupada en *Kotlin* i integra llibreries externes, a més d'usar una API externa. Les diferents capes de l'aplicació han estat separades per a facilitar l'expansió del projecte en el futur.

Paraules clau *Magic: The Gathering*, *Android*, Aplicació, Cartes, Baralles, Disseny.

Contents

1 Introduction **8**
1.1 Magic: The Gathering . . . 8
1.1.1 What is Magic: The Gathering . . . 8
1.1.2 Formats . . . 9
1.2 Deck Building . . . 9
1.2.1 The Need for Deck Building Tools . . . 9
1.3 Stakeholders . . . 10

2 Market Analysis **11**
2.1 Resource Applications . . . 11
2.1.1 MTG Assist . . . 11
2.1.2 Scryfall . . . 12
2.1.3 Commander Spellbook . . . 14
2.1.4 EDHREC . . . 14
2.2 Deck Building Apps . . . 17
2.2.1 TappedOut . . . 17
2.2.2 Moxfield . . . 17
2.2.3 Archidekt . . . 17
2.2.4 ManaBox . . . 21
2.2.5 TopDecked MTG . . . 21
2.3 Conclusion . . . 21

3 Scope of the Project **22**
3.1 Objectives . . . 22
3.2 Requirements . . . 23
3.3 Risks . . . 24

4 Methodology **25**
4.1 Project Management . . . 25
4.2 Software Management . . . 26
4.3 Technologies . . . 26
4.3.1 Additional Technologies . . . 27
4.4 Methodology Changes . . . 27
4.5 Knowledge Application . . . 27

5 Temporal Planning **29**
5.1 Tasks Definition . . . 29
5.1.1 Project Management . . . 29
5.1.2 Development . . . 30

5.1.3 Documentation/Communication . . . 32
5.2 Task Summary . . . 32
5.3 Gantt Diagram . . . 34
5.4 Planning Changes . . . 35

6 Risk Management **38**
6.1 Technology inexperience . . . 38
6.2 Closed calendar . . . 38
6.3 Bugs . . . 39
6.4 Obstacles . . . 39

7 Budget **41**
7.1 Identification of Costs . . . 41
7.1.1 Costs by Task . . . 41
7.1.2 General Costs . . . 43
7.1.3 Contingency Costs . . . 43
7.1.4 Risk Management Costs . . . 43
7.1.5 Final Budget Estimate . . . 43
7.2 Budget Management . . . 44
7.3 Budget Changes . . . 45

8 Sustainability **46**
8.1 Social Sustainability . . . 46
8.2 Economic Sustainability . . . 46
8.3 Environmental . . . 47

9 Laws and Regulations **48**

10 Architecture **49**
10.1 General Structure . . . 49
10.2 Class Diagram . . . 50
10.3 Use Cases . . . 51
10.3.1 Sequence Diagrams . . . 55
10.4 Interface Design . . . 57
10.4.1 Application Flow . . . 62
10.5 Database Design . . . 63

11 Project Development **66**
11.1 Implementation . . . 66
11.1.1 Card Search . . . 66
11.2 Combo Search . . . 69
11.3 Work-arounds . . . 69
11.3.1 Database Primary Keys . . . 69
11.3.2 Combo Database . . . 70

12 Conclusions **71**
12.1 Personal Conclusions . . . 71
12.2 Future Improvements . . . 72
12.2.1 User Interface . . . 72

12.2.2 Functionalities and Features 72
12.2.3 Optimisation and Code Refactoring 73

Glossary **74**

Acronyms **75**

Bibliography **76**

List of Figures

2.1 MTG Assist home page. 12
2.2 *Scryfall* advanced search, deck builder and deck viewer page. 13
2.3 *Commander Spellbook* interface captures. 15
2.4 *EDHREC* interface captures. 16
2.5 *TappedOut* interface captures. 18
2.6 *Moxfield* interface captures. 19
2.7 *Archidekt* interface captures. 20

5.1 Gantt diagram. 34
5.2 The revised Gantt diagram for the project 36

10.1 Layered architecture diagram. 50
10.2 UML Class Diagram for the project's application. 51
10.3 Main use cases for the application. 52
10.4 Search for cards sequence diagram. 56
10.5 Search for combos sequence diagram. 57
10.6 Card Search related views. 58
10.7 Combo Search related views. 59
10.8 Card Details view. 60
10.9 Combo Details and Card Download views. 61
10.10 Deck related views. 62
10.11 Cards section flow diagram. 62
10.12 Combos section flow diagram. 62
10.13 Decks section flow diagram. 63
10.14 Database Diagram for the project's application. 64

11.1 Card Search screen. 67
11.2 Combo Search view. 69

List of Tables

5.1 Task Summary Table 33

6.1 Risk probability and estimated resolution time. 38

7.1 Hourly salary average per employee role on the project. 41
7.2 Task Cost Summary Table 42
7.3 Budget management sheet 1. 44
7.4 Budget management sheet 2. 44

Chapter 1

Introduction

This project is my bachelor's degree thesis. In this project I will use all the knowledge obtained during my degree to design and implement a mobile application for constructing Magic: The Gathering decks.

This document is a report of all the processes and analysis performed during the making of the project, as well as the setbacks encountered and how I managed to work around them. Its chapters will contain different parts of the project, such as the scope definition, or the budget calculation for the project.

Before jumping into the project itself some concepts need to be explained to the reader. This chapter will introduce the main concepts of the game Magic: The Gathering needed to understand the project's objective, as well as the concept of deck building.

1.1 Magic: The Gathering

This section will explain what is Magic: The Gathering as a game, not as a brand. I will not explain the rules in depth (you can read more about the rules of the game and how to play in the *How to Play Magic: The Gathering* website [1]). The different formats of the game [2] will also be introduced with the objective of explaining on the following sections what deck building is and justifying why deck building applications are needed to facilitate and improve this process.

1.1.1 What is Magic: The Gathering

Magic: The Gathering is a Trading Card Game (TCG) in which players use their own deck of cards to battle against another player. Each player starts with the same life total, which is reduced when taking damage, and the objective of the game is to use the cards in your deck to reduce the opponent's life total to 0. The first player to reduce the opponent's life total to 0 wins the game.

There are different card types that follow different rulings and each card has its own text describing what it does when played. There currently exist more than 20000 different Magic: The Gathering cards that can be played through all the different formats.

1.1.2 Formats

A format in Magic: The Gathering is a set of rules that defines a way to play the game. These rules can be very different between formats: the Modern format is a 1v1 constructed format with 60 card minimum decks and with a limit of 4 copies of each card per deck, and each player starts the game with 20 life, on the other hand, the Commander format is a multiplayer constructed format with exactly 100 card decks and with a limit of 1 copy of each card per deck, and each player starts with 40 life.

The two main types of formats are limited formats and constructed formats. In limited formats players open booster packs (a booster pack usually contains 15 random cards from a specific set within it) and use the cards in them to play, in constructed formats players design decks using a specific card pool (the legal card pool is defined by the format) to compete against other players.

Constructed Formats

Constructed formats are the formats I will take into account when designing this application. These are the formats that need deck building the most. Some of them have very vast card pools, which need advanced card search mechanisms to filter through and find the cards each player needs for their deck. Some formats such as Modern have a very competitive scenario and need advanced analysis tools to help the players optimise and test their decks.

1.2 Deck Building

Deck building is the action of designing a deck to play in a certain format and have a certain strategy or objective. Some decks are designed to be aggressive and lower the life total of the opponent as fast as possible while others aim to create the most fun experience for all the players involved in the game.

Due to the great number of different cards currently in the game, the need to optimise competitive decks and many other needs from many different kinds of players, deck building applications provide players with many tools to help them make decks that fit their needs.

1.2.1 The Need for Deck Building Tools

As introduced through all this chapter, deck building can be a very complex and long process. To find the best cards for each deck without using any tool would mean you would have to search through more than 20000 cards to be sure you found the best possible cards to play. You would also have to manually calculate the rates of cards you play to balance the deck's performance, and you would have much difficulty to track what cards are in your deck, what cards are possible additions, etc. These are some of the reasons why deck building applications are needed.

In the following chapter we will take a look at the most popular deck building applications available on the market. We will see what tools they offer and what tools they lack.

1.3 Stakeholders

Stakeholders are the different parties that have an implication or interest for the project. It can be a direct interest, such as paying for the project to be done, or an indirect interest as in a possible future client/user of the product.

Stakeholders are important because their opinion on the project matters. They may be looking for a specific functionality, or maybe they do not want the final product to have a certain image or reputation, and these needs might shape the final aspect of the product, or change the course of development so the product can be adapted to any necessities or desires.

Below you will find a list of the stakeholders implicated in this project.

- **Wizards Of The Coast** The company that owns Magic: The Gathering. They are implicated on anything that uses Magic: The Gathering and could damage or boost the game's popularity or image.
- **All other similar platforms** The owners of the competence will have another competitor on the market, and they might lose clients potentially.
- **Magic: The Gathering players** Any Magic: The Gathering player that looks for a deck building mobile application might find this product and end up being a user. Their needs are certainly the most important factor to consider. This is the main audience for the product.
- **Universitat Politècnica de Catalunya** This project is a thesis for my degree at UPC, so the university is implicated on it.
- **Project director** The director of the project, Silvia Llorente, guides me through the project and gives me guidelines to follow.
- **Myself** As the project developer I am the most implicated stakeholder.

Chapter 2

Market Analysis

In this chapter we will look at the most popular web applications and mobile applications that are used for deck building purposes, either being applications that let you construct a deck and save it or share it, applications that allow to search for cards in a more advanced manner than just a name search or applications that have both functionalities or even extra ones. Each application will be discussed and their strong and weak points will be listed.

This analysis will be useful to find missing functionalities in the apps and gather ideas for the project, so it can be an application with some unique and useful functionalities and/or that incorporates multiple functionalities that are not implemented together in any of the options of the market. Using the other applications will also possibly uncover design flaws that we can improve to make the user experience faster, more comfortable and/or more enjoyable. Another point to consider is that the final product of this project is a mobile application, and thus even though web applications do not directly compete with the project it can be very useful to compare the current mobile and web applications and find what functionalities are missing from one platform to another, which can be used to boost the product's utility for the users implementing any of the functionalities (if any) that are not yet available to phone users.

The existing applications can be divided in two main groups: deck building applications and resource applications. Deck building applications are applications that are focused on providing the user with a complete deck builder that can analyse the deck or help the user to analyse it and optimise it. Resource applications are applications that focus on advanced card searches or card recommendations, with the objective of helping the user find the best cards for their deck.

2.1 Resource Applications

2.1.1 MTG Assist

MTG Assist[3] is a card search web application. It supplies its users with an advanced card search engine that allows not only to search by name, but to also search by mana cost, color, power, toughness and many more. The search engine also allows the users to select ranges for some values, or to search for cards that specifically do or do not have a specific text.

By far the most distinctive functionality that this website offers is the *Similar* *to:* option. It allows the user to provide a specific card so the engine can search for cards similar to the one specified. It even can search for similar cards to the one selected that also have the specified characteristics in the other search fields.

Despite having a unique functionality on the market (the *Similar to:* functionality) this website is not very popular. This is because the other search options could be expanded to allow for more customization of search, and also because the website's interface is not very good. The interface is very clunky and is not attractive visually. Another negative point is that this site has a lot of advertisement banners that make the user experience less appealing because the advertisements take a lot of screen space (Figure 2.1). One last important thing to note is that it is not adapted for screens bigger than 1080p: When used in a 2k screen there is a lot of space on the right of the window left blank because the content does not extent to all the dimension of the window.

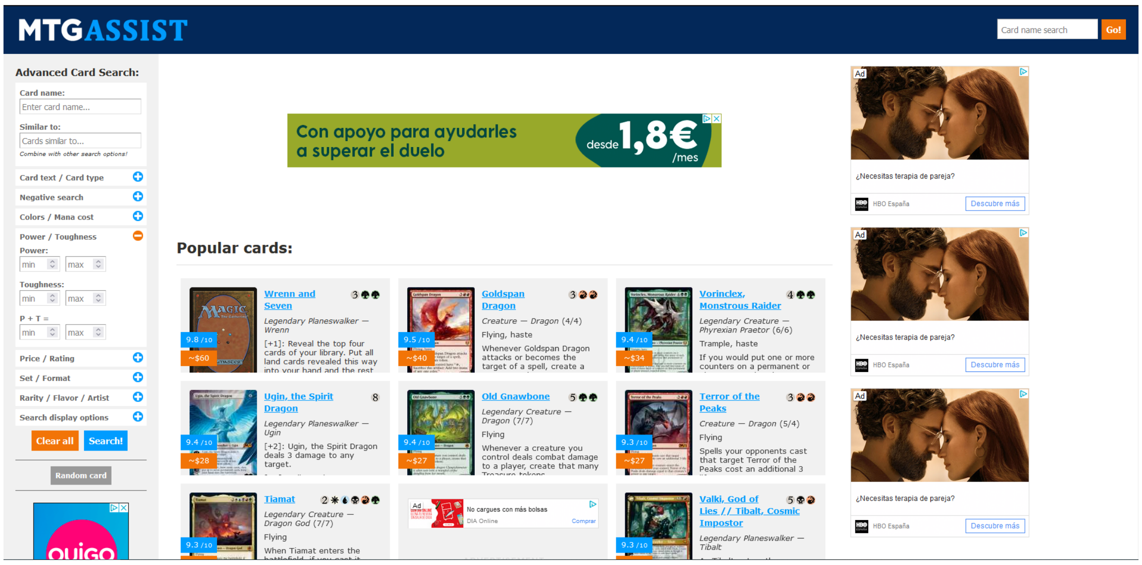


Figure 2.1: MTG Assist home page.

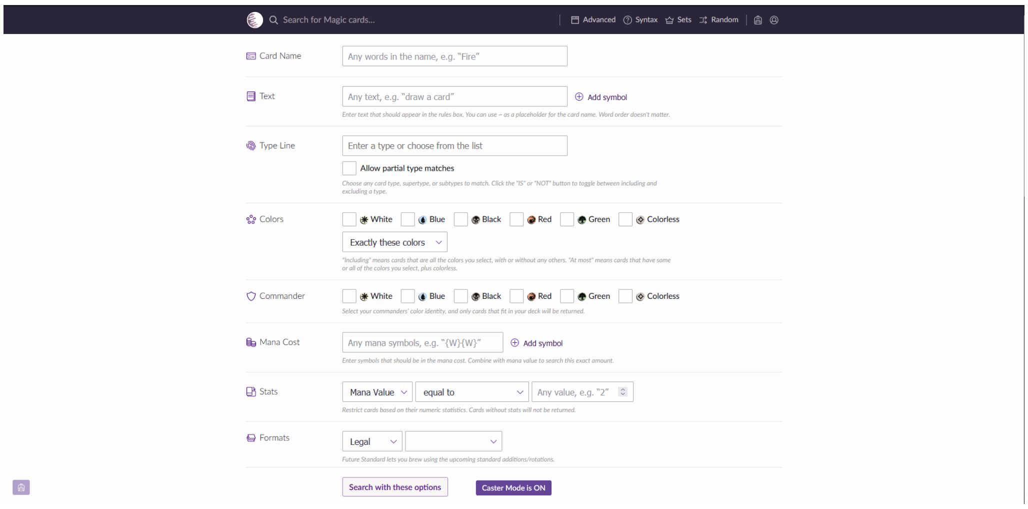
2.1.2 Scryfall

Scryfall[4] is a card search web application. Its search engine is the best on the market at the moment of writing: it is very extensive and clear to use, as well as having minimalist design which is pleasant to the eyes.

It is important to note that it does not do anything out of what you would expect from an advanced card search: it has no custom search options such as the *Similar to:* functionality from MTG Assist but, even though it has no "original" search options, it supplies its users with a myriad of options (which can be combined) and is very fast. You can see part of the advanced search page at Figure 2.2a.

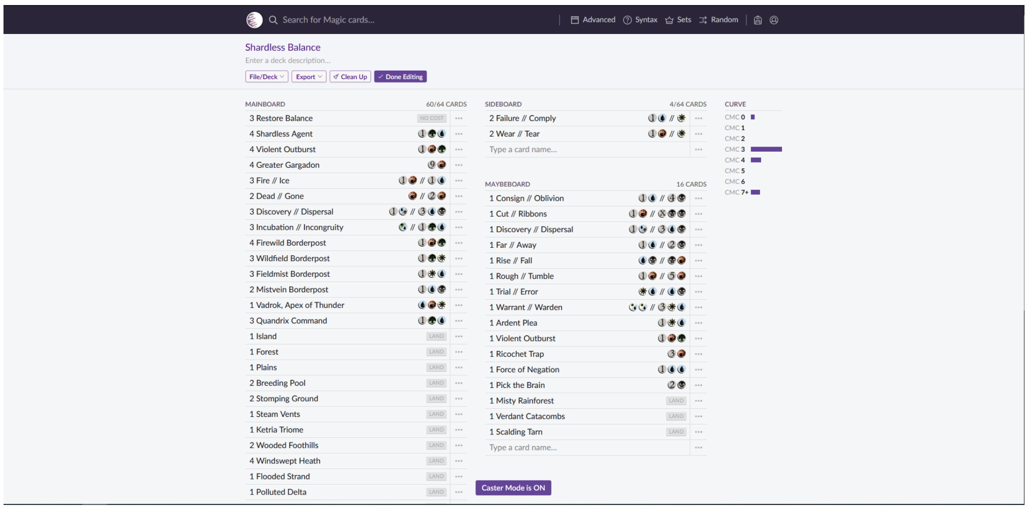
One thing we should comment on is the fact that *Scryfall* has a deck builder implemented. It is a very simple deck builder (it is also very recent), very fast to build decks on but with almost no customization, and almost no information about statistics in the user's deck. This makes it very difficult for the user to properly analyse their deck and try to make it better. It is useful as a deck list archive but not much more (see Figure 2.2b). Nevertheless, the page for viewing built decks

(Figure 2.2c is very good, can be customised (although not a lot) and has better organisation than the builder page.



Card Name	Any words in the name, e.g. "Fire"
Text	Any text, e.g. "draw a card" · Add symbol *Enter text that should appear in the rules box. You can use ~ as a placeholder for the card name. Word order doesn't matter.*
Type Line	Enter a type or choose from the list Allow partial type matches *Choose any card type, supertype, or subtypes to match. Click the "IS" or "NOT" button to toggle between including and excluding a type.*
Colors	White · Blue · Black · Red · Green · Colorless Exactly these colors *"Including" means cards that are all the colors you select, with or without any others. "At most" means cards that have some or all of the colors you select, plus colorless.*
Commander	White · Blue · Black · Red · Green · Colorless *Select your commanders' color identity, and only cards that fit in your deck will be returned.*
Mana Cost	Any mana symbols, e.g. "{W}{W}" · Add symbol *Enter symbols that should be in the mana cost. Combine with mana value to search this exact amount.*
Stats	Mana Value · equal to · Any value, e.g. "2" *Restrict cards based on their numeric statistics. Cards without stats will not be returned.*
Formats	Legal *Future Standard lets you brew using the upcoming standard additions/rotations.*

(a) Advanced Search (part of the total interface)



Shardless Balance
Enter a deck description...

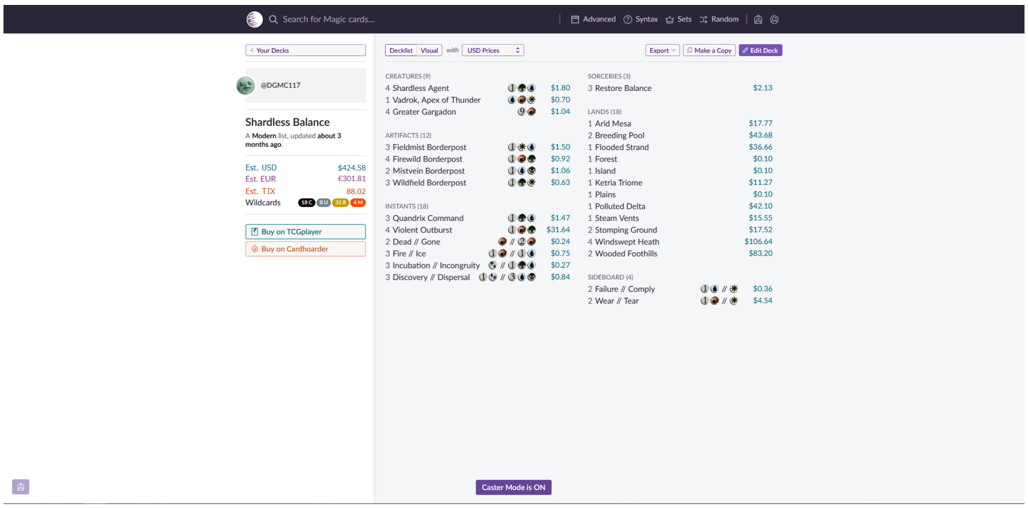
MAINBOARD	60/64 CARDS
3 Restore Balance	NO COST
4 Shardless Agent	
4 Violent Outburst	
4 Greater Gargadon	
3 Fire // Ice	
2 Dead // Gone	
3 Discovery // Dispersal	
3 Incubation // Incongruity	
4 Firewild Borderpost	
3 Wildfield Borderpost	
3 Fieldmist Borderpost	
2 Mistvein Borderpost	
1 Vadrok, Apex of Thunder	
3 Quandrix Command	
1 Island	LAND
1 Forest	LAND
1 Plains	LAND
2 Breeding Pool	LAND
2 Stomping Ground	LAND
1 Steam Vents	LAND
1 Ketria Triome	LAND
2 Wooded Foothills	LAND
4 Windswept Heath	LAND
1 Flooded Strand	LAND
1 Polluted Delta	LAND

SIDEBOARD	4/64 CARDS
2 Failure // Comply	
2 Wear // Tear	
Type a card name...	

MAYBEBOARD	16 CARDS
1 Consign // Oblivion	
1 Cut // Ribbons	
1 Discovery // Dispersal	
1 Far // Away	
1 Rise // Fall	
1 Rough // Tumble	
1 Trial // Error	
1 Warrant // Warden	
1 Ardent Plea	
1 Violent Outburst	
1 Ricochet Trap	
1 Force of Negation	
1 Pick the Brain	
1 Misty Rainforest	LAND
1 Verdant Catacombs	LAND
1 Scalding Tarn	LAND
Type a card name...	

CURVE
CMC 0
CMC 1
CMC 2
CMC 3
CMC 4
CMC 5
CMC 6
CMC 7+

(b) Deck builder



@DGMC117

Shardless Balance
A **Modern** list, updated **about 3 months ago**.

Est. USD	$424.58
Est. EUR	€301.81
Est. TIX	88.02
Wildcards	18 C · 8 U · 31 R · 4 M

CREATURES (9)	
4 Shardless Agent	$1.80
1 Vadrok, Apex of Thunder	$0.70
4 Greater Gargadon	$1.04
ARTIFACTS (12)	
3 Fieldmist Borderpost	$1.50
4 Firewild Borderpost	$0.92
2 Mistvein Borderpost	$1.06
3 Wildfield Borderpost	$0.63
INSTANTS (18)	
3 Quandrix Command	$1.47
4 Violent Outburst	$31.64
2 Dead // Gone	$0.24
3 Fire // Ice	$0.75
3 Incubation // Incongruity	$0.27
3 Discovery // Dispersal	$0.84

SORCERIES (3)	
3 Restore Balance	$2.13
LANDS (18)	
1 Arid Mesa	$17.77
2 Breeding Pool	$43.68
1 Flooded Strand	$36.66
1 Forest	$0.10
1 Island	$0.10
1 Ketria Triome	$11.27
1 Plains	$0.10
1 Polluted Delta	$42.10
1 Steam Vents	$15.55
2 Stomping Ground	$17.52
4 Windswept Heath	$106.64
2 Wooded Foothills	$83.20
SIDEBOARD (4)	
2 Failure // Comply	$0.36
2 Wear // Tear	$4.54

(c) Deck viewer

Figure 2.2: *Scryfall* advanced search, deck builder and deck viewer page.

2.1.3 Commander Spellbook

Commander Spellbook[5] is a card combo search web application. This site allows users to search for card combos legal in the commander format. It is very useful to find game-ending effects to use in your own decks to finish and win the game.

It has an advanced search feature so users can be more specific about the card combo characteristics such as how many cars are needed or which colors has to have the deck to play it. The simple search allows to input a card name to find combos using a specific card.

The website's interface is clear and not too clunky, so it provides a pleasant experience to its users. See Figure 2.3 for more detail.

An important mention is that the *Commander Spellbook* database for card combos is free and open source, so it can be interesting to consider it to use it in the project.

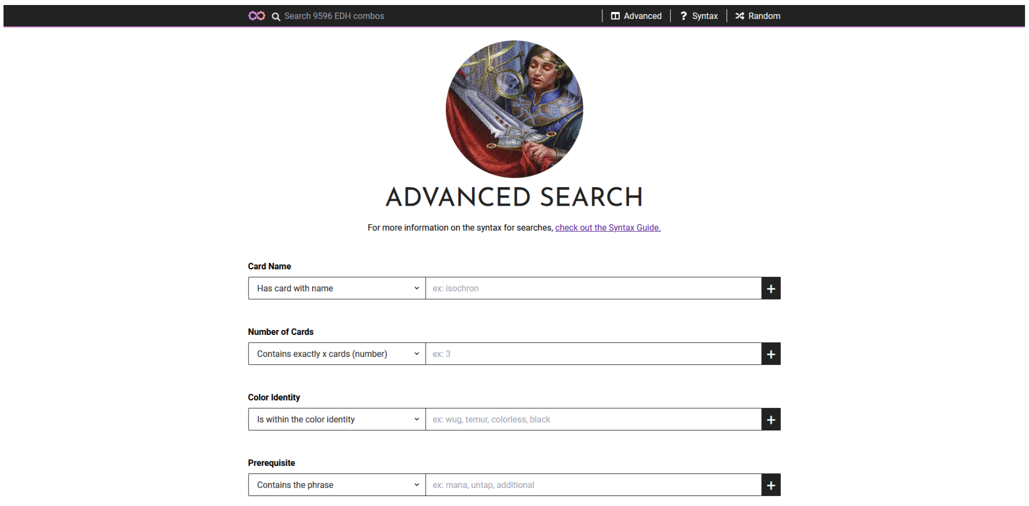
2.1.4 EDHREC

EDHREC[6] is a web application that provides users with different statistics from community submitted decks. The decks are commander format decks and some example statistics are a percentage of inclusion a card has for any deck that uses a specific commander, most played cards for each color or color combination and the most included cards for a specific commander deck.

This application also lets users select a specific strategy amongst the strategies used for a commander to improve the card recommendations it provides. Another thing users can see is which commander is the most popularly played with a specific card.

This website is very useful to build commander decks because it can provide a good starting point for a new deck, as well as allow the user to discover cards they never considered for their deck.

Figure 2.4 shows part of the information available on *EDHREC*.



ADVANCED SEARCH

For more information on the syntax for searches, check out the Syntax Guide.

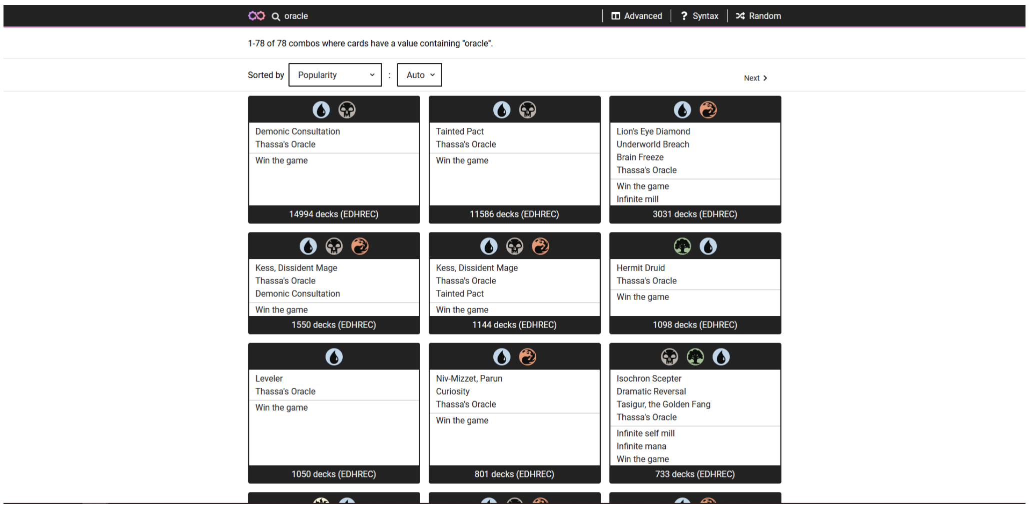
Card Name
Has card with name | ex: isochron

Number of Cards
Contains exactly x cards (number) | ex: 3

Color Identity
Is within the color identity | ex: wug, temur, colorless, black

Prerequisite
Contains the phrase | ex: mana, untap, additional

(a) Advanced Search (part of the total interface)

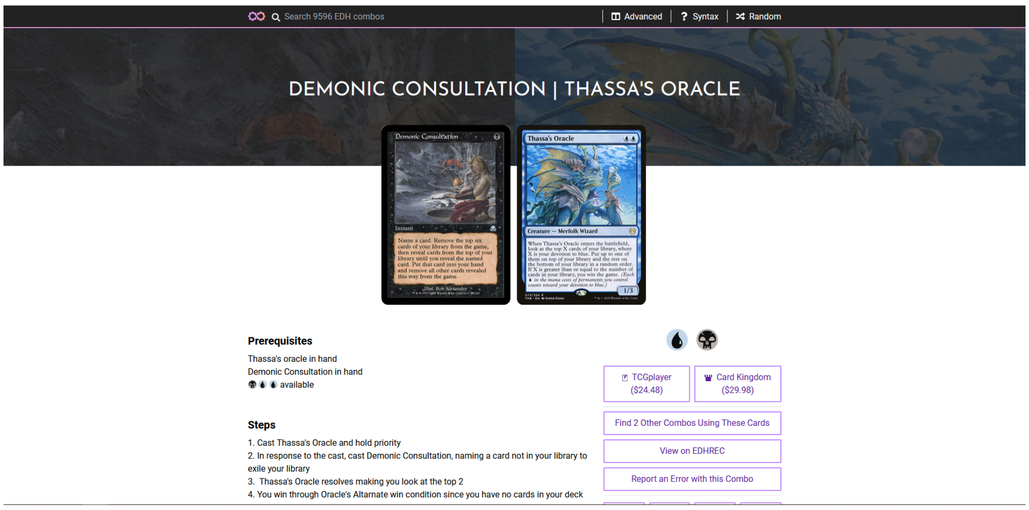


1-78 of 78 combos where cards have a value containing "oracle".

Sorted by Popularity : Auto

Demonic Consultation Thassa's Oracle Win the game 14994 decks (EDHREC)	Tainted Pact Thassa's Oracle Win the game 11586 decks (EDHREC)	Lion's Eye Diamond Underworld Breach Brain Freeze Thassa's Oracle Win the game Infinite mill 3031 decks (EDHREC)
Kess, Dissident Mage Thassa's Oracle Demonic Consultation Win the game 1550 decks (EDHREC)	Kess, Dissident Mage Thassa's Oracle Tainted Pact Win the game 1144 decks (EDHREC)	Hermit Druid Thassa's Oracle Win the game 1098 decks (EDHREC)
Leveler Thassa's Oracle Win the game 1050 decks (EDHREC)	Niv-Mizzet, Parun Curiosity Thassa's Oracle Win the game 801 decks (EDHREC)	Isochron Scepter Dramatic Reversal Tasigur, the Golden Fang Thassa's Oracle Infinite self mill Infinite mana Win the game 733 decks (EDHREC)

(b) List of card combos available with cards containing the word **oracle** in their name



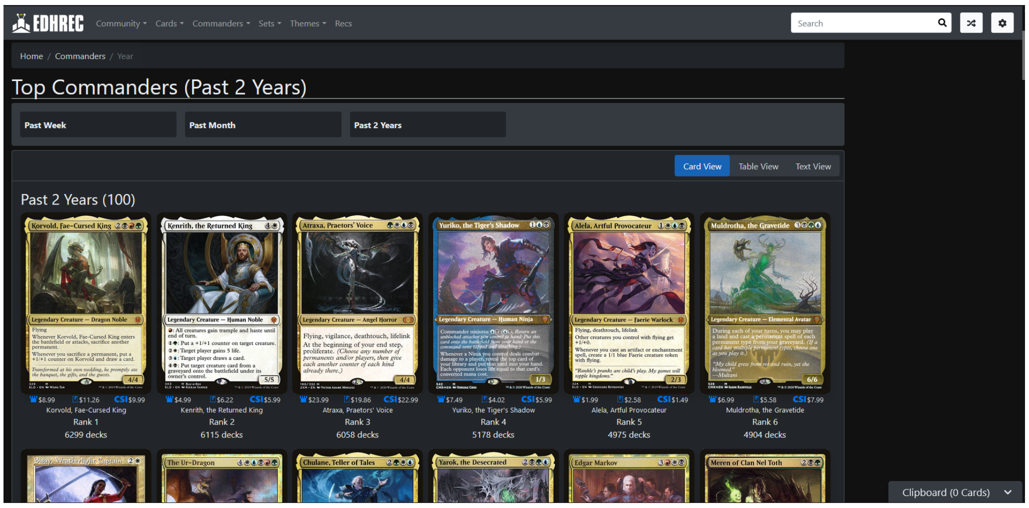
DEMONIC CONSULTATION | THASSA'S ORACLE

Prerequisites
Thassa's oracle in hand
Demonic Consultation in hand
available

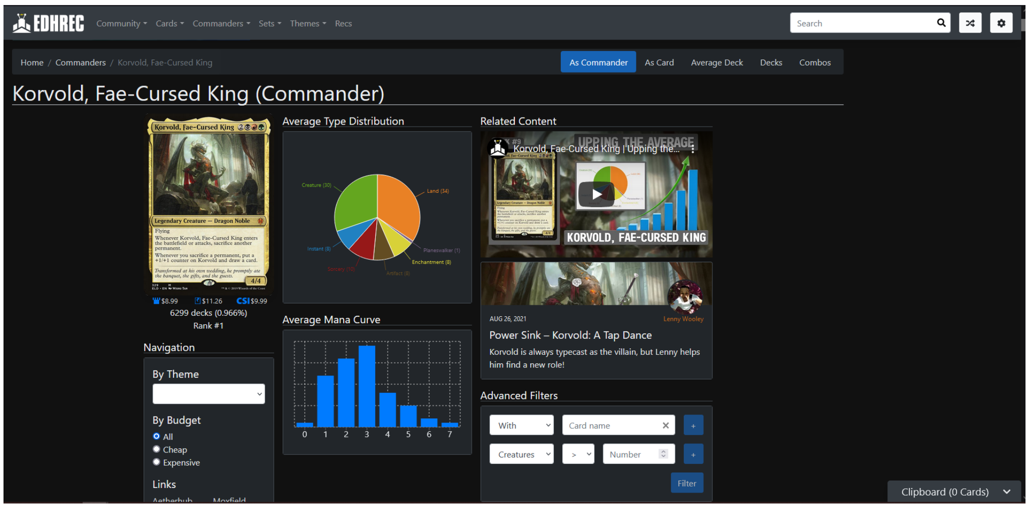
Steps
1. Cast Thassa's Oracle and hold priority
2. In response to the cast, cast Demonic Consultation, naming a card not in your library to exile your library
3. Thassa's Oracle resolves making you look at the top 2
4. You win through Oracle's Altarnate win condition since you have no cards in your deck

(c) Details page of a specific card combo

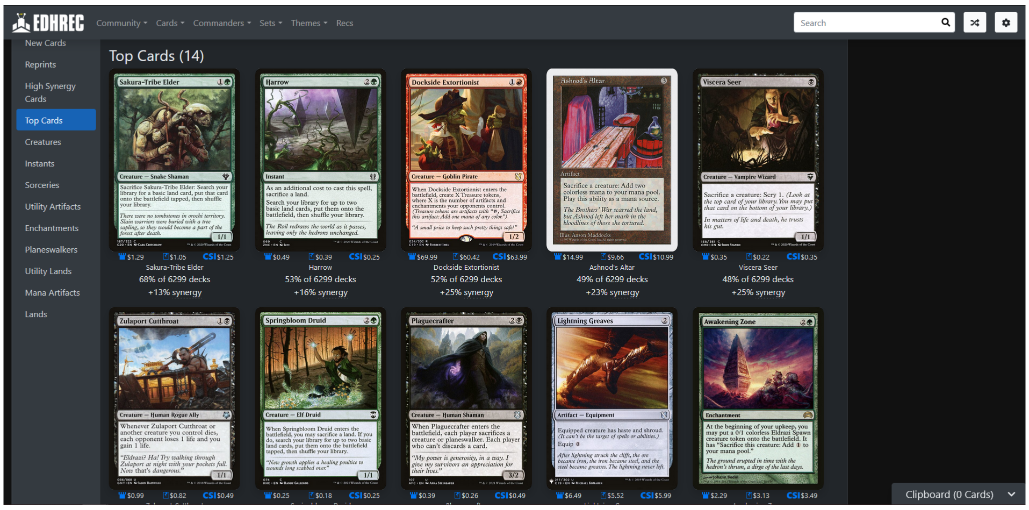
Figure 2.3: *Commander Spellbook* interface captures.



(a) List of the most used commanders



(b) General statistics from decks of a specific commander



(c) List of the most used cards with a specific commander

Figure 2.4: *EDHREC* interface captures.

2.2 Deck Building Applications

2.2.1 TappedOut

TappedOut[7] is a deck building web application. It has a very complete deck builder with lots of options to customize how the deck can be viewed. It also displays a good amount and variety of statistics of the deck built (like mana costs and mana production ratios by color).

There are two factors that hold back *TappedOut* from being an excellent deck building application: its interface is old and thus not very appealing and the card addition process during deck building is clunky, which makes deck building on this platform tedious sometimes. When adding a card the user can not indicate its name and add it, they have to specify a printing at the very least. There is not an auto-fill function for the name either. This makes the process very slow.

TappedOut allows users to search decks from other players and comment on decks. There is a big community of players that use *TappedOut*. Users can also write a description on their deck to help other players understand it or to remember some concepts themselves.

You can see *TappedOut*'s interface on Figure 2.5.

2.2.2 Moxfield

Moxfield[8] is a deck building web application. It resembles *TappedOut* because users can explore the community decks, but the difference is that on *Moxfield* users can not leave comments on other decks.

The interface on *Moxfield* is easy to use and appealing to the user. Creating and editing decks on this site is quick and easy, and there are multiple tools that help through the process. The user can select filters during deck edition to highlight specific cards on the deck, as well as group and/or sort card by different parameters.

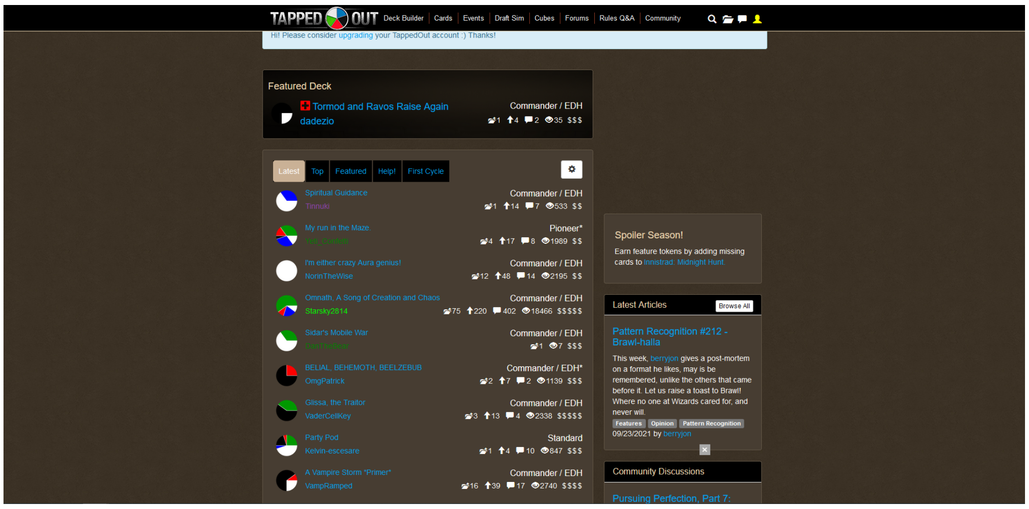
One great functionality *Moxfield* has is the packages functionality. It allows players to define a group of cards as a package so they can all be quickly added to any deck simultaneously. This is very useful, especially for format staples.

2.2.3 Archidekt

Archidekt[9] is a deck building web application. Archidekt is very visual, like moxfield, and is very dynamic aswell. It implements the functionalities that all the previous sites have in common, and in addition it has an in-built card recommendation functionality that recommends cards to add to the user's deck using data from *EDHREC*.

Archidekt is one of the top deck editors at the moment along *Moxfield*.

It also has a very complete stats functionality that even shows the odds of drawing a specific card on the opening hand.



Hi! Please consider upgrading your TappedOut account :) Thanks!

Featured Deck

Tormod and Ravos Raise Again — dadezio — Commander / EDH — 1 4 2 35 $$$

Spiritual Guidance — Tinnuki — Commander / EDH — 1 14 7 533 $$

My run in the Maze. — Yeti_Confetti — Pioneer* — 4 17 8 1989 $$

I'm either crazy Aura genius! — NorinTheWise — Commander / EDH — 12 48 14 2195 $$

Omnath, A Song of Creation and Chaos — Starsky2814 — Commander / EDH — 75 220 402 18466 $$$$$

Sidar's Mobile War — DanTheBear — Commander / EDH — 1 7 $$$

BELIAL, BEHEMOTH, BEELZEBUB — OmgPatrick — Commander / EDH* — 2 7 2 1139 $$$

Glissa, the Traitor — VaderCellKey — Commander / EDH — 3 13 4 2338 $$$$$

Party Pod — Kelvin-escesare — Standard — 1 4 10 847 $$$

A Vampire Storm *Primer* — VampRamped — Commander / EDH — 16 39 17 2740 $$$$

Spoiler Season!

Earn feature tokens by adding missing cards to Innistrad: Midnight Hunt.

Latest Articles

Pattern Recognition #212 - Brawl-halla

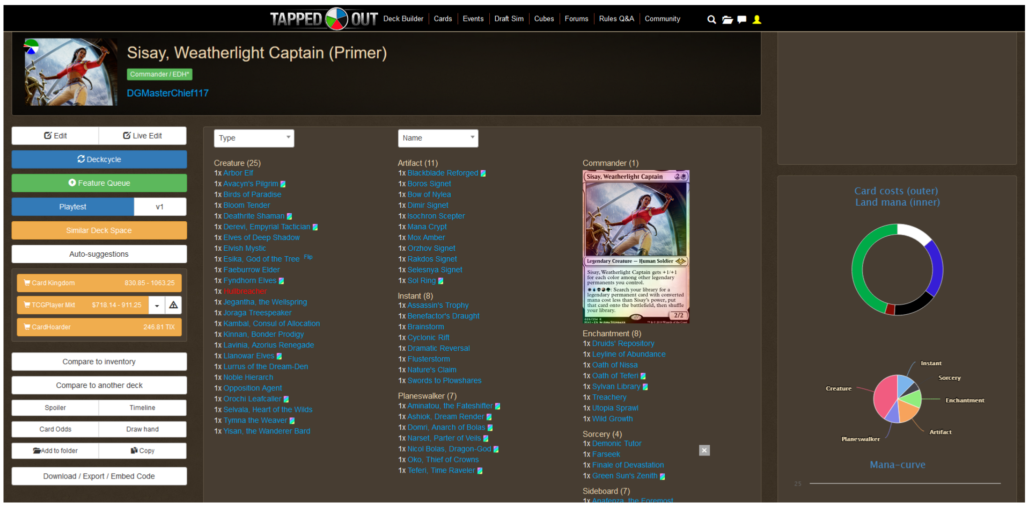
This week, berryjon gives a post-mortem on a format he likes, may is be remembered, unlike the others that came before it. Let us raise a toast to Brawl! Where no one at Wizards cared for, and never will.

09/23/2021 by berryjon

Community Discussions

Pursuing Perfection, Part 7:

(a) Home page



Sisay, Weatherlight Captain (Primer)

Commander / EDH*

DGMasterChief117

Card Kingdom 830.85 - 1063.25

TCGPlayer Mkt $718.14 - 911.25

CardHoarder 246.81 TIX

Creature (25)
1x Arbor Elf
1x Avacyn's Pilgrim
1x Birds of Paradise
1x Bloom Tender
1x Deathrite Shaman
1x Derevi, Empyrial Tactician
1x Elves of Deep Shadow
1x Elvish Mystic
1x Esika, God of the Tree Flip
1x Faeburrow Elder
1x Fyndhorn Elves
1x Hullbreacher
1x Jegantha, the Wellspring
1x Joraga Treespeaker
1x Kambal, Consul of Allocation
1x Kinnan, Bonder Prodigy
1x Lavinia, Azorius Renegade
1x Llanowar Elves
1x Lurrus of the Dream-Den
1x Noble Hierarch
1x Opposition Agent
1x Orochi Leafcaller
1x Selvala, Heart of the Wilds
1x Tymna the Weaver
1x Yisan, the Wanderer Bard

Artifact (11)
1x Blackblade Reforged
1x Boros Signet
1x Bow of Nylea
1x Dimir Signet
1x Isochron Scepter
1x Mana Crypt
1x Mox Amber
1x Orzhov Signet
1x Rakdos Signet
1x Selesnya Signet
1x Sol Ring

Instant (8)
1x Assassin's Trophy
1x Benefactor's Draught
1x Brainstorm
1x Cyclonic Rift
1x Dramatic Reversal
1x Flusterstorm
1x Nature's Claim
1x Swords to Plowshares

Planeswalker (7)
1x Aminatou, the Fateshifter
1x Ashiok, Dream Render
1x Domri, Anarch of Bolas
1x Narset, Parter of Veils
1x Nicol Bolas, Dragon-God
1x Oko, Thief of Crowns
1x Teferi, Time Raveler

Commander (1)

Sisay, Weatherlight Captain

Legendary Creature — Human Soldier

Sisay, Weatherlight Captain gets +1/+1 for each color among other legendary permanents you control.

: Search your library for a legendary permanent card with converted mana cost less than Sisay's power, put that card onto the battlefield, then shuffle your library.

2/2

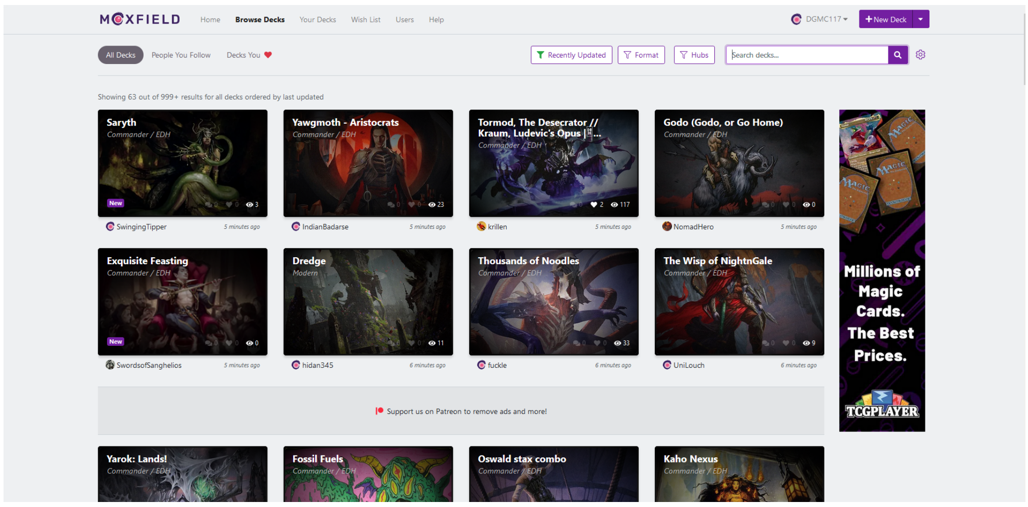
Enchantment (8)
1x Druids' Repository
1x Leyline of Abundance
1x Oath of Nissa
1x Oath of Teferi
1x Sylvan Library
1x Treachery
1x Utopia Sprawl
1x Wild Growth

Sorcery (4)
1x Demonic Tutor
1x Farseek
1x Finale of Devastation
1x Green Sun's Zenith

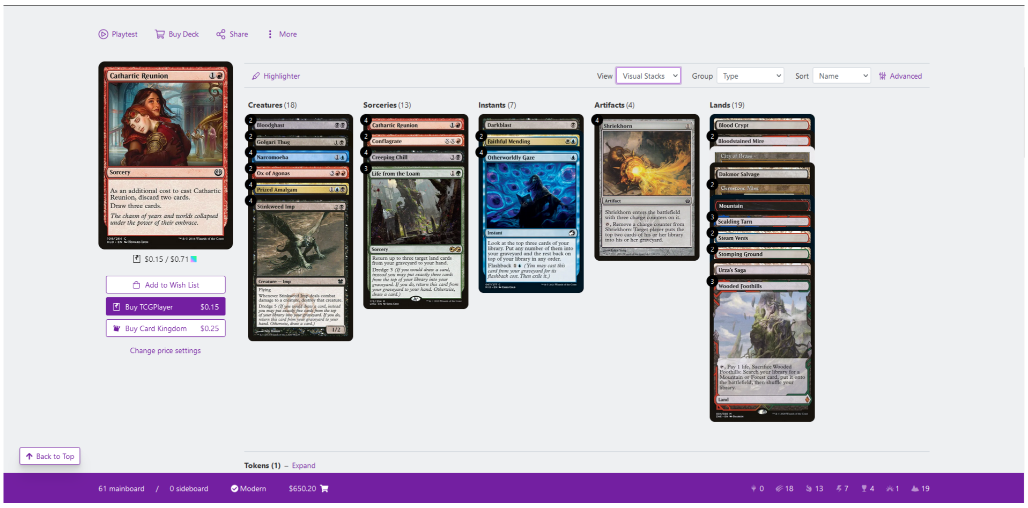
Sideboard (7)
1x Anafenza, the Foremost

(b) Deck view

Figure 2.5: *TappedOut* interface captures.

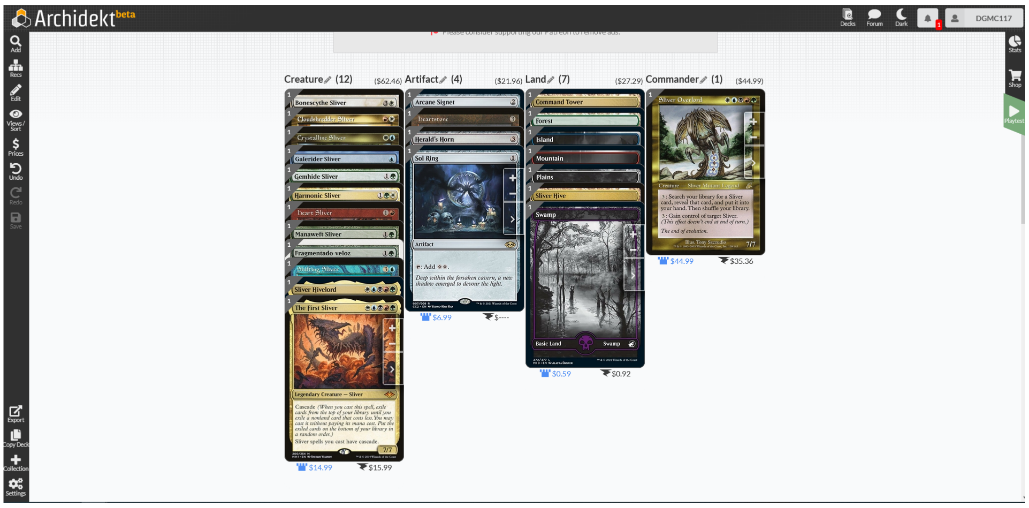


(a) Deck browser

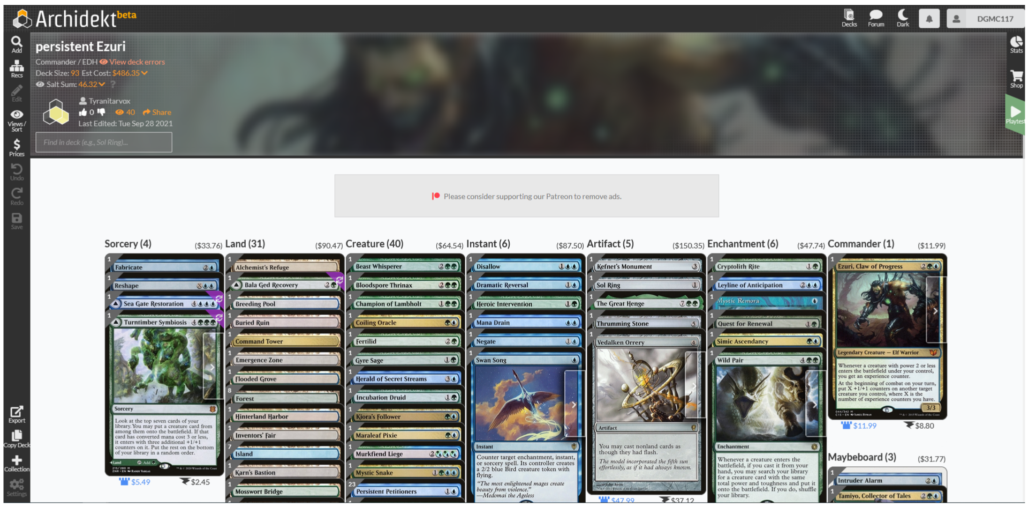


(b) Deck view

Figure 2.6: *Moxfield* interface captures.



(a) Deck editor



(b) Deck view

Figure 2.7: *Archidekt* interface captures.

2.2.4 ManaBox

ManaBox[10] is a companion mobile application. You can find a list of functionalities and interface captures at *ManaBox*'s *GooglePlay* page.

ManaBox's functionalities are easy and fast to use, but are fairly limited when compared with some of the functions that web applications bring to the table. The added functionalities of life counter and dice simulator are very useful when playing the game. In that sense it is more useful than web applications.

2.2.5 TopDecked MTG

TopDecked MTG[11] is a companion mobile application. You can find a list of functionalities and interface captures at *TopDecked MTG*'s *GooglePlay* page.

TopDecked MTG has more functionalities than *ManaBox*, but it comes at a price. It has a premium subscription to allow users to many functionalities such as a game simulator to test any deck.

The app also allows users to read Magic: The Gathering related articles and browse metagame decks.

2.3 Conclusion — Justification

As seen on this study, the world of Magic: The Gathering deck building has a lot of successful tools and many different variations on tools. Mobile applications are quite behind web applications, and this is something our product can take advantage of. This project can adapt the current web-exclusive functionalities to mobile to fill the current void that exists on the market.

We can learn a lot about the existing apps to make an interface that is intuitive and quick to use, and we can improve the existing companion model to another level.

One functionality I believe can be very useful is to adapt and improve the *Similar to:* functionality so it finds similar cards and then can search for cheaper alternative cards that users can use in their deck, reducing the cost of Magic: The Gathering for users that are more concerned about budget. This functionality could be the thing that gives the project a distinctive trait over other applications. Incorporating a database for combos and implementing a combo search in a mobile application would be very good too.

Chapter 3

Scope of the Project

The final product of the project will be a companion app that will allow the user to search for cards and create decks, implementing unique mobile functionalities that will improve the deck building process for the users. This app will be able to work offline, without the need of internet for the main functions (after an initial card download for new cards). The reason behind this is that this way the app can work in low connectivity environments (suck as the underground, for example) and also reduces mobile data consumption.

The focus of the project will be on the deck building functionality, which will incorporate a *Similar to:* functionality and a card combo search. An advanced card search will be implemented, so the final app will not have to depend on an external service to look for cards. For card images an external service will be used, since card images are not essential to the deck building process.

A dice simulator and a life counter will be added as well. These are useful when playing the game. The reasoning behind implementing these two functionalities over others is that the companion aspect is expected from the user, so it is better to implement these because if the user can not find the functionalities they may think the app is not as good as the others on the market (even when taking into account the extra deck building functionalities).

Additional features will be considered if time is available when all core functionalities are implemented. These include purely aesthetic upgrades to the interface, but also include card pack creation and advanced settings for the application.

Finally, the final product will be available in English and Spanish, and will feature a light and a dark theme.

3.1 Objectives

The main objective is to create a companion mobile application that provides users with a deck builder, card search, life counter and dice simulator. A list of objectives is listed below:

- Create a companion app that provides users with essential functionalities for deck building and playing with the deck.
- Provide users with a complete deck builder functionality that incorporates utilities new to the mobile market.

- Help users to potentially save money by recommending cheaper alternatives to any card (if they exist).
- Provide users with an advanced search tool that has a vast variety of search parameters.

Later in the document the objectives will be transformed into functional requirements that will also be transformed into tasks. For now non-functional requirements will be defined.

3.2 Non-Functional Requirements

Non-functional requirements are requirements related to the quality of the product, not the specific functionalities it implements. They are related to how good the product looks, or how easy to use is, or even how safe it is to use. Below you will find a list of non-functional requirements and a description of how the final product will incorporate them:

- **Appearance** The app has to be appealing to the user to invite them to use it. To improve the appearance of the app material design[12] colours, methods and animations will be incorporated, and visuals will have a minimalist approach.
- **Usability** The product has to be easy to use and intuitive. To accomplish this the implementation will be similar to what there is already on the market. If the user expects a button to do something specific, it probably should do it.
- **Availability** The app has to be available to use at any time. The app will be designed to work in offline mode, but with some restrictions such as not being able to load card images.
- **Learning** It has to be easy for the user to get used to how the app works. The approach is the same as in the usability requirement. If there is time left near the end of the project an interactive tutorial will be implemented.
- **Internationalization** The platform has to be available for many users, so it should be available in different languages. As explained earlier it will be available in English and Spanish.
- **Speed** The app has to be fast so the user can have a pleasant experience. To avoid large waits the need for remote services has been reduced.
- **Capacity** The app has to be able to handle all its data flux without provoking any errors. Using a light and fast database will help avoid capacity problems.
- **Scale** The product has to be able to scale to accommodate an increasing number of users. Since each user has a different app instance on their device and the app does not need a centralised server there will be no scale problems.
- **Adaptability** The app has to work properly on multiple devices. The app will be implemented with responsive layouts to accomplish this.
- **Security and privacy** The app has to protect the user's data. There are no plans of storing any personal data of the user in the app.

3.3 Risks and/or Obstacles

During development the project can stumble into some obstacles. There are known risks for software development that can lead to finishing a product late, or even to not completing the initial vision at all. Below you will find some of the risks that apply to this project.

- **Technology inexperience** Before starting this project I have never worked with Kotlin before, nor have I tried to use the material design guidelines. Although Kotlin is essentially Java, it may occur that I have a problem implementing a functionality during development because of my inexperience. This could lead to not reaching the deadlines agreed to before.
- **Closed calendar** The fact that the deadlines for the project are already set supposes a considerable risk. This makes the development time very strict and can be very unforgiving, specially with combined with the previous risk. This can lead to having to cut functionalities near the end of the project.
- **Bugs** Bugs can sometimes be very hard to find and difficult to follow, depending on how the functionality code is structured. Good testing can help prevent or find bugs, but also restricts even more development time. The decision on how many resources will be put into test implementation or making is not to be taken lightly.

Chapter 4

Methodology

Work methodology refers to how the development will be carried out by the development team. Choosing the right development methodology is critical to ensure a project's success. For example, a classic methodology requires the team to make exhaustive documentation of the project, setting all parameters before starting to implement it. This can be very helpful for a big project with a clear and fix objective that is not going to change, but it is a bad decision if the project is subject to change during development because it is very strict. Agile methodology, on the other hand, is a less documentation-driven methodology. It allows to make some changes in-between development phases (or sprints), and since it needs less documentation it leaves more room for implementation time.

For this project I have chosen to work with agile methodology. I think this is the right choice because it will allow me to make any suggested changes the director suggests after each sprint, without setting everything on stone from the start. Later on the sprints date and duration will be set, and tasks will be assigned to them.

4.1 Project Management

To manage the different development phases, and keep track of the backlog of tasks to do, or all the tasks that are in process or done I will use *Trello*[13]. *Trello* is a project management software. It allows users to create a board where they can put lists of tasks. In the project's *Trello* board there will be the following lists:

- **Backlog** The list of all pending tasks for the project.
- **Sprint Backlog** The list of all pending tasks planned to do in the current sprint.
- **In progress** The list of tasks currently being carried out.
- **dd-mm-yyyy Done** The list of tasks completed for a specific sprint.

Trello will be helpful because the director will be able to write their thoughts about a task on the comments of that task. A lot of information can be written on tasks, such as text, checklists, due date, references to other tasks, etc.

4.2 Software Management

GitHub[14] is a software management platform. It is the most widely used by developers around the world. It allows developers to upload code on the cloud to avoid losing code in case of an accident on the work computer.

The way I will use *GitHub* on my project is the following: The master branch will contain all functionalities that are fully developed. For each functionality, there will be a new branch declared where the development will take place. This is to avoid uploading wrong code, or damaging the working code. Once the functionality is finished, its branch will be merged into the master branch. This way, the master branch will always have a working version and features from the other branches can be added whenever they are finished.

4.3 Technologies

For my project, in addition of the two platforms mentioned earlier, I have chosen to work with the following technologies:

- **Platform** *Android*[15]
- **Language** *Kotlin*[16]
- **Database Language** *SQLite*[17]
- **IDE** *Android Studio*[18]

I chose *Android* as the platform because it is one of the most broadly used mobile operating systems, and also because I own an *Android* device so it will be easier for me to test the application. The minimum Android version chosen that can run the application is *Android* 6 Marshmallow. This is because this *Android* version allows the implementation of some newer features of the material library. Since 94% of the devices have *Android* 6 or greater this will not limit significantly the reach of the application.

Since *Android* became a *Kotlin*-first platform I think that *Kotlin* is the right choice for the app's language. In addition to that, *Kotlin* incorporates tools that make development faster, and given the limited development time it is a good choice.

The chosen database language is *SQLite*[17]. The two options considered for the database where *SQLite* and *MongoDB*[19]. On the one hand, *SQLite* offers the most stable platform for *Android* and the most extended, so a lot of documentation should be available to help with development. On the other hand, *MongoDB* offers direct compatibility with the JSON format, which is the format in which cards are served by the servers, and also allows more freedom when creating the database. These were the initial beliefs, but after thorough investigation *SQLite* proved to be the best option for this project. More about this on the 6.4 Obstacles Encountered section.

Android Studio is the most used IDE for android development and is compatible with *Kotlin*.

4.3.1 Additional Technologies

Multiple additional technologies were used when documenting the project.

Overleaf[20] was used to write this document. *Overleaf* is an online text editor for documents written in LaTeX.

Lucid[21] is an online application used to draw graphs and charts. This resource was used to draw the Class Diagrams, Sequence Diagrams and any other diagram used in the documentation.

AnyChart[22] is a library that allows to generate graphs with any given data set. This was used in the stats screen for decks.

Inkscape[23] is a program for creating vectorial images. I used this tool to create icons for the app, such as the card, combo and deck icons.

4.4 Methodology Changes

During the development of the project, due to setbacks and limited time, some changes in the methodology mentioned above were made.

The project management has finally been done on Github. Github incorporates a functionality that allows to have tasks on a board in a style similar to Trello, but with a big difference: Only a title can be set to a task on the board. This limits the tool a lot compared to Trello, because the user cannot write additional information on the tasks for future reference. This change was made because I personally worked faster just Bookmarking any website with useful information, and by keeping track of only the state of a task it was enough. After all, I am working alone so I do not need any partner know where to continue a task from, for example. And since I work tasks from start to finish I did not have that problem either.

About the database language, SQLite was selected. There was a big and long effort to make MongoDB work, or even other NoSQL databases (more on this on later chapters), but finally it has been determined that the best fit for the application developed would be SQLite. SQLite is a small database that works on most of mobile devices, so it is perfect for this project. Of course, this database would have to be tested to see if it supports some of the most work-intense queries, but it should be able to manage the amount of data reasonably.

4.5 Knowledge Application

Since the aim of this project is to use all the techniques and knowledge gathered through the degree, I tried to use as much as possible when carrying it on.

Firstly, to plan the phases of the project and the different tasks, as well as documenting most of the project. In the specialisation branch I chose for the degree, software engineering, a focus is made into the variety of manners to plan a project and how to use them, so I really could use this to my benefit for developing this application. I could document important information, such as a class diagram to help me choose a database design that fits the needs of the application users.

Secondly, my specialisation also taught me design patterns for applications, which I could apply in my project to make my code easier, cleaner, scalable and more adaptable. A good example of this is the Adapter pattern I used for the database library, which would allow me to change the code that communicates with the database without modifying interface code.

Thirdly, Interacció i Disseny d'Interfícies (IDI)[24] and Aplicacions i Serveis Web (ASW)[25] take a focus on interface design and user focused designs. These subjects allowed me to recognise bad or non intuitive design and improve it, so I would say they have helped me implementing a more user-friendly application.

Finally, Software Lliure i Desenvolupament Social (SLDS)[26] taught me all i needed to know for this project about licenses and legal-related information.

Chapter 5

Temporal Planning

This chapter contains a task specification with more concrete metrics so they can be estimated in an easier way, a task summary in table format and a Gantt diagram that shows the work flow and planning on a temporal scale.

5.1 Tasks Definition

For a better understanding and organisation of tasks, these will be divided into groups and subgroups. For example, planning and scope definition are contained in project management which contains both and others. These classification will help when planning sprints, since it will be less desirable to split a group between two sprints if it can be put on only one.

Below you will find all the project tasks with a corresponding code that will be referenced in the table on the following section. The tasks have been organised in three main sections: project management, development and documentation/communication. Project management refers to tasks related to planning the project, development are functionalities of the app and refer to the coding of such functionalities, and finally documentation/communication refers to tasks like weekly meetings with the project's director and writing documentation of the project.

5.1.1 Project Management

- **PM1 - Market Analysis and Scope Definition** Alternatives to the project app currently on the market will be analysed and the objectives of the project will be set, justified with information extracted in the market analysis. The methodology that I will follow to carry out the project will be explained as well. This task has an estimated time of 20 hours.

- **PM2 - Planning** Tasks will be specified in a detailed and concise way. A time estimation will be assigned to each of the project's tasks and they will be organised in groups and be put in a time schedule using a Gantt diagram. This task has an estimated time of 10 hours. Dependencies: PM1.

- **PM3 - Budget and Sustainability** A budget will be drafted using the estimate times for each task and a salary reference for each of the employees needed for those tasks. The economic, social and environmental sustainability of the project will be planned and explained. This task has an estimate time of 5 hours. Dependencies: PM2.

5.1.2 Development

Inception

- **IC1 - Environment Setup** This includes the installation of all the software needed to develop the application. The time estimate for this task is 10 hours. Dependencies: PM2.
- **IC2 - Interface Drafting** A draft of the interfaces for the app will be made. This will help tune the interface to serve the user better. This task has an estimate time of 20 hours. Dependencies: IC1.
- **IC3 - Database Definition** Definition of the database structure and organisation. This task has an estimate time of 10 hours. Dependencies: IC2.

Data Obtaining

- **DO1 - Card Data Download** Implementation of the code that downloads raw card data from an external service to keep the database updated. This task has an estimate time of 15 hours. Dependencies: IC3.
- **DO2 - Card Combo Data Download** Implementation of the code that downloads raw card combos data from an external service to keep the database updated. This task has an estimate time of 15 hours. Dependencies: DO1.
- **DO3 - Card Pricing Download** Implementation of the code that downloads raw card data pricing an external service to keep the database updated. This task has an estimate time of 10 hours. Dependencies: DO1.

Card Search

- **CS1 - Name Search** Implementation of the code that searches cards by their name. This task has an estimate time of 25 hours. Dependencies: DO3.
- **CS2 - Advanced Card Search** Implementation of the code that searches cards by multiple parameters such as cost, power or toughness. This task has an estimate time of 15 hours. Dependencies: CS1.
- **CS3 - *Similar to:* Search** Implementation of the code that searches for cards similar to a specific card. This task has an estimate time of 30 hours. Dependencies: CS2.
- **CS4 - Card Combo Search** Implementation of the code that searches card combos using a specific card. This task has an estimate time of 20 hours. Dependencies: CS1, DO2.

Card View

- **CV1 - Card Information Screen** Implements the code of the card details page that includes the text, rulings and other information of the card. This task has an estimate time of 30 hours. Dependencies: CS1.
- **CV2 - Card Image** Implements the code of the card image's retrieval and presentation. This task has an estimate time of 5 hours. Dependencies: CV1.

Deck Building

- **DB1 - Create an Empty Deck** Implementation of the code that creates an empty deck and takes the user to the editing screen. This task has an estimate time of 20 hours. Dependencies: IC3.
- **DB2 - Add a Card to a Deck** Implementation of the code that adds a card to the deck the user is editing. This task has an estimate time of 20 hours. Dependencies: DB1, CS2, CV2.
- **DB3 - Filter/Group/Sort Deck Cards** Implements the code that allows the user to filter cards in a deck to obtain certain cards that have specific characteristics. Also implements the code that allows users to group and sort cards using a specific criteria. This task has an estimate time of 20 hours. Dependencies: DB2.
- **DB4 - Budgetfy** Implements the code that searches for cheaper card alternatives using a given budget per card and the *Similar to:* search. This task has an estimate time of 30 hours. Dependencies: DB2, CS3.
- **DB5 - Combo Wizard** Implements the code that searches for combos that use cards in a specific deck. This task has an estimate time of 10 hours. Dependencies: DB2, CS4.
- **DB6 - Deck Statistics** Implements the code that calculates deck statistics and displays it on screen. This task has an estimate time of 15 hours. Dependencies: DB2.
- **DB7 - Card Pack Creation** Allows the users to create card packages that can be added to a deck to avoid having to add the cards one by one. This task has an estimate time of 10 hours. Dependencies: CS1, DB1.

Companion Features

- **CF1 - Dice Simulator** Implements the code that simulates a dice throw with an arbitrary number of faces. Essentially a Random Number Generator (RNG). This task has an estimate time of 10 hours. Dependencies: IC3.
- **CF2 - Life Counter** Implements the code of a life simulator that keeps track of the life total and other data of the players for an arbitrary number of players between 2 and 6. This task has an estimate time of 14 hours. Dependencies: IC3.

Additional Features

This features are features that I will not consider for the base of the project but I will put in the document to leave as a possibility if the project is finished ahead of schedule.

- **Interactive Tutorial** Add an interactive tutorial to the final app to ease the users into learning how it works. This task has an estimate time of 15 hours.

5.1.3 Documentation/Communication

- **DC1 - Project Following** Tasks performed and meetings will be noted to keep a record of the project making. The time estimate for this task is 6 hours.
- **DC2 - Project Documentation** All the parts of the projects have to be well documented. This includes more evident parts like the project management, but also how a specific functionality works or drafts of the interface. The time estimate for this task is 60 hours.

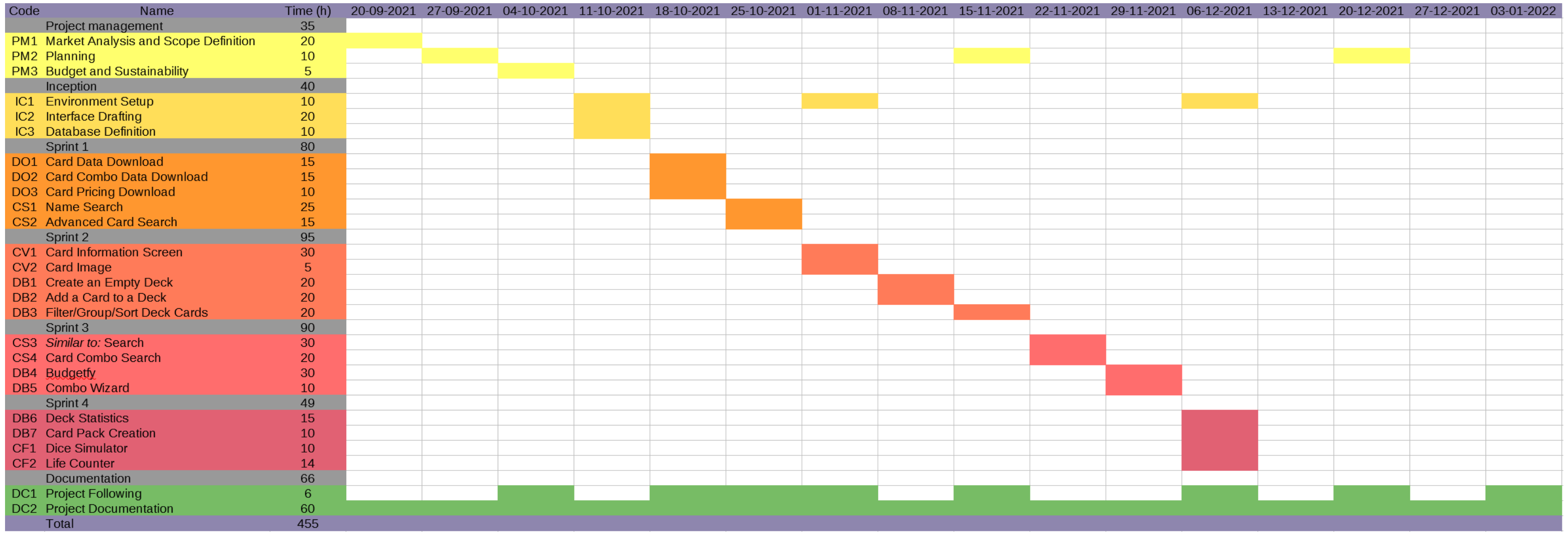
5.2 Task Summary

Table 5.1 will not include the additional features listed on the previous section. This is because this table will serve as a reference to the base project completion. Resources used in the table are the following: personal computer (PC), LaTeX[20], *Gimp*[27], *Adobe Illustrator* (IL)[28], *Android Studio* (AS)[18], *Trello* (TR)[13], *Google Meet* (GM)[29].

Code	Task Name	Time (h)	Dependencies	Resources
PM	Project Management	35		
PM1	Market Analysis and Scope Definition	20	-	PC
PM2	Planning	10	PM1	PC
PM3	Budget and Sustainability	5	PM2	PC
IC	Inception	40		
IC1	Environment Setup	10	PM2	PC
IC2	Interface Drafting	20	IC1	PC, Gimp, IL
IC3	Database Definition	10	IC2	PC, AS
DO	Data Obtaining	40		
DO1	Card Data Download	15	IC3	PC, AS
DO2	Card Combo Data Download	15	DO1	PC, AS
DO3	Card Pricing Download	10	DO1	PC, AS
CS	Card Search	90		
CS1	Name Search	25	DO3	PC, AS
CS2	Advanced Card Search	15	CS1	PC, AS
CS3	*Similar to:* Search	30	CS2	PC, AS
CS4	Card Combo Search	20	CS1, DO2	PC, AS
CV	Card View	35		
CV1	Card Information Screen	30	CS1	PC, AS
CV2	Card Image	5	CV1	PC, AS
DB	Deck Building	125		
DB1	Create an Empty Deck	20	IC3	PC, AS
DB2	Add a Card to a Deck	20	DB1, CS2, CV2	PC, AS
DB3	Filter/Group/Sort Deck Cards	20	DB2	PC, AS
DB4	Budgetfy	30	DB2, CS3	PC, AS
DB5	Combo Wizard	10	DB2, CS4	PC, AS
DB6	Deck Statistics	15	DB2	PC, AS
DB7	Card Pack Creation	10	CS1, DB1	PC, AS
CF	Companion Features	24		
CF1	Dice Simulator	10	IC3	PC, AS
CF2	Life Counter	14	IC3	PC, AS
DC	Documentation/Communication	66		
DC1	Project Following	6		PC, TR
DC2	Project Documentation	60		PC, LaTeX
Total		455		

Table 5.1: Task Summary Table

5.3 Gantt Diagram



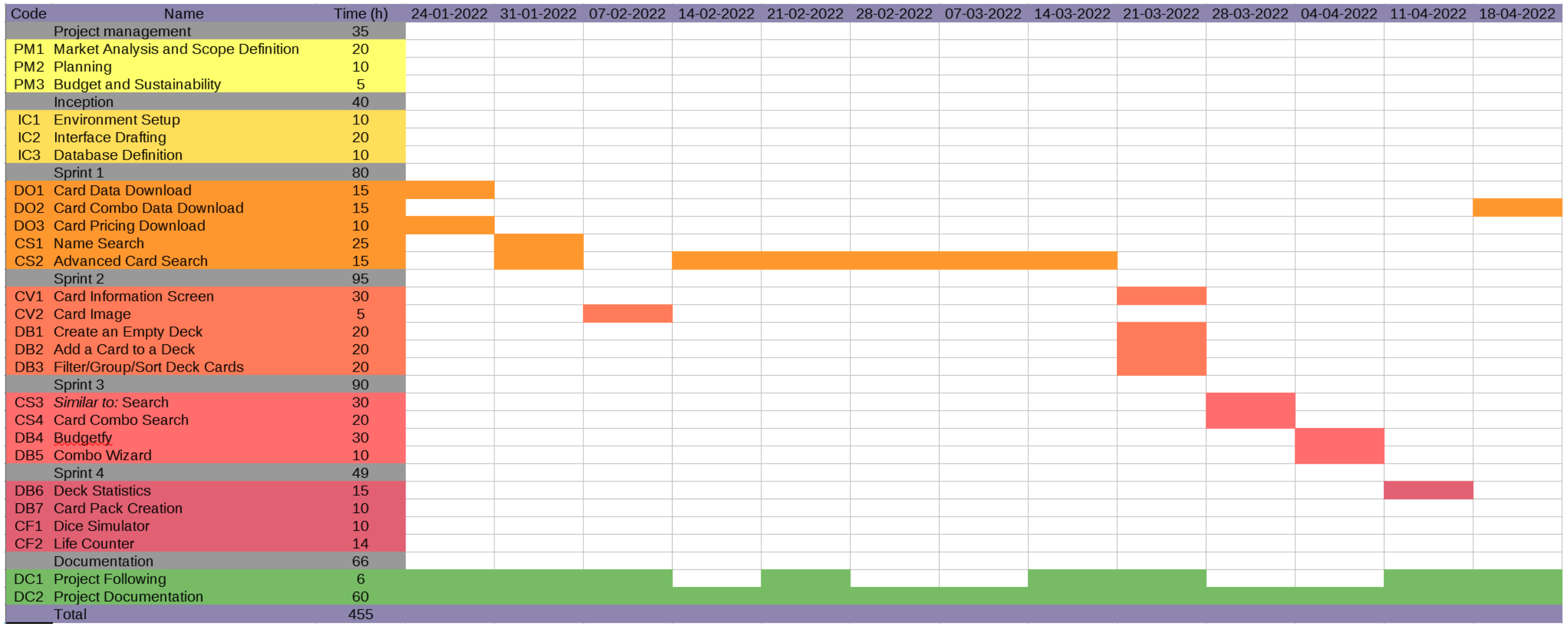
Code	Name	Time (h)	20-09-2021	27-09-2021	04-10-2021	11-10-2021	18-10-2021	25-10-2021	01-11-2021	08-11-2021	15-11-2021	22-11-2021	29-11-2021	06-12-2021	13-12-2021	20-12-2021	27-12-2021	03-01-2022
	Project management	35																
PM1	Market Analysis and Scope Definition	20	🟨															
PM2	Planning	10		🟨							🟨					🟨		
PM3	Budget and Sustainability	5			🟨													
	Inception	40																
IC1	Environment Setup	10				🟨			🟨					🟨				
IC2	Interface Drafting	20				🟨												
IC3	Database Definition	10				🟨												
	Sprint 1	80																
DO1	Card Data Download	15					🟧											
DO2	Card Combo Data Download	15					🟧											
DO3	Card Pricing Download	10					🟧											
CS1	Name Search	25						🟧										
CS2	Advanced Card Search	15						🟧										
	Sprint 2	95																
CV1	Card Information Screen	30							🟥									
CV2	Card Image	5							🟥									
DB1	Create an Empty Deck	20								🟥								
DB2	Add a Card to a Deck	20								🟥								
DB3	Filter/Group/Sort Deck Cards	20									🟥							
	Sprint 3	90																
CS3	*Similar to:* Search	30										🟥						
CS4	Card Combo Search	20										🟥						
DB4	Budgetfy	30											🟥					
DB5	Combo Wizard	10											🟥					
	Sprint 4	49																
DB6	Deck Statistics	15												🟥				
DB7	Card Pack Creation	10												🟥				
CF1	Dice Simulator	10												🟥				
CF2	Life Counter	14												🟥				
	Documentation	66																
DC1	Project Following	6			🟩		🟩	🟩	🟩		🟩			🟩		🟩		🟩
DC2	Project Documentation	60	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	Total	455																

Figure 5.1: Gantt diagram.

5.4 Planning Changes

Because of the obstacles encountered, a re-scheduling for the project had to be made. As explained before in the document, some tasks were removed from the project's scope and the planning overall was re-scheduled to fit the new time frame. On Figure 5.2 you can see the changes made to the schedule.

Unfortunately, the card search functionalities proved to be a tougher challenge than expected, mainly because they are formed by two views that are fairly complex, especially given my inexperience with UI implementation. This would count for the inexperience with the technology risk that we pointed out before starting the project, but it still represents a big hit at my productivity during the project's making. In addition, my personal job interfered with my project hours due to an increase of projects commissioned during the development of the project, which made me work in a double schedule.



Code	Name	Time (h)	24-01-2022	31-01-2022	07-02-2022	14-02-2022	21-02-2022	28-02-2022	07-03-2022	14-03-2022	21-03-2022	28-03-2022	04-04-2022	11-04-2022	18-04-2022
	Project management	35													
PM1	Market Analysis and Scope Definition	20													
PM2	Planning	10													
PM3	Budget and Sustainability	5													
	Inception	40													
IC1	Environment Setup	10													
IC2	Interface Drafting	20													
IC3	Database Definition	10													
	Sprint 1	80													
DO1	Card Data Download	15	🟧												
DO2	Card Combo Data Download	15													🟧
DO3	Card Pricing Download	10	🟧												
CS1	Name Search	25		🟧											
CS2	Advanced Card Search	15		🟧		🟧	🟧	🟧	🟧	🟧					
	Sprint 2	95													
CV1	Card Information Screen	30									🟥				
CV2	Card Image	5			🟥										
DB1	Create an Empty Deck	20									🟥				
DB2	Add a Card to a Deck	20									🟥				
DB3	Filter/Group/Sort Deck Cards	20									🟥				
	Sprint 3	90													
CS3	*Similar to:* Search	30										🟥			
CS4	Card Combo Search	20										🟥			
DB4	Budgetfy	30											🟥		
DB5	Combo Wizard	10											🟥		
	Sprint 4	49													
DB6	Deck Statistics	15												🟥	
DB7	Card Pack Creation	10													
CF1	Dice Simulator	10													
CF2	Life Counter	14													
	Documentation	66													
DC1	Project Following	6	🟩	🟩	🟩		🟩			🟩	🟩			🟩	🟩
DC2	Project Documentation	60	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	Total	455													

Figure 5.2: The revised Gantt diagram for the project

Due to the obstacles of the first weeks, the final weeks have a higher hour dedication in order to compensate for the missing hours of the first weeks. Note that the tasks missing from the Gantt's Diagram are either tasks that were already done or tasks that were cut from the project. These are left there for the sake of comparison and context.

Chapter 6

Risk Management

As mentioned previously on the document there are a number of situations that can cause delays on the project, forcing the development to take a different course to handle any problem that is found during implementation. Each of the previously mentioned risks have been assigned a probability of occurrence and an estimated resolution time.

Risk	Probability	Hours to resolve
Technology inexperience	Medium	20
Closed calendar	Low	10
Bugs	Medium	20

Table 6.1: Risk probability and estimated resolution time.

Below you can find the proposed solutions for the risks.

6.1 Technology inexperience

Since it is the first time I work with *Kotlin*, it is to expect that I have doubts or do not know how to implement some functionalities.

Kotlin is a popular language and has a large community around the world, and a large quantity of documentation. Any problem that I find during development should be resolved by looking up some documentation on the internet and thus it should not be a problem that impacts the project in a significant manner.

If I could not find any solution to my problem in existing documentation or forum threads, I would open a new thread on a *Kotlin* forum to see if anyone can help with my current problem.

6.2 Closed calendar

The strict delivery dates and low amount of time available for the project can impact the number of tasks that will be complete at the end of the project.

The great amount of planning performed on the project should be able to reduce the probability of this risk affecting the development. If it happens, the less important tasks could be cut from the project to attempt to compensate for the hours needed to complete another task.

6.3 Bugs

In order to avoid bugs or detect them on time for a fix, testing will be performed on different functionalities. Manual testing will be performed on the different interfaces, and intentionally malicious inputs will be introduced on search fields, etc. to try the sturdiness of the program and its consistency.

6.4 Obstacles Encountered

Unfortunately, some of the discussed risks happened during development. In this section we will discuss which obstacles I encountered and what measures were taken to work around these setbacks.

The first obstacle I encountered was the technology inexperience that was expected, but instead of the expected difficulty with Kotlin, the problem was with the database. For this project, I wanted to use MongoDB because I thought that since the program almost exclusively makes queries, and not that many writes or updates, it would be best to use a NoSQL database. I spent several weeks trying to make Kmongo[30], a library for using MongoDB wih Kotlin, work. I realised later that this library only works to connect with a remote MongoDB database, so using it to create and manage a local database on the device was not possible.

After finding this issue, I searched for alternative libraries to create local MongoDB databases. The libraries I found were, in order: MongoDB Realm[31], kotlin-nosql[32] and KodeinDB[33]. The problems encountered with these libraries where that for Realm, I could not manage to get the code to work, even when following the guides line by line. Moreover, one of the solutions suggested in the Realm website was to start a hybrid project for iOS and android to get Realm to work[34]. Given the limited time I had to complete the project and my non-existing experience with iOS this was not a viable option.

About the other two libraries, they seemed promising. Their websites explained how to use them and what could they do, but this was the problem: they could not do very much. Both of these libraries are in a very early version (alpha or beta) and this means that their functionalities are very limited and they are not stable versions. Another reason that I did not realise until later is that even though NoSQL databases are great for simple queries, they are much less efficient when performing complex queries with multiple parameters and conditions. Join queries are quite slow as well. For these reasons and the state of the current libraries, SQLite was chosen as the database library I would use for the project.

The second obstacle was a consequence that came after the first obstacle. Since it had been a long time that I was trying to find a NoSQL library that worked well creating and running a local database on the device, a lot of development time was spent and the closed calendar risk manifested. All the tasks that were planned to be done were not possible due to time restrictions. As a consequence, I had to made some cuts to deliver most of the functionalities. I cut the less critical ones that would make the project unique and gave it their own space in the market. The cut functionalities are: Spanish translation, dice roller functionality, life counter functionality and material design animations. The budget alternatives functionality (*Budgetfy*) and the search combos in deck functionalities have also been cut, although cheaper alternatives to a card can be searched using the card search *similar* *to* field and the *price* fields.

Chapter 7

Budget

7.1 Identification of Costs and Estimate

In this section the different tasks will be matched with the corresponding team roles that have to perform them, and using an estimate hourly salary for each role an estimate cost of the full project (in workforce) will be calculated.

For salary references, *LinkedIn Salary* [35] has been used. This website is useful to scope the average salary of any job on the market. The salaries found on *LinkedIn Salary* are yearly salaries, so to make the transformation to hour salaries years of 1,824 hours will be assumed (information calculated as 1 year having 228 work days on average and 8 hour shifts).

In addition to its base cost, a 30% of the employees' base salary is added to the total to pay for social security for the employee. Find below on Table 7.1 the calculations of every salary by employee.

Role	Year Salary (€)	Salary/h (€)	Salary/h + SS (€)
Project Manager	40,000	21.92	28.50
Market Analyst	26,300	14.41	18.74
Software Architect	44,800	24.56	31.92
UI Designer	28,000	15.35	19.95
Programmer	26,000	14.25	18.53
Tester	22,300	12.22	15.89

Table 7.1: Hourly salary average per employee role on the project.

7.1.1 Costs by Task

On Table 7.2 you can find the different tasks of the project and the implicated employees on each of them, with the specific hours. The final cost is calculated per task and then added to obtain the total cost of the project.

Code	Time (h)	PM	MA	SA	UI	PR	TE	Cost (€)
PM	35							
PM1	20	10	10					472.40
PM2	10	10						285.00
PM3	5	5						142.50
IC	40							
IC1	10					8	2	180.02
IC2	20				20			399.00
IC3	10			10				319.20
DO	40							
DO1	15					15		277.95
DO2	15					15		277.95
DO3	10					10		185.30
CS	90							
CS1	25					24	1	460.61
CS2	15					13	2	272.67
CS3	30					28	2	550.62
CS4	20					19	1	367.96
CV	35							
CV1	30					29	1	553.26
CV2	5					5		92.65
DB	125							
DB1	20					19	1	367.96
DB2	20					19	1	367.96
DB3	20					19	1	367.96
DB4	30					28	2	550.62
DB5	10					9	1	182.66
DB6	15					14	1	275.31
DB7	10					9	1	182.66
CF	24							
CF1	10					9	1	182.66
CF2	14					13	1	256.78
DC	66							
DC1	6	6						171.00
DC2	60	20		10	10	10	10	1,432.90
Total	455	51	10	20	30	315	29	9,175.56

Table 7.2: Task Cost Summary Table

7.1.2 General Costs

General costs are costs that do not depend on the number of employees and are fix. This section will break down the general costs for the project and estimate its value.

The first cost would be the computer used by me, the author of the project. The computer was bought by parts and the total price of the parts was 1,600€. To calculate the amortization of a product within the project the formula used is

$$\frac{Cost(euro)}{Product\ Life(y) \times Year\ workdays \times Daily\ hours(h)} \times Project\ Duration(h)$$

The average useful life time for a computer is of 5 years, and as stated before on the document the average year has 228 work days. The estimation is to work 4 hours each day. This leaves the calculation as $\frac{1600}{5\times228\times4} \times 455 = 159.65$ €.

The other cost for the project would be the place for the employees to work. The best option for a short duration project like this would be to rent a workspace office. In *Aurea*'s website [36] we can see that a co-working rental for 6 hours per day costs 149€ a month. Since the project lasts from the 20th of september to the 19th of April approximately, we can state that seven months of rent have to be paid, which are 1043€ in total for renting a workspace for the project.

When we add these two payments we obtain that the total amount spent on the general costs of the project is 1,202.65€.

7.1.3 Contingency Costs

Contingency costs are budget added to the total costs to pay for unexpected items needed or problems encountered. To calculate this a percentage of the total project costs is used. This percentage is usually between 10 and 20%, so 15 will be used for reference in this project. The percentage is applied to the total cost of the project, that is the tasks cost and the general costs.

$$(9,175.56 + 1,202.65) \times 0.15 = 1,556.73$$

The contingency costs are 1,556.73€.

7.1.4 Risk Management Costs

The risk management costs are calculated using the risk management estimate times. You can find them at Table 6.1. Two of the risks on the table (technology inexperience and bugs) will be resolved by a programmer, so that will be the hour cost used to multiply the hours of resolution. Closed calendar should be resolved by the project manager. The cost for risk management are calculated below.

$$40 \times 18.53 + 10 \times 28.50 = 1026.20$$

The total cost is 1,026.20€.

7.1.5 Final Budget Estimate

The total cost of the project is calculated below.

$$9,175.56 + 1,202.65 + 1,556.73 + 1,026.20 = 12,961.14$$

The final cost for the project is 12,961.14€.

7.2 Budget Management

To keep track of how the budget is spent and to also check if the project is exceeding the initially planned budget an excel sheet is going to be used. You can find how the excel used looking at Tables 7.3, 7.4.

Code	Cost estimate (h)	Cost estimate (€)	Real cost (h)	Real cost (€)	Difference (h)	Difference (€)
PM	35.00	899.90	34.00	871.40	1.00	28.50
PM1	20.00	472.40	20.00	472.40	0.00	0.00
PM2	10.00	285.00	8.00	228.00	2.00	57.00
PM3	5.00	142.50	6.00	171.00	-1.00	-28.50
IC	40.00	898.22	29.00	754.70	11.00	143.52
IC1	10.00	180.02	4.00	76.40	6.00	103.62
IC2	20.00	399.00	10.00	199.50	10.00	199.50
IC3	10.00	319.20	15.00	478.80	-5.00	-159.60
DO	40.00	741.20	30.00	555.90	10.00	185.30
DO1	15.00	277.95	25.00	463.25	-10.00	-185.30
DO2	15.00	277.95	5.00	92.65	10.00	185.30
DO3	10.00	185.30	0.00	0.00	10.00	185.30
CS	90.00	1,651.86	46.00	864.87	44.00	786.99
CS1	25.00	460.61	10.00	186.44	15.00	274.17
CS2	15.00	272.67	25.00	472.75	-10.00	-200.08
CS3	30.00	550.62	6.00	112.32	24.00	438.30
CS4	20.00	367.96	5.00	93.36	15.00	274.60
CV	35.00	645.91	18.00	335.06	17.00	310.85
CV1	30.00	553.26	16.00	298.00	14.00	255.26
CV2	5.00	92.65	2.00	37.06	3.00	55.59
DB	125.00	2,295.13	18.00	336.58	107.00	1,958.55
DB1	20.00	367.96	6.00	112.04	14.00	255.93
DB2	20.00	367.96	2.00	37.35	18.00	330.62
DB3	20.00	367.96	0.00	0.00	20.00	367.96
DB4	30.00	550.62	0.00	0.00	30.00	550.62
DB5	10.00	182.66	0.00	0.00	10.00	182.66
DB6	15.00	275.31	10.00	187.20	5.00	88.11
DB7	10.00	182.66	0.00	0.00	10.00	182.66
CF	24.00	439.44	0.00	0.00	24.00	439.44
CF1	10.00	182.66	0.00	0.00	10.00	182.66
CF2	14.00	256.78	0.00	0.00	14.00	256.78
DC	66.00	1,603.90	50.00	1,276.87	16.00	327.03
DC1	6.00	171.00	10.00	285.00	-4.00	-114.00
DC2	60.00	1,432.90	40.00	991.87	20.00	441.03

Table 7.3: Budget management sheet 1.

Total Task Estimate (€)	Total Task Cost (€)	Difference (€)
9,175.56	4,995.38	4,180.18
Total General Estimate (€)	Total General Cost (€)	Difference (€)
1,202.65	1,202.65	0.00
Total Contingency Estimate (€)	Total Contingency Cost (€)	Difference (€)
1556.73	1556.73	0.00
Total Risk Management Estimate (€)	Total Risk Management Cost (€)	Difference (€)
1,026.2	1,026.2	0.00
Total Final Cost Estimate (€)	Total Final Cost Cost (€)	Difference (€)
12,961.14	8,780.96	4,180.18

Table 7.4: Budget management sheet 2.

7.3 Budget Changes

As commented before on this document, all of the obstacles encountered were expected when the risks were predicted, and thus they were taken into account when assigning hours to tasks and risk management. Hence the budget for the project remains unchanged.

Chapter 8

Sustainability

After taking the EDINSOST survey, I have realized that I have to learn more about sustainability, because the field is larger and deeper that what I thought it was. I first imagined it was about using less materials and avoiding unnecessary expenses but it goes a long way. The part I was less aware of is the social part of sustainability. The social part is as important, or arguably more important than the economic part and the environmental part, so this self-evaluation has been very useful by helping me be more aware of the importance of social sustainability in projects.

8.1 Social Sustainability

Personal growth I think this project will help me be more organised with the projects I perform in the future and will allow me to set a good work methodology that I can use in my job. I consider myself a not very organised person when referring to work or projects and surely this project will help me improve my methods.

State of the art and improvements Currently similar applications do a very good work on deck building and card search, but I think that this project will improve the experience of users by bringing to mobile multiple features only available in web applications at the moment that are huge quality of life improvements to the existing applications.

Is there a need for the project? As a Magic: The Gathering player I have used lots of applications, including the best rated applications for deck building, but the current applications are always missing very useful functionalities. I think this justifies the need to make an alternative to the current applications on the market that can offer all the functionalities that the others are missing.

8.2 Economic Sustainability

Cost estimate for the project I think that I did a good job when estimating the costs of the project. I read the documentation to learn about what are the costs for a project like mine and explained all the different parts and why they are needed or why they have to be taken into account. I am happy with the result of my budget calculation.

Economic improvement compared with other solutions Firstly, the application will have no premium membership, users will have access to all functionalities without any pay wall. This already gives users some incentive to try the application over other alternatives. Secondly, using the card search with the *similar to* and the *price* options, the user can adapt a deck to any budget. This is very useful for users that want to save money and no other app has a similar functionality.

If a new application that implements any of the unique functionalities I implement would be released during development, it would impact the viability of this project creating competition in the market.

8.3 Environmental Sustainability

Environmental impact of the project The project is strictly digital, and will not use a central server, so the environmental impact it will have is only the materials used during development and the energy consumed, and the energy consumed by the devices that will use the application once it is distributed. I think it is a reasonable cost for a project like this one.

Reducing the impact of the project There are not many ways to reduce the environmental impact for a project like this. The computer used to make the project is not a new computer, instead I am using a 2 year old computer.

State of the art I do not think this is applicable to this project. I have no information on how other solutions are managing environmental sustainability. In terms of implementation, the main way of minimising environmental impact would be to implement the application as efficiently as possible to reduce processing and, consequently, energy consumed. If I built this project again, and with more development time available, I would focus more on improving the application's efficiency, since some of the algorithms can be improved having more time.

For users that use analogical methods to take notes and conceptualise when designing decks, using this application would reduce the amount of paper used in that process.

Chapter 9

Laws and Regulations

This chapter goes through the laws and/or regulations that this project should take into account during development.

Since the application built in this project is intended to work offline and without an account login, no laws apply to this development. If the app required the user to login into an account and use personal credentials the Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)[37] would have to be followed, but this is not the case.

The application uses intellectual property (*Magic: the Gathering* cards) that is copyrighted by the company *Wizards of the Coast*. Fortunately, there is no need to look specifically to the laws of copyrighted material or fair use, because *Wizards of the Coast* has a clear policy that defines what can be done in fan made content[38]. This policy mainly states that the access to the copyrighted content should be free and that it should be clearly stated that what the user is using is an unofficial tool.

To comply with these requirements the final application should have an *About the app* screen that states that it is unofficial and is not sponsored by *WOTC*. Also, all of the application's monetising should be implemented without prohibiting access to specifically the cards themselves (maybe by making some functionalities premium, or implementing advertisements in-app).

Regarding the font mana[39], which is used for the symbols of the search interfaces, the font keyrune[40], used for the commander symbol, and the Commander Spellbook database[41], all three of these resources are licensed under the MIT License[42].

AnyChart[22] has multiple licences available, including paid licenses. For academic and non-profit projects they offer a free license. This is the one i asked for. At the moment of writing the license has not been supplied by AnyChart.

Chapter 10

Software Architecture

In this chapter we will go in depth on how the software developed is structured, looking at class and database diagrams and use cases. A general view is provided as well, to see how the app is conceptually structured.

10.1 General Structure

The general organisation for the application's code implements the Layered Architecture pattern. This design pattern consists of separating the program's code into three different categories, and in a layered structure such that one layer only interacts with the layers adjacent to it (see Figure 10.1). This structure's main purpose is to isolate the different layers as much as possible so that if one component has to change, only the adjacent layers would be affected by this change. It is a very sensible idea to use this pattern when working on a program that works through the full stack of program layers (front-end as well as back-end). This pattern also would allow to change some of the parts of the application to an external source, such as using an external database and changing the persistence layer to an interface to the database's API.

For this project specifically, the presentation layer is composed of the interface definition files (the files in XML format) and the activity/fragment files, which are the files that define how an interface works in android.

The business layer is composed of intermediate classes that process data for the presentation layer or request data to the persistence layer, such as the adapter classes for recycler views.

Finally, the persistence layer processes data from the database and communicates directly with it to request data. The helper class built for the database would be in this layer.

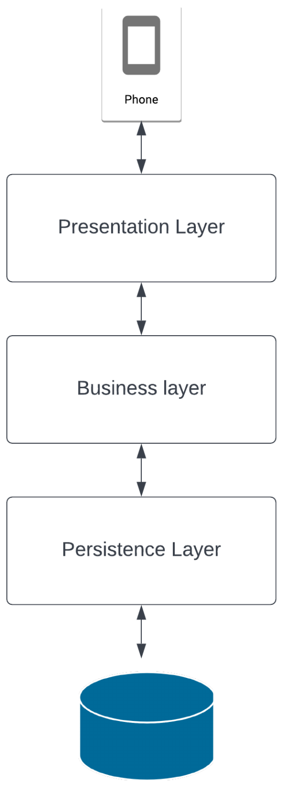


Figure 10.1: Layered architecture diagram.

10.2 Class Diagram

This diagram represents the main classes implemented in the application and their relationships. It purposefully omits classes such as the fragment/activities to not overload the diagram with unnecessary information. See Figure 10.2

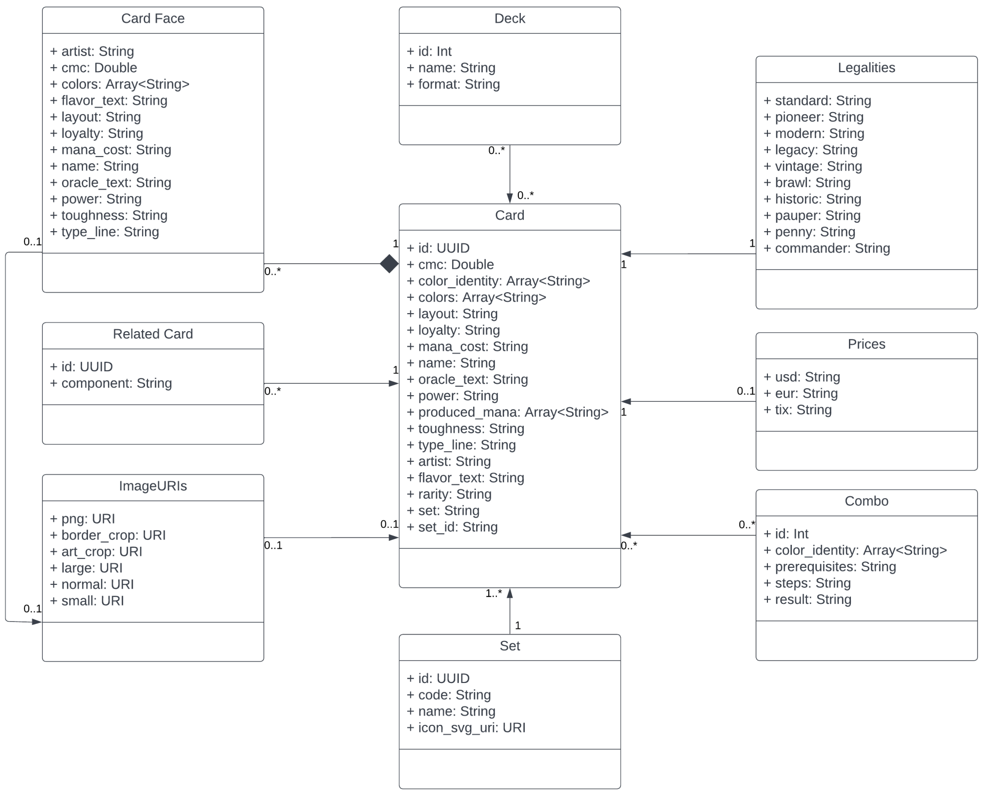


Figure 10.2: UML Class Diagram for the project's application.

10.3 Use Cases

In Figure 10.3 you can see the graph for the main use cases of this application.

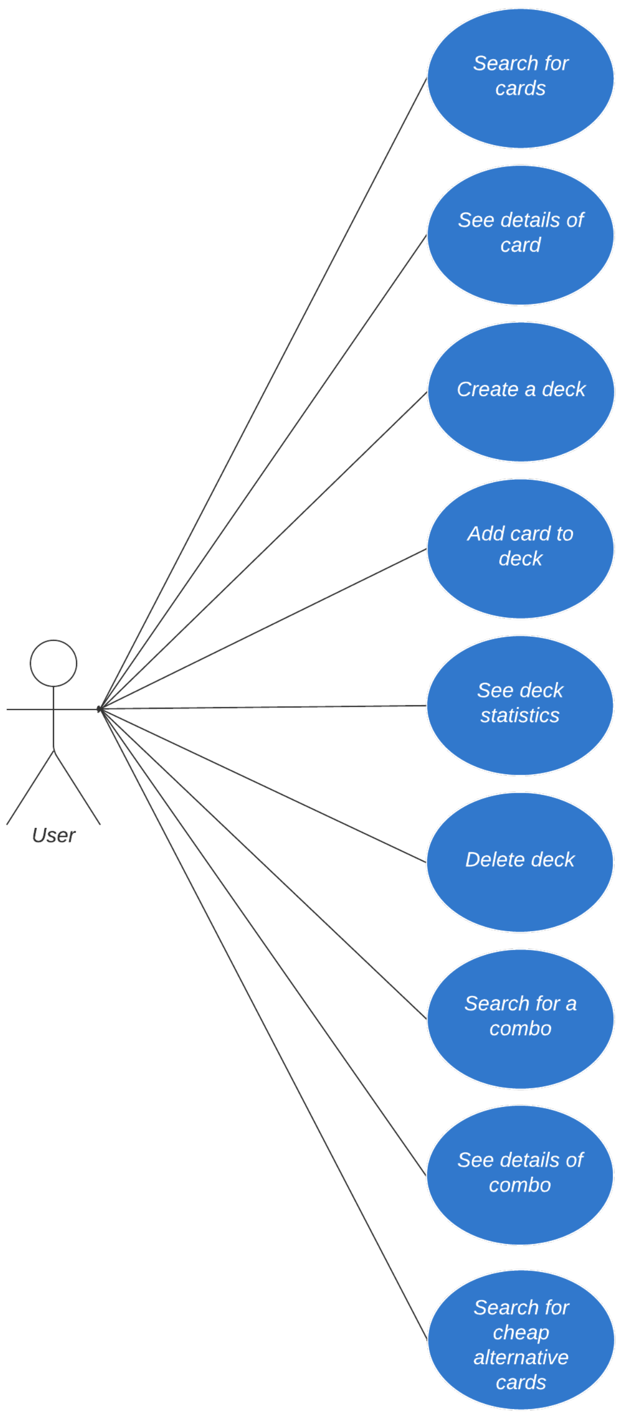


Figure 10.3: Main use cases for the application.

Here follows a list of use cases, which are the different scenarios where a user interacts with the application. Use cases are accompanied with a description that has the steps followed to successfully accomplish the action and with requirements that have to be fulfilled for this action to take place:

- Search for cards

Use case	Search for cards	Main actor	User
Precondition	Card database has card tuples		
Trigger	User presses the card search menu button		
Main scenario of success			
1. The search screen is shown 2. The user fills any search parameters they wish to filter by 3. The user presses the search button 4. The program takes the parameters specified and builds a search query 5. The search query is passed to database, and search results are returned 6. The search results are shown to the user in another screen as a list 7. The user can scroll freely through the list to see the results			
Extensions			
If the user has made a mistake when filling a search field, the program notifies the user and does not let him proceed			

- See details of a card

Use case	See details of card	Main actor	User
Precondition	A screen where data related to a card is shown		
Trigger	The user presses on the card information		
Main scenario of success			
1. The program queries the database about the specific card's data 2. the database returns the card's information 3. A screen with the card's information is shown to the user.			
Extensions			
Not all ocurrences of card related data lead to this action being possible. Two examples that do would be the card list view and the deck edit view			

- Create a deck

Use case	Create a deck	Main actor	User
Precondition	No precondition		
Trigger	the user is in the decks screen		
Main scenario of success			
1. The user presses the add deck button 2. A screen is shown that allows the player to select a format and name for the deck 3. The user fills the input fields 4. The user presses the create deck button 5. The program creates an insert statement for the deck with the parameters given by the player 6. The statement is sent to the database and the deck is created 7. The user is presented with the deck edit screen			
Extensions			
In case of the player choosing the commander format, the create deck screen unlocks another input field that allows to set a commander for the deck			

- Add a card to a deck

Use case	Add card to deck	Main actor	User
Precondition	The user has created a deck and is in the edit screen		
Trigger	No trigger. User has to be on deck edit screen		
Main scenario of success			
1. The user searches for a card with the text input field 2. The program provides suggestions for the name written on the field 3. The user selects one of the suggestions 4. The selected card is added to the deck's list			
Extensions			

- See deck statistics

Use case	See deck statistics	Main actor	User
Precondition	A deck is already created and the user is on deck edit view		
Trigger	the user presses the statistics menu button		
Main scenario of success			
1. The program calculates statistics based on the cards already in the deck 2. The statistics are shown to the user on the statistics view			
Extensions			

- Delete deck

Use case	Delete deck	Main actor	User
Precondition	A deck has already been created		
Trigger	the user selects the option to delete a deck		
Main scenario of success			
1. A confirmation dialog pops up to confirm if the deck has to be deleted 2. the user presses confirm 3. The program processes the order and deletes the deck from database 4. the user is returned to the deck list view			
Extensions			

- Search for a combo

Use case	Search for a combo	Main actor	User
Precondition	The database for combos has been initialised		
Trigger	The user presses the search combo menu button		
Main scenario of success			
1. The search screen is shown 2. The user fills any search parameters they wish to filter by 3. The user presses the search button 4. The program takes the parameters specified and builds a search query 5. The search query is passed to database, and search results are returned 6. The search results are shown to the user in another screen as a list 7. The user can scroll freely through the list to see the results			
Extensions			
If the user has made a mistake when filling a search field, the program notifies the user and does not let him proceed			

- See details of a combo

Use case	See details of combo	Main actor	User
Precondition	A screen where data related to a combo is shown		
Trigger	The user presses on the combo information		
Main scenario of success			
1. The program queries the database about the specific combo's data 2. the database returns the combo's information 3. A screen with the combo's information is shown to the user.			
Extensions			
Not all ocurrences of combo related data lead to this action being possible. An example that does would be the combo list view			

- Search for cheaper alternative cards for a deck

Use case	Search for cheap alternative cards	Main actor	User
Precondition	A deck has been created		
Trigger	the user presses the "Budgetfy" button		
Main scenario of success			
1. A dialog asking for a price limit per individual card is shown to the user 2. The user introduces a value 3. the user presses the button to accept 4. The program makes a list of all the cards that exceed the budget provided by the user 5. the program searches for similar cards that do not exceed the provided budget for each of the cards in the list 6. The program displays the cards to the user in a list view. Three cards maximum per card searched 7. the user can select any of the alternatives to replace the card in the deck			
Extensions			
If no cheaper alternatives are found, a dialog notifies the user			

10.3.1 Sequence Diagrams

For this section two sequence diagrams will be displayed to get a general knowledge of how the application operates. The two use cases drawn will be the Search for cards and the Search for combos.

Search for cards

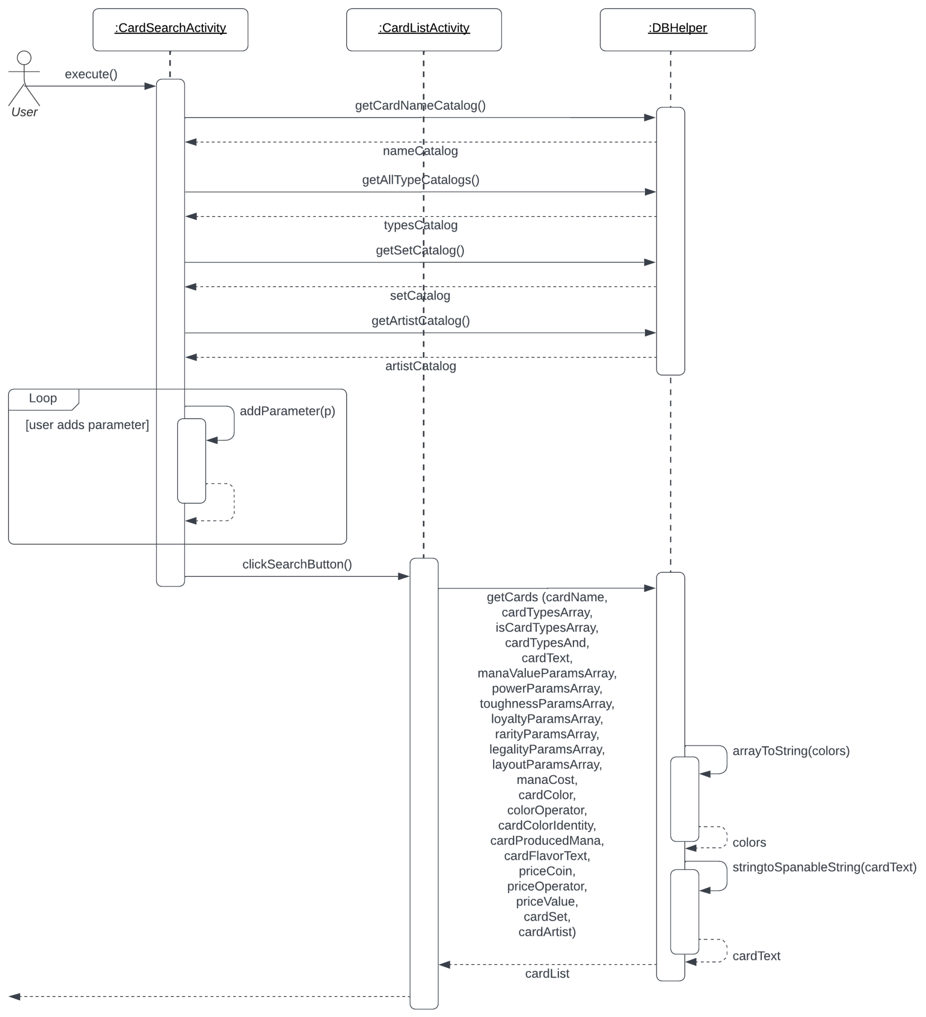

:CardSearchActivity
:CardListActivity
:DBHelper
User
execute()
getCardNameCatalog()
nameCatalog
getAllTypeCatalogs()
typesCatalog
getSetCatalog()
setCatalog
getArtistCatalog()
artistCatalog
Loop
[user adds parameter]
addParameter(p)
clickSearchButton()
getCards (cardName, cardTypesArray, isCardTypesArray, cardTypesAnd, cardText, manaValueParamsArray, powerParamsArray, toughnessParamsArray, loyaltyParamsArray, rarityParamsArray, legalityParamsArray, layoutParamsArray, manaCost, cardColor, colorOperator, cardColorIdentity, cardProducedMana, cardFlavorText, priceCoin, priceOperator, priceValue, cardSet, cardArtist)
arrayToString(colors)
colors
stringtoSpanableString(cardText)
cardText
cardList

Figure 10.4: Search for cards sequence diagram.

Search for combos

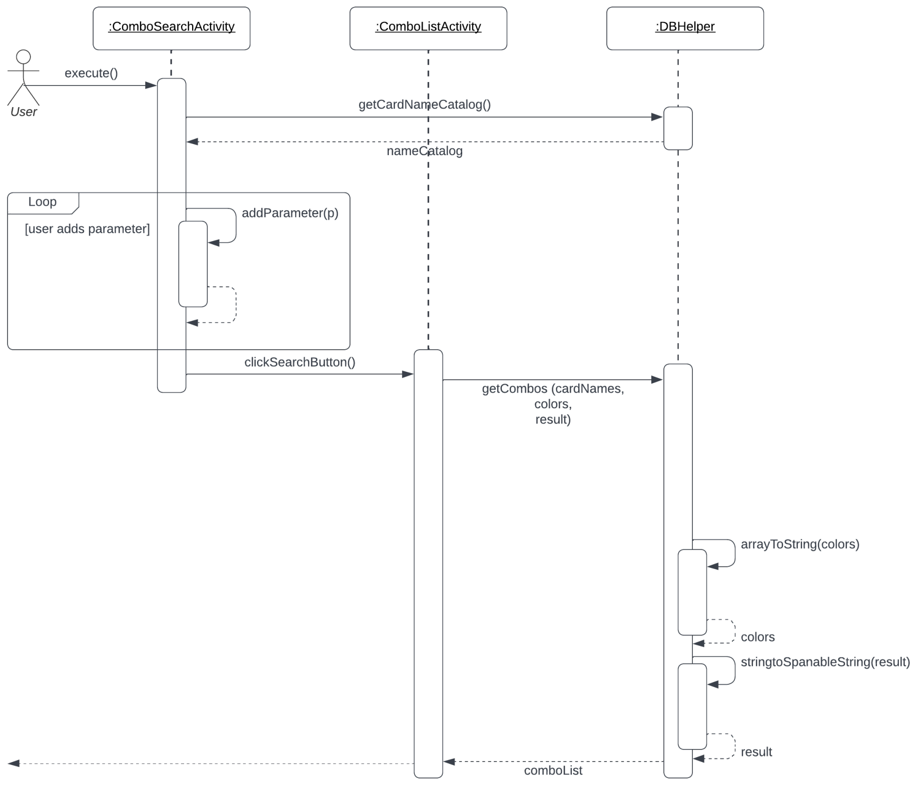
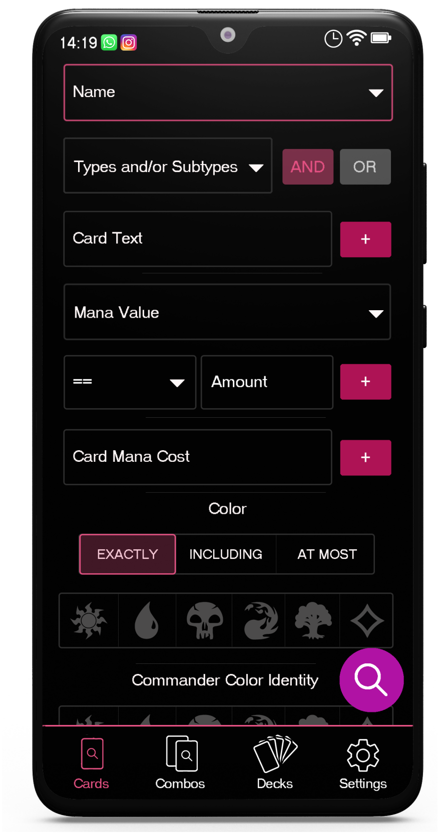


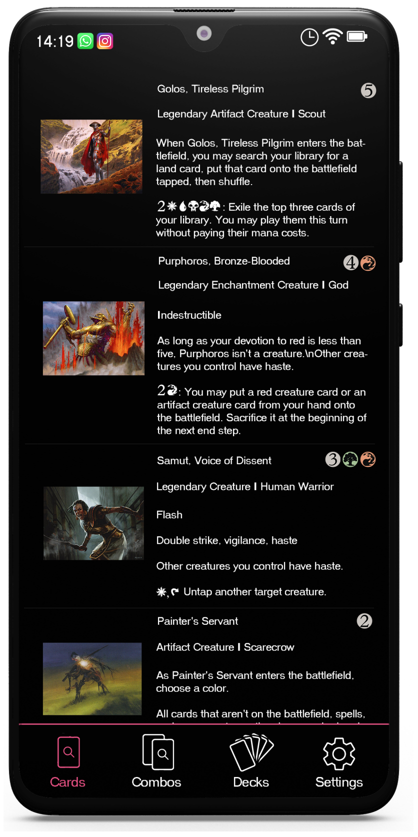
Figure 10.5: Search for combos sequence diagram.

10.4 Interface Design

In this section you will find the mock-ups for the different views of the application.



(a) Card Search screen.



14:19

Golos, Tireless Pilgrim 5

Legendary Artifact Creature I Scout

When Golos, Tireless Pilgrim enters the battlefield, you may search your library for a land card, put that card onto the battlefield tapped, then shuffle.

2: Exile the top three cards of your library. You may play them this turn without paying their mana costs.

Purphoros, Bronze-Blooded 4

Legendary Enchantment Creature I God

Indestructible

As long as your devotion to red is less than five, Purphoros isn't a creature.\nOther creatures you control have haste.

2: You may put a red creature card or an artifact creature card from your hand onto the battlefield. Sacrifice it at the beginning of the next end step.

Samut, Voice of Dissent 3

Legendary Creature I Human Warrior

Flash

Double strike, vigilance, haste

Other creatures you control have haste.

, Untap another target creature.

Painter's Servant 2

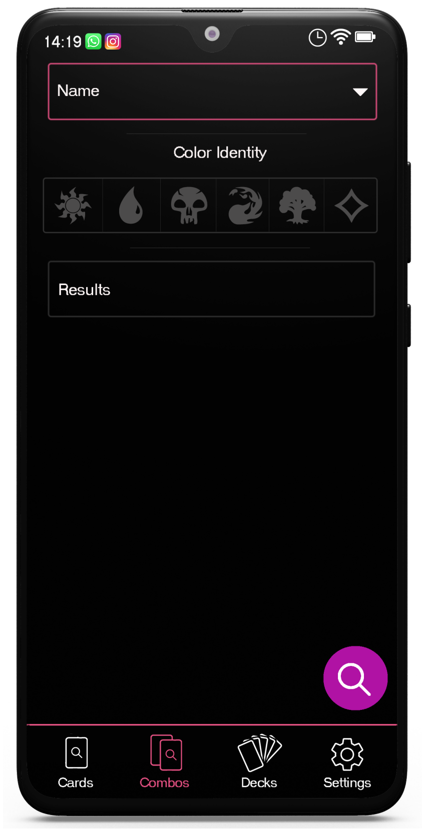
Artifact Creature I Scarecrow

As Painter's Servant enters the battlefield, choose a color.

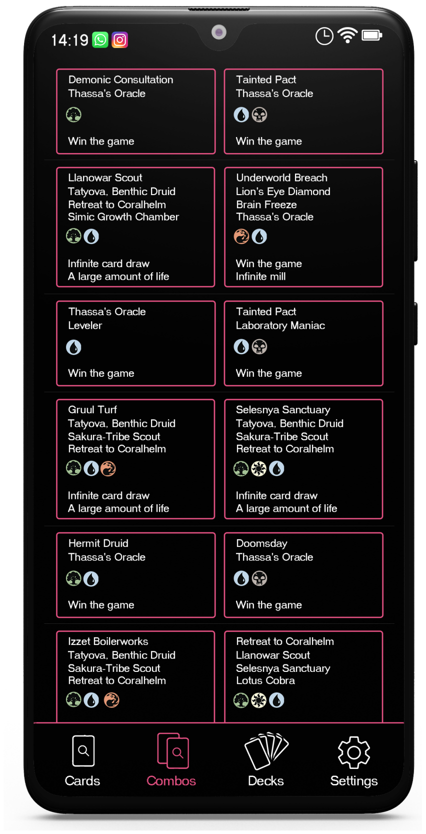
All cards that aren't on the battlefield, spells,

(b) Card List screen.

Figure 10.6: Card Search related views.



(a) Combo Search screen.



14:19

Demonic Consultation
Thassa's Oracle
Win the game

Tainted Pact
Thassa's Oracle
Win the game

Llanowar Scout
Tatyova, Benthic Druid
Retreat to Coralhelm
Simic Growth Chamber
Infinite card draw
A large amount of life

Underworld Breach
Lion's Eye Diamond
Brain Freeze
Thassa's Oracle
Win the game
Infinite mill

Thassa's Oracle
Leveler
Win the game

Tainted Pact
Laboratory Maniac
Win the game

Gruul Turf
Tatyova, Benthic Druid
Sakura-Tribe Scout
Retreat to Coralhelm
Infinite card draw
A large amount of life

Selesnya Sanctuary
Tatyova, Benthic Druid
Sakura-Tribe Scout
Retreat to Coralhelm
Infinite card draw
A large amount of life

Hermit Druid
Thassa's Oracle
Win the game

Doomsday
Thassa's Oracle
Win the game

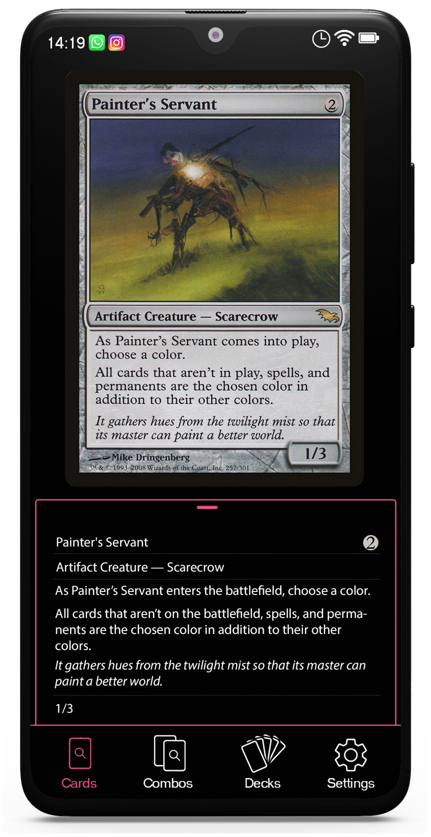
Izzet Boilerworks
Tatyova, Benthic Druid
Sakura-Tribe Scout
Retreat to Coralhelm

Retreat to Coralhelm
Llanowar Scout
Selesnya Sanctuary
Lotus Cobra

Cards
Combos
Decks
Settings

(b) Combo List screen.

Figure 10.7: Combo Search related views.



Painter's Servant 2

Artifact Creature — Scarecrow

As Painter's Servant comes into play, choose a color.

All cards that aren't in play, spells, and permanents are the chosen color in addition to their other colors.

It gathers hues from the twilight mist so that its master can paint a better world.

1/3

Mike Dringenberg

Painter's Servant 2

Artifact Creature — Scarecrow

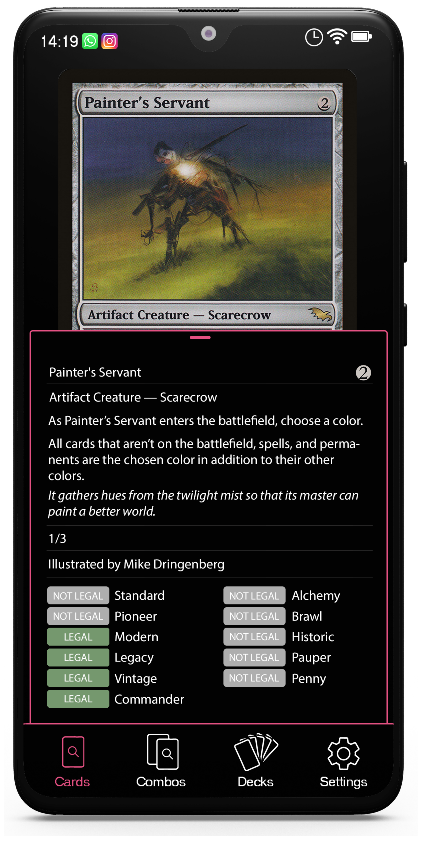
As Painter's Servant enters the battlefield, choose a color.

All cards that aren't on the battlefield, spells, and permanents are the chosen color in addition to their other colors.

It gathers hues from the twilight mist so that its master can paint a better world.

1/3

(a) Card Details screen 1.



Painter's Servant 2

Artifact Creature — Scarecrow

Painter's Servant 2

Artifact Creature — Scarecrow

As Painter's Servant enters the battlefield, choose a color.

All cards that aren't on the battlefield, spells, and permanents are the chosen color in addition to their other colors.

It gathers hues from the twilight mist so that its master can paint a better world.

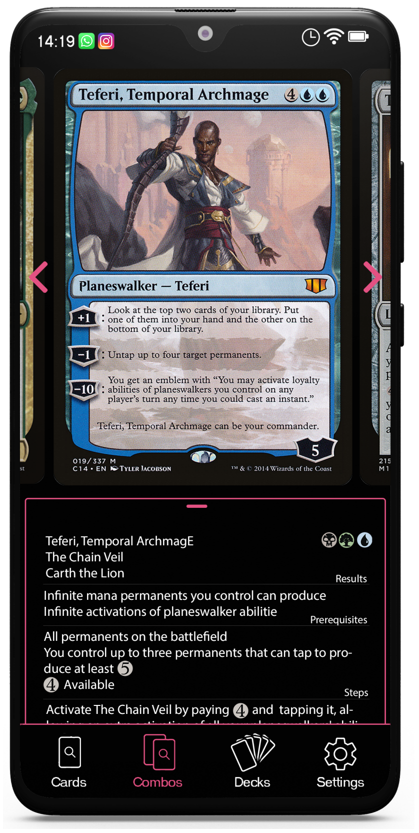
1/3

Illustrated by Mike Dringenberg

NOT LEGAL	Standard	NOT LEGAL	Alchemy
NOT LEGAL	Pioneer	NOT LEGAL	Brawl
LEGAL	Modern	NOT LEGAL	Historic
LEGAL	Legacy	NOT LEGAL	Pauper
LEGAL	Vintage	NOT LEGAL	Penny
LEGAL	Commander		

(b) Card Details screen 2.

Figure 10.8: Card Details view.


14:19
Teferi, Temporal Archmage
Planeswalker — Teferi
+1: Look at the top two cards of your library. Put one of them into your hand and the other on the bottom of your library.
−1: Untap up to four target permanents.
−10: You get an emblem with "You may activate loyalty abilities of planeswalkers you control on any player's turn any time you could cast an instant."
Teferi, Temporal Archmage can be your commander.
5
019/337 M
C14 • EN Tyler Jacobson
™ & © 2014 Wizards of the Coast
Cards
Combos
Decks
Settings

Teferi, Temporal ArchmagE
The Chain Veil
Carth the Lion

Results

Infinite mana permanents you control can produce
Infinite activations of planeswalker abilitie

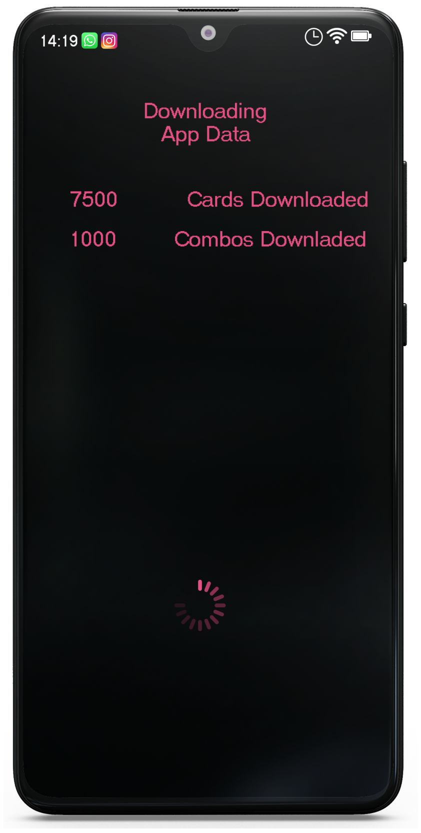
Prerequisites

All permanents on the battlefield
You control up to three permanents that can tap to produce at least ⑤
④ Available

Steps

Activate The Chain Veil by paying ④ and tapping it, al-

(a) Combo Details screen.

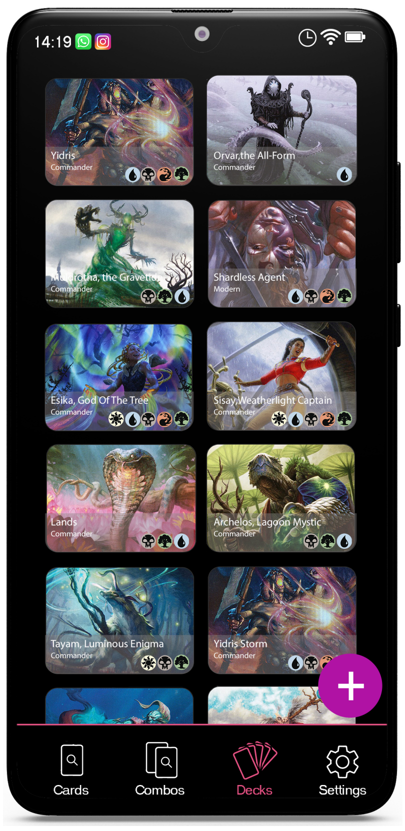


Downloading
App Data

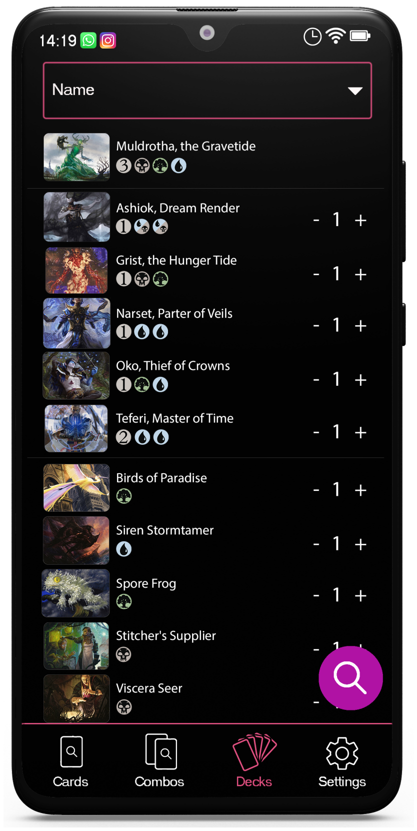
7500	Cards Downloaded
1000	Combos Downloaded

(b) Card Download Screen.

Figure 10.9: Combo Details and Card Download views.



(a) Decks List screen.



Muldrotha, the Gravetide
Ashiok, Dream Render - 1 +
Grist, the Hunger Tide - 1 +
Narset, Parter of Veils - 1 +
Oko, Thief of Crowns - 1 +
Teferi, Master of Time - 1 +
Birds of Paradise - 1 +
Siren Stormtamer - 1 +
Spore Frog - 1 +
Stitcher's Supplier
Viscera Seer

(b) Edit Deck screen.

Figure 10.10: Deck related views.

10.4.1 Application Flow

The bottom navigation bar that can be seen in the mock-ups allows the user to swap between the four sections of the application at any time (and return to the previous state of the other section). These sections are: Cards, Combos, Decks and Settings. Because of this, the four sections have their own independent flow diagram to navigate inside of each section. Since the settings screen is a simple view, its flow diagram is only a screen and will not be shown. Here follow the other section's flow diagrams.

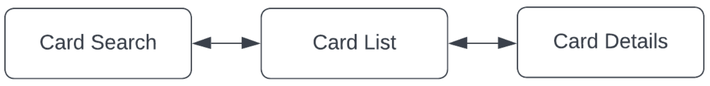


Figure 10.11: Cards section flow diagram.

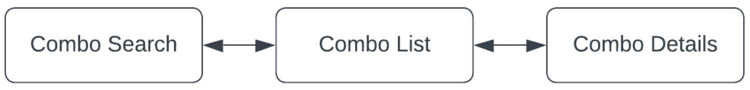


Figure 10.12: Combos section flow diagram.

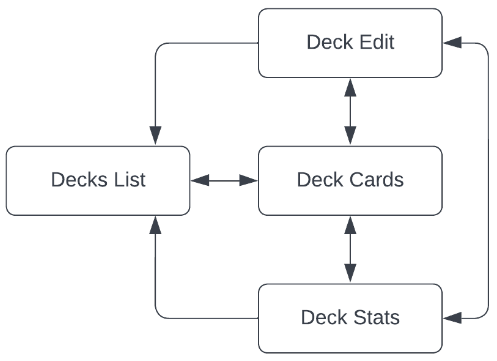


Figure 10.13: Decks section flow diagram.

For future work, more navigation options such as navigability between separate sections (whenever it makes sense) could be implemented. For example, being able to access card details directly from the combo details section.

10.5 Database Design

This section covers the design for the database by providing a diagram of the relationships between the main tables, and then a list of the different tables that result from the diagram analysis. You can see the diagram looking at Figure 10.14.

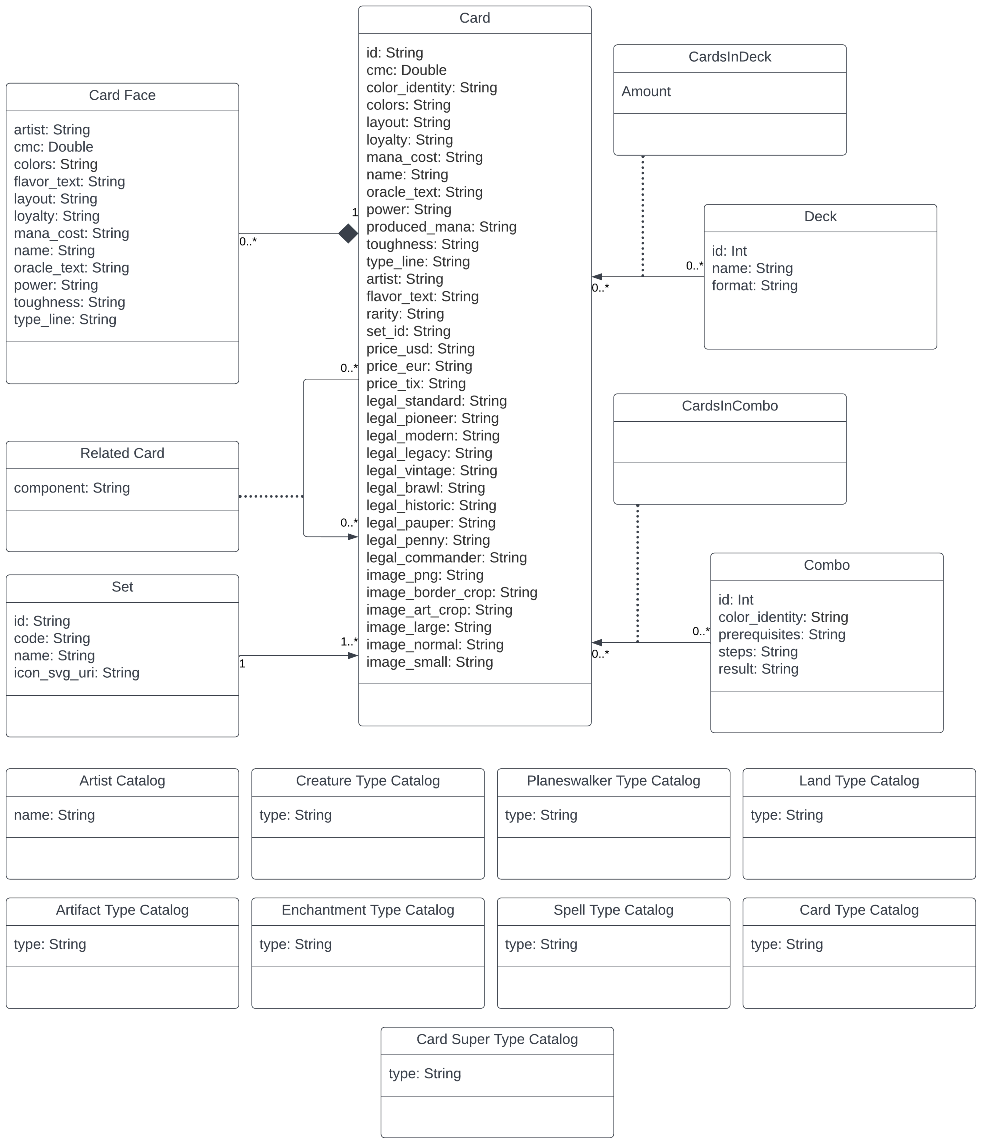


Figure 10.14: Database Diagram for the project's application.

The diagram translates to the table definitions for the application's database. The notation for this tables is the following:

Table Name (<u>**PK**</u>, **FK**, additional data)
FK references Table Name

Table list:

Card (<u>**id**</u>, cmc, color_identity, colors, layout, loyalty, mana_cost, name, oracle_text, power, produced_mana, toughness, type_line, artist, flavor_text, rarity, **set_id**, price_usd, price_eur, price_tix, legal_standard, legal_pioneer, legal_modern, legal_legacy, legal_vintage, legal_brawl, legal_historic, legal_pauper, legal_penny, legal_commander, image_png, image_border_crop, image_art_crop, image_large, image_normal, image_small)
set_id references Set

CardFace (**id**, **id_main_card**, artist, cmc, colors, flavor_text, layout, loyalty, mana_cost, name, oracle_text, power, toughness, type_line, image_png, image_border_crop, image_art_crop, image_large, image_normal, image_small)
id_main_card references Card

RelatedCard (**id_main**, **id_related**, component)
id_main references Card
id_related references Card

Set (**id**, code, name, icon_svg_uri)

Deck (**id**, name, format)

CardsInDecks (**id_card**, **id_deck**, amount, board)
id_card references Card
id_deck references Deck

Combo (**id**, color_identity, prerequisites, steps, result)

CardsInCombos (**id_card**, **id_combo**)
id_card references Card
id_combo references Combo

ArtistCatalog (**name**)

CreatureTypeCatalog (**type**)

PlaneswalkerTypeCatalog (**type**)

LandTypeCatalog (**type**)

ArtifactTypeCatalog (**type**)

EnchantmentTypeCatalog (**type**)

SpellTypeCatalog (**type**)

CardTypeCatalog (**type**)

CardSuperTypeCatalog (**type**)

Chapter 11

Project Development

This chapter explains how the development of the project was made, which changes were made to adapt to obstacles encountered, and how some parts of the project where implemented.

11.1 Functionality Implementation

This section will provide a thorough explanation of how the more complex functionalities of the application are implemented. The objective of this is to allow the reader to understand the complexity of some of the functionalities.

11.1.1 Card Search

The card search functionality allows the user to filter the cards in the database with multiple filter options. Here follows a list of filtering options:

- **Name** The card's name
- **Types and/or Subtypes** The card's types and/or subtypes
- **Text** The card's oracle text, i.e. the text that describes the card's effect
- **Mana Value** The numeric value for the mana cost of the card
- **Power** The power of the card
- **Toughness** The toughness of the card
- **Loyalty** The loyalty of the card
- **Rarity** The rarity of the card
- **Legality** The legality of the card, and the format where that loyalty applies
- **Layout** The layout of the card
- **Mana Cost** The mana cost of the card
- **Color** Any combination of colors and their relation to filter the color of the card

- **Color Identity** Any combination of colors to filter the color identity of the card
- **Produced Mana** Any combination of colors to filter the produced mana of the card
- **Flavor Text** The flavor text of the card, i.e. text that contains a brief piece of lore from Magic: The Gathering
- **Price** The price of the card, in US Dollars, Euros or TIX, a coin used in a Magic: The Gathering online game
- **Set** The Set to which the card belongs to
- **Artist** The artist of the card

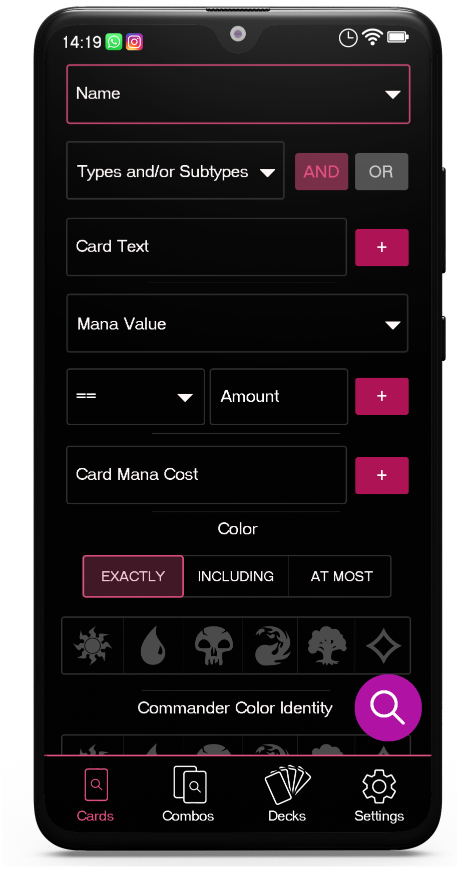


Figure 11.1: Card Search screen.

After the user presses the search button each parameter that the user has filled is added to a String in order. This String forms the SQL query.

For the card's name, it searches for cards that contain the String written by the user. When filling the input field, the user is suggested card names that contain the value the user is currently writing. To avoid too many suggestions at the same time, suggestions only appear when the input has 3 characters or more. When selecting a suggestion, the input field is auto-completed with the selected value.

Types and/or card types work in a similar way than the card's name. Suggestions are offered when writing on the field, but there is a main difference: The user can add multiple types and/or subtypes, and for each selected type and/or subtype choose if the resulting card will have or not have those types (individually). There is a toggle that allows the user to choose if the algorithm should search for cards that have all types selected (AND) or any of the selected (OR). When adding these parameters to the query String, they are chained with AND or OR instructions depending on user choice.

The search finds cards that contain all the words and symbols written in the text field, in any order. The card text field has a button next to it that allows the user to select a symbol, and writes the code for the symbol into the text field.

For the mana value, power, toughness and loyalty parameters, the user can select a mathematical operator (==, !=, ¿, ¡, etc.) and a numerical value. The user can add multiple restrictions for each of those parameters, and can select how those parameters are related between each other (AND, OR).

For the rarity, legality and layout parameters, field-specific options can be selected by the user. The user can add multiple restrictions for each of those parameters as well, and can select how those parameters are related between each other (AND, OR).

The mana cost field allows the user to write symbol codes to specify part/all of the mana cost of a card. To help the user, a button press shows a dialog with symbols that writes the selected symbol into the field. The function searches for cards that contain the input String in their mana cost.

For the color of the card, the user can select any combination of colors and the relation between those colors with a toggle: *Exactly*, meaning the card is all of the selected colors and no more, *Including*, meaning the card is at least the selected colors, and *At Most*, meaning that the card is at most the selected colors.

For the color identity of the card, the user can select any combination of colors and the function searches for cards that are at most those colors in their identity. This is because for color identity only this setting makes sense for game play purposes.

For the produced mana of the card, the user can select any combination of colors and the function searches for cards that include those colors in their produced mana. This is because for produced mana only this setting makes sense for game play purposes.

The search finds cards that contain all the words and symbols written in the flavor text field, in any order.

The card price works in the same way as the mana value, power, toughness and loyalty fields. The user selects an operator and inputs a numerical value for the price of the card. The user also has a toggle available to select which denomination is used when comparing the inputted price.

The Set field allows the user to select a set name. The field suggests Set names from database.

The Artist field works in the same way than the Set field.

After the search query is built and called, the resulting list is shown to the user in a secondary screen.

11.2 Combo Search

This functionality works the same way as the card search functionality, with the main difference being the parameters that the user has to fill in order to filter.

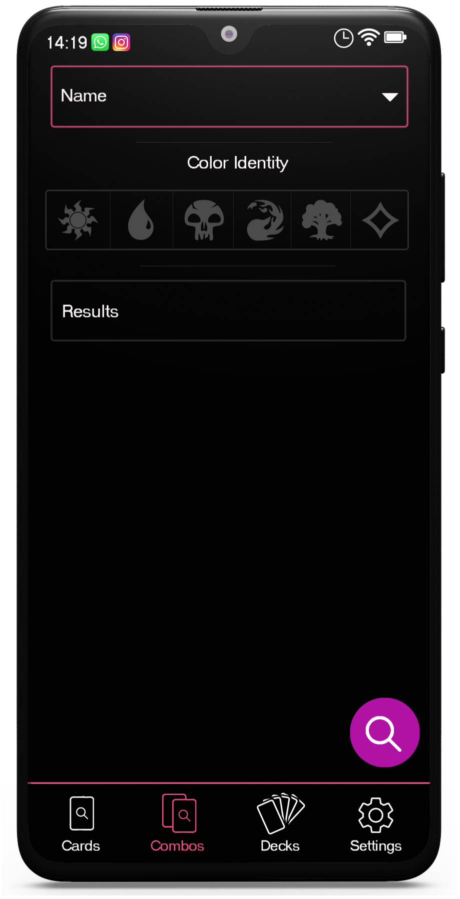


Figure 11.2: Combo Search view.

11.3 Work-arounds

This section explains the obstacles encountered that needed a work-around to solve them, either because of time limitations or because of the nature of the error.

11.3.1 Database Primary Keys

When building the database, the main idea was to use the UUID received by the Scryfall API that uniquely identifies each card as the primary key for the table on the database. The obstacle that was encountered was that the database was throwing a unique clause violation exception, implying that these UUIDs weren't unique. Performing a test, I found out that if I uset an integer that auto-incremented as a primary key, and inserted the unique id as a string with a unique clause on the table, the unique violation error did not pop up. This was very strange, but since I did not have more time to dedicate to this matter and I found an article that advocated against using strings as primary keys (and made really good arguments against it) I left the database tables with the auto-incremented integer primary key.

11.3.2 Combo Database

The database for combos[41] is hosted on Google Drive, and with the time available for the project learning how to use the Google Drive API was not a viable option. Because of this, the approach taken to solve this issue was to download the database as a CSV file, since JSON was not an available option, use csvjson[43] to create a JSON file, and manually edit the json file to erase unnecessary empty elements. This file was included in the app code.

Chapter 12

Conclusions

This was an ambitious project with many different functionalities and screens. Some of the functionalities proved to be much more difficult than expected, and that added to the obstacles encountered unfortunately forced the decision of cutting some functionalities off of the final product. The fact that I started working full time on a company also reduced considerably the time I had available to develop the application.

The initial idea of the application also changed during the development, leaning more into the search of cards and combos, and less into the deck building. This is because when deciding which functionalities to cut, I decided that the better functionalities to keep were the ones that separate the application from the other options on the market.

Overall, the performance of the application could be improved. Because of the limited time available, I could not optimise the application to a point I think it can be published, so if I were to publish this app to the store in the future, I would optimise it first to reduce unnecessary database calls and unnecessary iteration through variables.

12.1 Personal Conclusions

Developing this project has been a positive experience overall, and helped me understand better what a real project development looks like, especially which obstacles can be encountered and how to work around these problems.

I learned new technologies to develop this project, which I think will be very useful in my career, as well as the experience of implementing a big project on my own. I also learned about the importance of documentation. I do not like documenting projects, I prefer the coding of the application, but I understand now and acknowledge how important it is, especially early documentation used to design the project.

The greatest technical difficulty has been learning how *Android*'s components work and understanding their lifecycle, in order to properly use them. *Android* is constantly evolving and updating, and because of this many sources of information are outdated and no longer correct.

Overall I feel happy with the result of the project, even though it is not what I anticipated it would be. I think I learned a lot, and I look differently to real projects made by small teams, because I understand how complicated it is to do them. Since

I liked it, I will consider implementing the missing functionalities and publishing the application on *Google Play*.

12.2 Future Improvements

This section contains the different improvements that should follow after this project's development. The improvements have been grouped into three categories: User Interface for improvements regarding the interface, Functionalities and Features for improving any functionality or creating a new one, and Optimisation and Code Refactoring for optimising processes or cleaning/refactoring code.

12.2.1 User Interface

- **Add dialog to card download screen warning the user about download size**
- **General visual upgrades, such as colors.**
- **Adapting interface to multiple devices** The interface is responsive, but may display in an undesired way on certain devices.

12.2.2 Functionalities and Features

- **Search for combos in deck**
- **Budgetfy** Functionality that searches for cheaper alternatives to cards in a deck that exceed budget limit.
- **Include card face data as target for the card search** Currently the card search functionality only considers card data.
- **Implement field data recovery after navigating back to a screen**
- **Add options to order results by specific parameters to card search and combo search**
- **Implement the automatic updating of the database without the need of re-installing the app.**
- **Add rulings for cards to database and display them in card details**
- **Functionality to check if a deck is legal in any format**
- **Learn more about Google Drive's API to download the combo database**
- **New navigability options** For example, being able to select a card from a combo and viewing its details.

12.2.3 Optimisation and Code Refactoring

- **Optimise database searches by removing unnecessary data retrievals, or reducing amout of queries**
- **Implementing classes or interfaces with common functions to be called from different classes in the program**
- **Code minimising**
- **Improve error handling** For example, when a network error occurs when downloading cards.
- **Remove redundant fields in database**
- **Optimising queries in database**
- **Turn any hard-coded values into variables**

Glossary

card combo A combination of two or more cards that together form an infinite loop or a very powerful, game-ending effect. 14, 15

companion A companion app is an app that is designed to do various functionalities that the user may need to help them in building and/or playing Magic: The Gathering decks. For example, a companion can have card search, deck builder, dice simulator and life tracker functionalities. 21

metagame The metagame in a card game is the group of most played decks on a specific format. It usually corresponds with the top performing decks of the format. 21

staple A card that is so good in a format it is usually included automatically in any deck that can play it. 17

Acronyms

ASW Aplicacions i Serveis Web. 28

CSV Comma Separated Values. 70

IDI Interacció i Disseny d'Interfícies. 28

JSON Java Script Object Notation. 70

LOPDGDD Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales. 48

RNG Random Number Generator. 31

SLDS Software Lliure i Desenvolupament Social. 28

TCG Trading Card Game. 8

Bibliography

[1] Wizards of the Coast. *How to Play — Magic: The Gathering*. URL: https://magic.wizards.com/en/magic-gameplay.

[2] Wizards of the Coast. *Formats — Magic: The Gathering*. URL: https://magic.wizards.com/en/game-info/gameplay/rules-and-formats/formats.

[3] MTG Assist. *MTG Assist*. URL: https://www.mtgassist.com/index.php.

[4] Scryfall. *Scryfall Magic: The Gathering Search*. URL: https://scryfall.com/.

[5] Commander Spellbook. *Commander Spellbook: The Search Engine for EDH Combos*. URL: https://www.commanderspellbook.com/.

[6] EDHREC. *EDHREC*. URL: https://edhrec.com/.

[7] TappedOut. *TappedOut*. URL: https://tappedout.net/.

[8] Moxfield. *Moxfield*. URL: https://www.moxfield.com/.

[9] Archidekt. *Archidekt*. URL: https://archidekt.com/.

[10] SkillDevs. *ManaBox*. URL: https://play.google.com/store/apps/details?id=skilldevs.com.manabox.

[11] TopDecked Limited. *TopDecked MTG*. URL: https://play.google.com/store/apps/details?id=com.maritlabs.topdecked.mtg.

[12] Google. *Design — Material Design*. URL: https://material.io/design.

[13] Atlassian. *Trello*. URL: https://trello.com.

[14] GitHub. *GitHub: Where the world builds software*. URL: https://github.com/.

[15] Android. *Android*. URL: https://www.android.com/.

[16] JetBrains. *Kotlin Programming Language*. URL: https://kotlinlang.org/.

[17] SQLite. *SQLite Home Page*. URL: https://sqlite.org/index.html.

[18] Google. *Download Andoid Studio and SDK Tools*. URL: https://developer.android.com/studio/.

[19] MongoDB. *MongoDB*. URL: https://www.mongodb.com/.

[20] Overleaf. *Overleaf, Online LaTeX Editor*. URL: https://www.overleaf.com/.

[21] Lucid Software. *Homepage — Lucid*. URL: https://lucid.co/.

[22] AnyChart. *AnyChart/AnyChart-Android: AnyChart Android Chart is an amazing data visualization library for easily creating interactive charts in Android apps. It runs on API 19+ (Android 4.4) and features dozens of built-in chart types*. URL: https://github.com/AnyChart/AnyChart-Android.

[23] Inkscape. *Draw Freely — Inkscape*. URL: https://inkscape.org/es/.

[24] FIB UPC. *Interacció i Disseny d'Interfícies — Facultat d'Informàtica de Barcelona*. URL: https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/pla-destudis/assignatures/IDI.

[25] FIB UPC. *Aplicacions i Serveis Web — Facultat d'Informàtica de Barcelona*. URL: https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/pla-destudis/assignatures/ASW.

[26] FIB UPC. *Software Lliure i Desenvolupament Social — Facultat d'Informàtica de Barcelona*. URL: https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/pla-destudis/assignatures/SLDS.

[27] The GIMP Team. *GIMP - GNU Image Manipulation Program*. URL: https://www.gimp.org/.

[28] Adobe. *Software de gráficos vectoriales líder del sector — Adobe Illustrator*. URL: https://www.adobe.com/es/products/illustrator.html.

[29] Google. *Google Meet*. URL: https://meet.google.com/.

[30] Kmongo. *KMongo*. URL: https://litote.org/kmongo/.

[31] mongoDB. *Quick Start - Kotlin Multiplatform SDK — MongoDB Realm*. URL: https://docs.mongodb.com/realm/sdk/kotlin-multiplatform/quick-start/.

[32] cheptsov. *kotlin-nosql - master - Kotlin Resources*. URL: https://www.kotlinresources.com/library/kotlin-nosql/.

[33] Kodein. *Kodein-Framework*. URL: https://docs.kodein.org/kodein-framework/index.html.

[34] MongoDB. *Install Realm - Android SDK — MongoDB Realm*. URL: https://docs.mongodb.com/realm/sdk/android/install/#std-label-android-install.

[35] LinkedIn. *LinkedIn Salary*. URL: https://www.linkedin.com/salary/.

[36] Aurea. *Coworking en Barcelona*. URL: https://www.aureacoworking.com/precios-coworking/.

[37] BOE. *BOE.es - BOE-A-2018-16673 Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*. URL: https://boe.es/buscar/act.php?id=BOE-A-2018-16673.

[38] Wizards of the Coast. *Fan Content Policy — Wizards of the Coast*. URL: https://company.wizards.com/en/legal/fancontentpolicy.

[39] andrewgioia. *andrewgioia/mana: Magic: the Gathering mana symbol pictographic font*. URL: https://github.com/andrewgioia/Mana.

[40] Andrew Gioia. *Set Symbol Icons — Keyrune*. URL: https://keyrune.andrewgioia.com/icons.html.

[41] Commander Spellbook. *Commander Spellbook Database - Hojas de cálculo de Google*. URL: https://docs.google.com/spreadsheets/d/1KqyDRZRCgy8YgMFnY0tHSw_3jC99Z0zFvJrPbfm66vA/edit#gid=0.

[42] Opensource.org. *The MIT License — Open Source Initiative*. URL: https://opensource.org/licenses/MIT.

[43] Flatfile. *CSV to JSON - CSVJSON*. URL: https://csvjson.com/csv2json.

[44] Google. *Google Play icon design specifications — Android Developers*. URL: https://developer.android.com/google-play/resources/icon-design-specifications.

[45] Google. *Kotlin and Android — Android Developers*. URL: https://developer.android.com/kotlin.

[46] Google. *MTG Unofficial - Aplicaciones en Google Play*. URL: https://play.google.com/store/apps/details?id=io.carlol.mtgu.mobile.

[47] Google. *MTG Familiar - Aplicaciones en Google Play*. URL: https://play.google.com/store/apps/details?id=com.gelakinetic.mtgfam.

[48] Overleaf. *How to Write a Thesis*. URL: https://www.overleaf.com/learn/latex/How_to_Write_a_Thesis_in_LaTeX_(Part_1):_Basic_Structure.

[49] Overleaf. *Glossaries - Overleaf, Online LaTeX Editor*. URL: https://www.overleaf.com/learn/latex/Glossaries.

[50] Overleaf. *Bibliography management with biblatex - Overleaf, Online LaTeX Editor*. URL: https://www.overleaf.com/learn/latex/Bibliography_management_with_biblatex.

[51] OODesign. *Design Patterns — Object Oriented Design*. URL: https://www.oodesign.com/.

[52] Ahmed Shaaban. *Design Pattern With Kotlin. 1- singleton design pattern and its... — by ahmed shaaban — The Startup — Medium*. URL: https://medium.com/swlh/singleton-design-pattern-with-kotlin-2e6c8d42fc11.

[53] Tutorials Point. *NHibernate - Architecture*. URL: https://www.tutorialspoint.com/nhibernate/nhibernate_architecture.htm.

[54] André Mion. *Animated icons on Android. How to improve the user experience... — by André Mion — UpLabs Success Stories*. URL: https://stories.uplabs.com/animated-icons-on-android-ee635307bd6.

[55] Nick Rout. *Setting up a Material Components theme for Android — by Nick Rout — Over Engineering — Medium*. URL: https://medium.com/over-engineering/setting-up-a-material-components-theme-for-android-fbf7774da739.

[56] GeeksforGeeks. *Assets Folder in Android Studio - GeeksforGeeks*. URL: https://www.geeksforgeeks.org/assets-folder-in-android/.

[57] GeeksforGeeks. *Resource Raw Folder in Android Studio - GeeksforGeeks*. URL: https://www.geeksforgeeks.org/resource-raw-folder-in-android-studio/.

[58] Kotlin Codes. *Shared Preferences With Kotlin - Kotlin Codes*. URL: https://kotlincodes.com/kotlin/shared-preferences-with-kotlin/.

[59] Google. *Create dynamic lists with RecyclerView — Android Developers*. URL: https://developer.android.com/guide/topics/ui/layout/recyclerview#key-classes.

[60] Kory Dondzilla. *How to Create a SearchView with Suggestions in Kotlin*. URL: https://spin.atomicobject.com/2019/11/11/how-to-create-a-searchview-with-suggestions-in-kotlin/.

[61] EDMT Dev. *Kotlin Android Tutorial - Material Components V2 Chip and Chip Group - YouTube*. URL: https://www.youtube.com/watch?v=Q_iVwGz5eMQ.

[62] AristiDevs. *(5) CHIPS y CHIPGROUP - El componente que ya deberías estar usando [TUTORIAL ANDROID STUDIO] - YouTube*. URL: https://www.youtube.com/watch?v=AEmgxSTupoU.

[63] Tutorials Point. *How to pass a String from one Activity to another Activity in Android using Kotlin?* URL: https://www.tutorialspoint.com/how-to-pass-a-string-from-one-activity-to-another-activity-in-android-using-kotlin.

[64] Scryfall. *REST API Documentation · Scryfall Magic: The Gathering Search*. URL: https://scryfall.com/docs/api.

[65] HiManShoe. *Mongo DB in Kotlin*. URL: https://himanshoe.com/mongodb-in-ktor.

[66] MongoDB. *Quick Start - POJOs*. URL: http://mongodb.github.io/mongo-java-driver/3.7/driver/getting-started/quick-start-pojo/.

[67] Ray Rodrigo Guerrero. *Realm Database on Android: Getting Started — raywenderlich.com*. URL: https://www.raywenderlich.com/25768145-realm-database-on-android-getting-started.

[68] JetBrains. *Kotlin Multiplatform Mobile - Android Studio Plugin — Marketplace*. URL: https://plugins.jetbrains.com/plugin/14936-kotlin-multiplatform-mobile.

[69] MongoDB. *Install Realm - Kotlin Multiplatform SDK — MongoDB Realm*. URL: https://docs.mongodb.com/realm/sdk/kotlin-multiplatform/install/.

[70] realm. *GitHub - realm/realm-kotlin: Kotlin Multiplatform and Android SDK for the Realm Mobile Database: Build Better Apps Faster*. URL: https://github.com/realm/realm-kotlin.

[71] Salomon BRYS. *Announcing a painless Kotlin/Multiplatform NoSQL embedded database — by Salomon BRYS — Kodein Koders — Medium*. URL: https://medium.com/kodein-koders/announcing-a-painless-kotlin-multiplatform-nosql-embedded-database-30fed677549c.

[72] GeeksforGeeks. *Android SQLite Database in Kotlin - GeeksforGeeks*. URL: https://www.geeksforgeeks.org/android-sqlite-database-in-kotlin/.

[73] Tutorials Point. *How to use a simple SQLite database in Kotlin android?* URL: https://www.tutorialspoint.com/how-to-use-a-simple-sqlite-database-in-kotlin-android.

[74] JavaTPoint. *Kotlin Android SQLite Tutorial - javatpoint*. URL: https://www.javatpoint.com/kotlin-android-sqlite-tutorial.

[75] Charlie Custer. *SQL Commands: The Complete List (w/ Examples) – Dataquest*. URL: https://www.dataquest.io/blog/sql-commands/.

[76] Google. *SQLiteDatabase.CursorFactory — Android Developers*. URL: https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.CursorFactory.html.

[77] Foxxtrot. *A SQLiteOpenHelper Is Not A SQLiteTableHelper - Mad, Beautiful Ideas*. URL: https://blog.foxxtrot.net/2009/01/a-sqliteopenhelper-is-not-a-sqlitetablehelper.html.

[78] Google. *android.database.sqlite — Android Developers*. URL: https://developer.android.com/reference/android/database/sqlite/package-summary.

[79] StackOverflow. *Efficient paging in SQLite with millions of records - Stack Overflow*. URL: https://stackoverflow.com/questions/14468586/efficient-paging-in-sqlite-with-millions-of-records.

[80] Tom Harrison. *UUID or GUID as Primary Keys? Be Careful! — by Tom Harrison — Tom Harrison's Blog*. URL: https://tomharrisonjr.com/uuid-or-guid-as-primary-keys-be-careful-7b2aa3dcb439.

[81] Curso Kotlin. *Tutorial Retrofit 2 en Kotlin con Corrutinas - Consumiendo API [Capítulo 20 v2] - Curso Kotlin Para ANDROID*. URL: https://cursokotlin.com/tutorial-retrofit-2-en-kotlin-con-corrutinas-consumiendo-api-capitulo-20-v2/.

[82] OkHttp. *OkHttp*. URL: https://square.github.io/okhttp/.

[83] Camposha. *Kotlin Android DownloadManager Tutorial and Examples - Android Examples*. URL: https://camposha.info/android-examples/android-downloadmanager/#gsc.tab=0.

[84] Kory Dondzilla. *Download Files in Kotlin for Android Using Ktor and Intents*. URL: https://spin.atomicobject.com/2020/05/18/android-download-files-kotlin/.

[85] Elye. *Download File in Android with Kotlin — by Elye — Mobile App Development Publication — Medium*. URL: https://medium.com/mobile-app-development-publication/download-file-in-android-with-kotlin-874d50bccaa2.

[86] Picasso. *Picasso*. URL: https://square.github.io/picasso/.

[87] BezKoder. *Kotlin - Convert object to/from JSON string using Gson - BezKoder*. URL: https://www.bezkoder.com/kotlin-parse-json-gson/.

[88] BezKoder. *Leveraging the Gson Library — CodePath Android Cliffnotes*. URL: https://guides.codepath.com/android/leveraging-the-gson-library.

[89] Jesse Wilson. *Streaming - gson*. URL: https://sites.google.com/site/gson/streaming.

[90] Peter Vogel. *Converting JSON Objects to Relational Tables: Flattening the Object – Visual Studio Magazine*. URL: https://visualstudiomagazine.com/articles/2021/06/01/converting-json.aspx.

[91] StackOverflow. *How to add external fonts to android application - Stack Overflow*. URL: https://stackoverflow.com/questions/5634245/how-to-add-external-fonts-to-android-application.

[92] Google. *material-components-android/getting-started.md at master · material-components/material-components-android*. URL: https://github.com/material-components/material-components-android/blob/master/docs/getting-started.md.

[93] Solution Code Android. *(12) kotlin android studio How to create Material Design Dialog with Radio buttons (Single Selection) - YouTube*. URL: https://www.youtube.com/watch?v=mjIMaEv_VpA.

[94] Google. *VectorDrawable — Android Developers*. URL: http://android.cn-mirrors.com/reference/android/graphics/drawable/VectorDrawable.html.

[95] StackOverflow. *android - How to insert drawables in text - Stack Overflow*. URL: https://stackoverflow.com/questions/25521685/how-to-insert-drawables-in-text.

[96] StackOverflow. *android - SpannableString with Image example - Stack Overflow*. URL: https://stackoverflow.com/questions/3176033/spannablestring-with-image-example.

[97] Kotlin Foundation. *String - Kotlin Programming Language*. URL: https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/-string/.

[98] whichdigital. *whichdigital/ksv: A library for parsing csv-files written in and supporting features of Kotlin*. URL: https://github.com/whichdigital/ksv.

[99] RJ Smith. *Switching Between Fragments Without the Mindless Killing Spree — by RJ Smith — Sweet Bytes — Medium*. URL: https://medium.com/sweet-bytes/switching-between-fragments-without-the-mindless-killing-spree-9efee5f51924.

[100] StackOverflow. *How to move from one fragment to another fragment on click of an ImageView in Android? - Stack Overflow*. URL: https://stackoverflow.com/questions/23212162/how-to-move-from-one-fragment-to-another-fragment-on-click-of-an-imageview-in-an.

[101] StackOverflow. *android - how pass parameters between two fragment? - Stack Overflow*. URL: https://stackoverflow.com/questions/37024023/how-pass-parameters-between-two-fragment.

[102] StackOverflow. *android - findViewById in Fragment - Stack Overflow*. URL: https://stackoverflow.com/questions/6495898/findviewbyid-in-fragment.

[103] Google. *Los fragmentos y el componente de Navigation*. URL: https://developer.android.com/codelabs/basic-android-kotlin-training-fragments-navigation-component#10.

[104] Ankur Pratik. *Fragment Navigation Simplified: Android — by Ankur Pratik — Medium*. URL: https://medium.com/@ankurpratik/android-fragment-navigation-3e2c2c42081d.

[105] rajshekh. *How to make Flip animation image in android?* URL: https://findnerd.com/list/view/How-to-make-Flip-animation-image-in-android/22811/.

[106] StackExchange. *adobe illustrator - Inkscape: How to partly remove a stroke with a different object? - Graphic Design Stack Exchange.* URL: https://graphicdesign.stackexchange.com/questions/101958/inkscape-how-to-partly-remove-a-stroke-with-a-different-object.

[107] StackOverflow. *android - Material Component TabLayout inside Fragment in Kotlin - Stack Overflow.* URL: https://stackoverflow.com/questions/51014462/material-component-tablayout-inside-fragment-in-kotlin.

[108] StackOverflow. *android - Fragment Tabs inside Fragment - Stack Overflow.* URL: https://stackoverflow.com/questions/41413150/fragment-tabs-inside-fragment.

[109] StackOverflow. *android - Recycler View Horizontal Scroll with 2 columns - Stack Overflow.* URL: https://stackoverflow.com/questions/47300862/recycler-view-horizontal-scroll-with-2-columns.

[110] StackOverflow. *gridview - Android Recyclerview GridLayoutManager column spacing - Stack Overflow.* URL: https://stackoverflow.com/questions/28531996/android-recyclerview-gridlayoutmanager-column-spacing.

[111] StackOverflow. *java - How to use TabLayout with ViewPager2 in Android - Stack Overflow.* URL: https://stackoverflow.com/questions/55372259/how-to-use-tablayout-with-viewpager2-in-android.

[112] StackOverflow. *How to access a TextView outside RecyclerView in Android? - Stack Overflow.* URL: https://stackoverflow.com/questions/63807972/how-to-access-a-textview-outside-recyclerview-in-android.

[113] StackOverflow. *java - Kotlin: Interface ... does not have constructors - Stack Overflow.* URL: https://stackoverflow.com/questions/43737785/kotlin-interface-does-not-have-constructors.

[114] StackOverflow. *android - Detect when RecyclerView reaches the bottom most position while scrolling - Stack Overflow.* URL: https://stackoverflow.com/questions/36127734/detect-when-recyclerview-reaches-the-bottom-most-position-while-scrolling.