Midterm Software Proposal

“Collection Manager”

A tool for tracking and managing various collectibles.

Authors:
Juan Valdivia
Guy Moses
Micheal Olawale
Nicholas Novelle

CM2020 Agile Software Development
Module Leader: Dr Sean McGrath

Table of Contents

Introduction

Our goal with this report is to present our findings on the feasibility and usability of a collection management tool for physical and digital collectibles. We will investigate the need for such a tool, what the Minimum Viable Product (MVP) feature set is, and the potential for such a tool to become a key player in the fintech space.

We begin by stating the problem we are trying to solve, and then providing our concept for an application that fits the niche described. We continue by exploring this application, suggesting initial implementations, and evaluating these implementations to deduce what the best feature set for an initial release of the platform is.

Statement of Problem

Collectibles have often been considered a high-risk investment. Still, with changes in the financial landscape and evolving technologies, they have become more attractive for those investors looking to gain an edge over traditional financial instruments. In particular, we saw during the Covid Pandemic of 2019-2022 a huge increase in buyouts on physical collectibles and the emergence of Non-fungible Tokens (NFT) for digital collectibles (Raj). This created a renewed interest in a traditionally high-risk field.

While there have been hyper-focused tools for tracking collections and their values, no general tool exists that can show a complete picture of what a person has, and what those things are worth, across a diverse set of collectible items. For example, you can find trackers for a single card game, or perhaps even multiple card games, but not a unified application that can show you the value of both your trading cards and NFTs.

Because of this, we feel there is a new niche in a tool or platform that makes it simple to know what you have, and how much it’s worth, and convert that to actual money through trading and selling.

Concept

Our project is an application and platform for tracking the value of a user's collection over time. Initially, we would present the user with a simple interface to add items to the collection, and would automatically acquire pricing information from various sources. Additionally, we would offer a platform that allows the user to convert their collectible into money by presenting trade and buy offers on any object the user marks as available for sale or trade. The ability to carry out trades on our platform also creates a unique market where a diverse set of collectibles can be exchanged for each other, providing more interesting investment opportunities than traditional platforms which normally only allow the trading of a single type of item.

We differentiate ourselves from existing products by providing a unified platform for investing, tracking, and trading across all kinds of collectibles. By making the platform extensible, we allow for unique functionality for specific collectibles to exist alongside the general features and guarantee that our product will keep up with the industry as it continues to evolve.

The core of the application is a web service that allows users to log their collectibles. Each type of collectible can define its schema and present the user with the option to add additional fields to aid in cataloguing their growing collection. Tied to this, we have background scripts that gather pricing information from as many sources as possible and cache this data locally. This pricing information is then used to present the going rates for various items on distinct services. Once we launch our internal marketplace, we can include the internal prices along with that of our competitors, providing our users with the ability to always pick the option that is best for them.

To ensure our users have a world class experience, we can also provide many quality of life features–such as posting sales on external websites for our users and automatic pricing–as well as internal policies which assist our users in making money. Key among these is advanced data analysis tools which show how a given item is performing, and pull in key information which may affect the price of that item in the future. For example, a user who is investing in a Trading Card Game could see the latest news and deck lists for cards in their collection and use this to help gauge if a given card is over- or undervalued.

Aims and Objectives

The primary goal is to create a platform to capitalise on the growing interest in the collectibles market as a form of investment, and provide our users with best-in-class tools and support to aid in the continued growth of their investment.

Key Objectives:

1. Provide a unified platform that can serve as the basis for any collectible investment
2. Provide users with access to data gathered on their collectibles to help them gauge their value
3. Make it simple to trade and sell collectibles
4. Use data analysis to understand the market and then capitalise on opportunities as they present themselves
5. Make the platform extensible in order to grow with the industry

Stakeholders

The primary stakeholders are the investors using our system. This includes both users who are interested in the investment aspect of collectibles, as well as those who are interested in managing and growing their collections.

A potential secondary stakeholder are partners who are interested in leveraging our data set to provide goods and services that may interest our users. These will have to be carefully selected to ensure a minimum quality, but can enrich our offering by providing products we may not have the time or inclination to create ourselves.

Stakeholder requirements gathering

We are making a collectibles tracking app and these are the following questions we needed to answer to gather stakeholder data correctly.

1. What information do we need from users?

We need to collect information about if the collectibles application would be useful to the Stakeholders and how they would mainly want to use it.

We need to ask them if they collect something and if the application would be useful to them. Then we need to understand what they collect and how much of it they have. With that in mind we can find out if they keep track of their collectibles in some way already. And then we can ask them if they think it would be helpful if they had a good collectibles tracking system. With some thoughts that they had at the end.

The way we achieved this is by creating a google forms form and putting the following questions in the google forms, after that we sent it out and got our results.

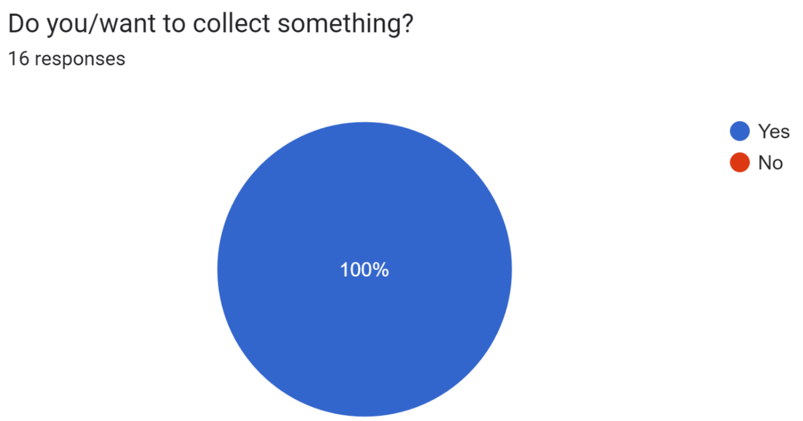
2. Are there any features we might not have thought about?

We have noticed that most of the there were a lot of video game collections out there as well as some unexpected items like chicken and hotel key cards, that we didn’t account for when starting planning.

As a result we adjusted from more specific collections to more general collections.

3. What does the User require?

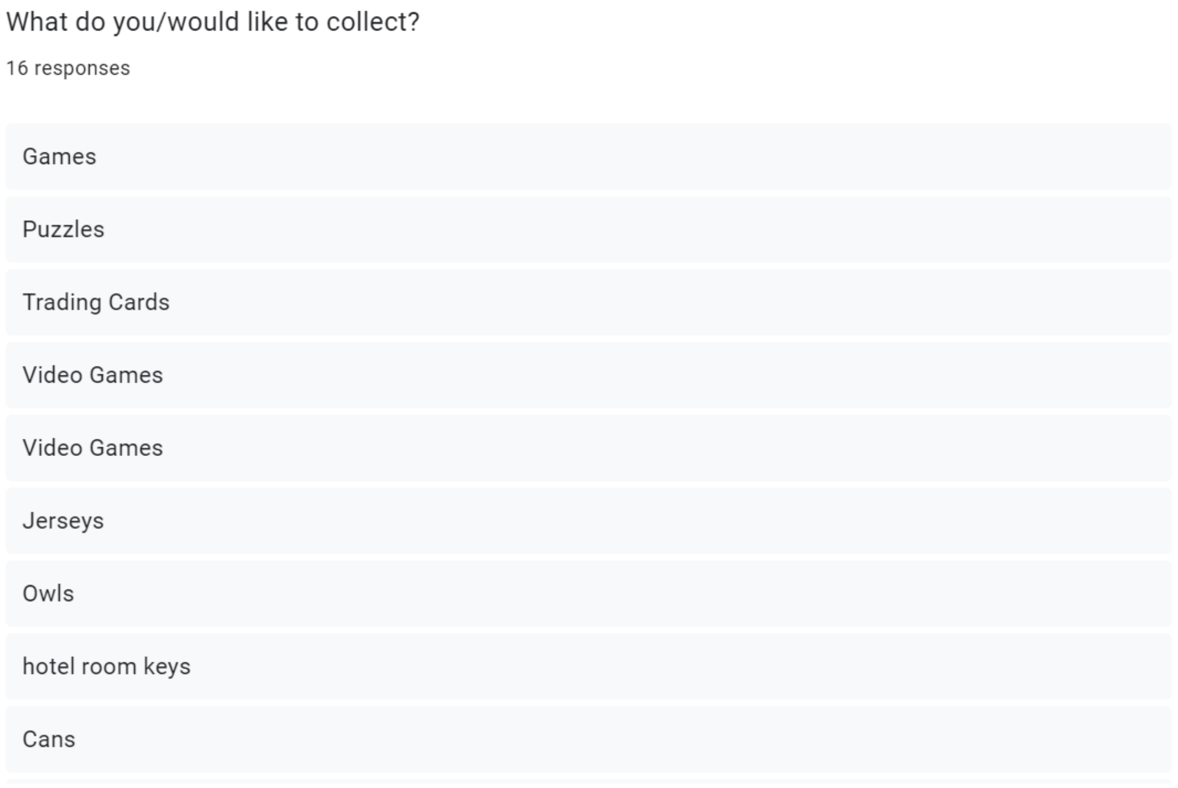
After going over the analysis we found out that everyone who took our survey was collecting something as well as the the other following information:



Do you/want to collect something?

16 responses

Here are some answers about what people like to collect. This is where we decided That it is best to have a broader collection system without a focus on anything specific:



What do you/would like to collect?

16 responses

Games

Puzzles

Trading Cards

Video Games

Video Games

Jerseys

Owls

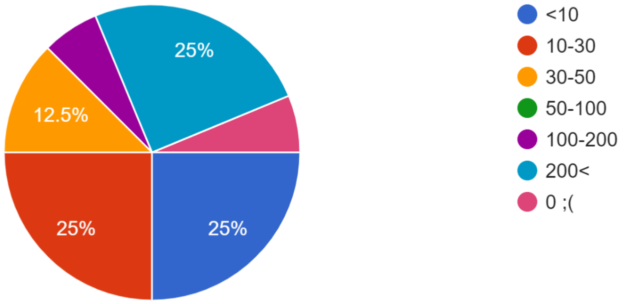
hotel room keys

Cans

Then we tried to figure out the scale of all of these collections and understand if there will be a need to accommodate bigger more complex collections often:

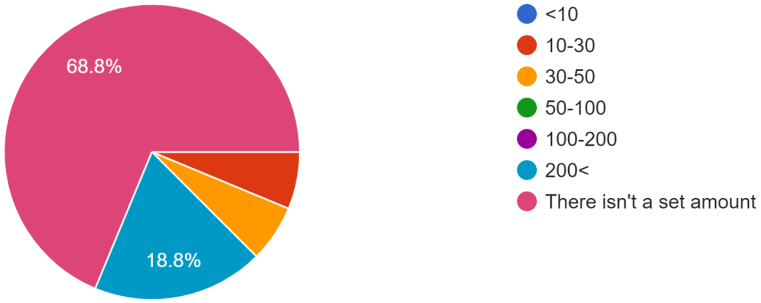
How many items do you currently have in your collection?

16 responses



How many items would you need to complete your collection?

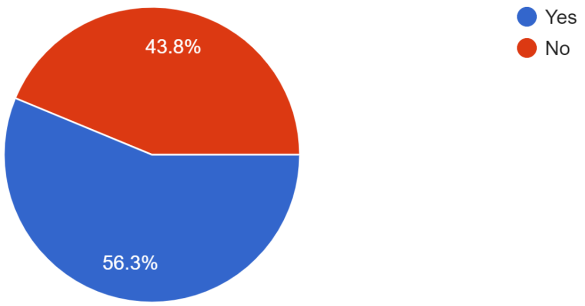
16 responses



Then we needed to figure out if the users already used a system to keep track of collectibles or not. To see if existing systems are already used and if they are well adopted. We also asked if they thought a system like that would be helpful and we have noticed that more people would want a system like that then the amount of people that already used existing systems.

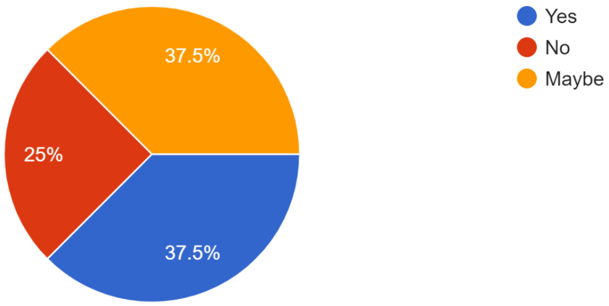
Do you keep track of your collectibles or any information about them?

16 responses

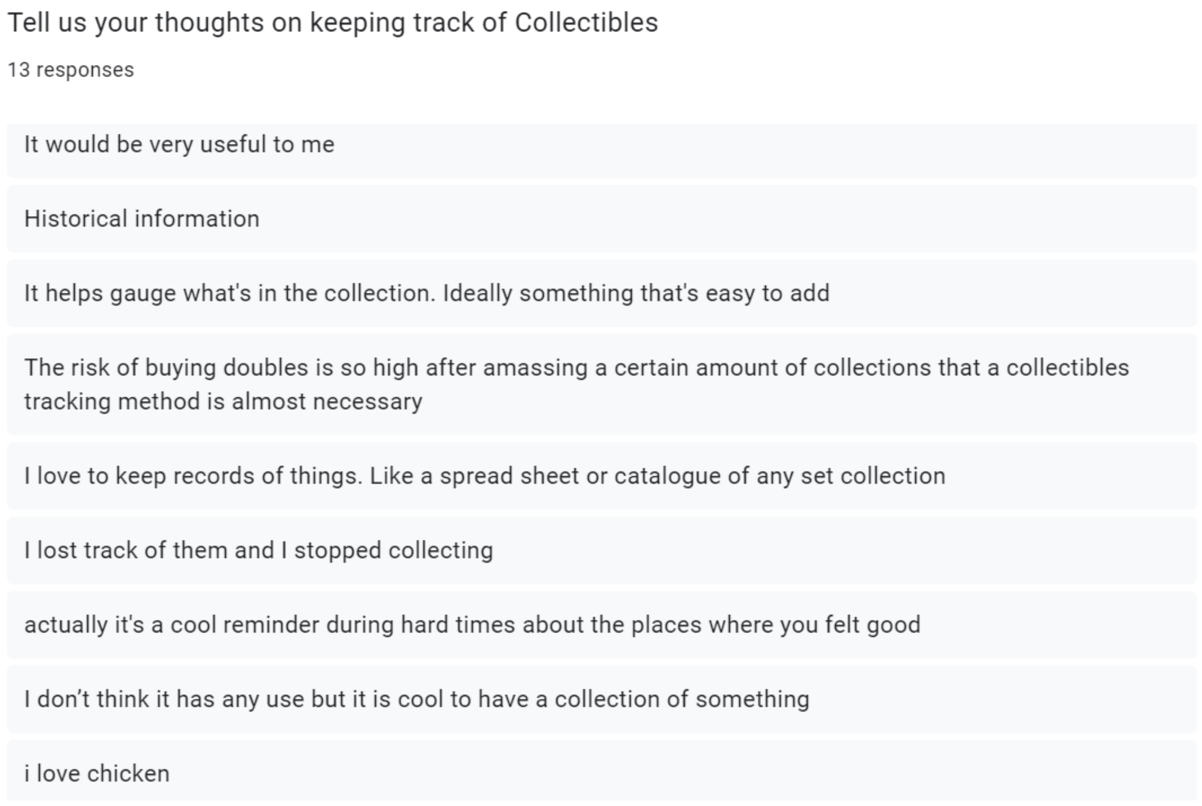


Do you think it would be helpful to keep track of those collectibles or the information about them?

16 responses



We have also asked users what their thoughts are on keeping track of collectibles. And we have noticed from a few responses that ease of use would be very important in a collectibles system. That means that it should be as simple as possible to add a new collection or a new collectable.



Tell us your thoughts on keeping track of Collectibles

13 responses

It would be very useful to me

Historical information

It helps gauge what's in the collection. Ideally something that's easy to add

The risk of buying doubles is so high after amassing a certain amount of collections that a collectibles tracking method is almost necessary

I love to keep records of things. Like a spread sheet or catalogue of any set collection

I lost track of them and I stopped collecting

actually it's a cool reminder during hard times about the places where you felt good

I don't think it has any use but it is cool to have a collection of something

i love chicken

As a conclusion we will do another requirement gathering after we make a prototype that users can test and experiment with for the second half of this project.

Scope

The initial scope for our product must be limited to ensure we can get it into the user's hands as quickly as possible. This will help us evaluate our assumptions in a real-world setting and make adjustments as needed. As such, our focus is on making a simple application with a solid base to grow from.

1. Standard CRUD operations for a database designed to store collectible information
    a. The schema must account for the diversity of the collectibles market. It does so by defining a ‘core’ table, that all collectibles share, and ‘extension’ tables, which stores information unique to a collectible type.
    b. An additional table for user tracked fields can be used not only to allow users to track arbitrary data points, but also give us insight into what users consider important.
2. An application to view and interact with the database
    a. Initially, a web application gives us the widest reach by being available on both mobile and desktop platforms.
    b. The application should allow a user to view and manage their collection. It is the entry point to the database and the CRUD operations.
3. Web scraping scripts to gather price information
    a. Run exclusively on back end servers, the scripts will visit common selling sites–like Ebay–to find price information for our users.
    b. The set of sites that we visit can be adjusted for each type of collectible. For example, when working with Magic the Gathering cards, we can visit sites like ScryFall.

Out of Scope

1. Market Place
    a. We eventually do want to provide marketplace services, however the initial launch of the platform does not include a marketplace in order to facilitate a smooth release.
    b. Further research must be done to determine how the marketplace will be presented, as well as which integrations–such as payment processors–we will use to service our users.
2. Unique Collection Features
    a. We also want to focus on a set of broadly applicable features, so highly unique features, eg. a deck builder for a card game, are not part of the initial implementation.

SWOT Analysis

Strengths

The primary strength is the unified and extensible nature of the platform. By providing a one-stop-shop solution, we can entice users to focus on our platform rather than having to use many different applications and services. The unified interface makes it easier to track new collections and provides a standard metaphor for interacting with collections digitally.

The modular design in particular will allow us to adapt with new collectible markets as they appear, thereby keeping us relevant in the space.

Weaknesses

The choice to broadly support many different types of collectibles also means it will be more difficult to add in features that specific collectible communities may find desirable. For example, trading card game players would like tools like deck builders and simulators which would be difficult to implement in the framework of a generic collection service. While not impossible, each hyper-focused feature will have to be evaluated on a case-by-case basis to determine if it is worth expending resources to build.

Opportunities

The Antiques & Collectibles market is estimated to have a size of $2.8bn in 2023 (IBIS World). As shown in the market research, there are not many companies that seek to service this market and make it easy for new investors to get involved. Additionally, the market is fragmented into sub-markets based on the type collectible. By providing a unified platform, we can entice many different types of collectible investors and make it simple to diversify into new sub-markets.

Threats

Initially, the largest threat is the lack of an internal marketplace. We will depend on external sources for price data, along with buying and selling capabilities, and as such are at the mercy of our competitors. Once the internal market place is mature, we no longer require external sources, though we can still leverage them. In the meantime, however, we must tread lightly to ensure we don’t run afoul of any service agreement.

Market Research

The existing market is largely focused on Web and Mobile apps that are dedicated to tracking one specific type of collection. In general, each focuses on a limited set of collectible items. We can also see that the more generic a collection tool is, the fewer unique features it offers. The most generic option offering only the ability to add items to a database.

I. Web

A. Collection Tracker – Allows Tracking of Trading Cards, Sports Cards, Video Games, and Comics. Each broad category includes subcategories for specific publishers, games, sports, etc. Focuses primarily on tracking the value of your collection by scrapping Ebay for sold prices on items. The website does provide limited API access.

B. TCGPlayer – A marketplace for trading cards, TCGPlayer also includes a collection manager for users. The manager focuses only on trading cards for the games that TCGPlayer sells. Like most collection managers, it focuses primarily on the value of the collection. Price information is retrieved directly from the vendors that sell on the website. The website also has a mobile app which replicates the functionality of the website.

C. Comics Price Guide – A very similar website to TCGPlayer, but focused on comics instead of Trading Cards. It uses a way to connect with each other and sell their comics back issues, as well as tracking their collection. It also appears to be a primary source for prince information.

D. Action Figures 411 – A website to track Action Figure collections, it functions similar to classic collecting Magazines. It provides users with price guides and visual references that they can use when trading in action figures.

E. Collectorz – Web and Mobile app that allows users to track theri Games, Movie, Book, Comics, and Music collection. The application also integrates with IMDB in order to pull in metadata for Movies. The mobile app version includes a barcode scanner to make adding items to a collection simpler.

II. Mobile

A. Dragon Shield TCG Scanner Apps – Dragon Shield provides a set of apps designed for different Trading Card Games. Each app provides a collection manager, which is populated by either manually entering a card or scanning it. It also allows users to separate their collection into subcategories and provides pricing information for the subcategories and the collection as a whole. Additionally, it provides a Deck Management tool for each of the games.

B. Colka – General purpose collection manager that focuses on completing collections. Instead of historical price information, allows the user to buy items off of eBay or list an item directly.

C. My Game Collection – An app for tracking Video Games and Console collections. Allows the user to import from a variety of other game-collecting databases and sources. Allows the user to rate their games.

D. iCollect Everything – A general-purpose cataloguing app for mobile. Users can scan the barcode of the item they collect in order to add and track it.

III. Desktop

A. Callibre – Ebook Management software for desktops. Allows the user to manage their digital book collection by providing tools to tag, update, remove, and manage their digital books.

B. Ant Movie Catalog – Desktop software that manages a user's movie collection. Once a movie has been added to the database, the user can add metadata or pull it in from IMDB. The application also provides statistics on the movie collection and the ability to generate an HTML or Print report of the collection.

C. Sports Card Collection Manager – Does what it says on the tin. Implements only the required feature set to be called a collection manager.

Feature Comparison

All competitors provide the option to perform standard CRUD operations on their collection database. The attributes of each table are tailored to the type of item being collected. Additionally, all have the option of tagging items with arbitrary values in order to easily find them. These features are considered core and part of the MVP so are not included in the table below. Instead, the table includes features which distinguish each app from the others.

App	Online Database	Current Prices	Price Tracking	Scan to Add	Buy and Sell	Collection Specific Features	Data Export
Collection Tracker	X	X	X	X	-	-	-
TCGPlayer	X	X	-	X	X	X	-
Comics Price Guide	X	X	-	-	X	X	-
Action Figure 411	X	X	-	-	-	X	-
Collectroz	X	-	-	X	-	X	-
Dragon Shield	X	X	X	X	-	X	-
Coleka	-	-	-	X	X	-	-
My Game Collection	X	-	-	-	-	X	-
iCollect Everything	-	-	-	X	-	X	-
Callibre	-	-	-	-	-	X	X
Ant Movie Catalog	-	-	-	-	-	X	X
Sports Card Collection Manager	-	-	-	-	-	-	-

Work Style

The work style we will be using follows the agile software development lifecycle process instead of the conventional approach, essentially because the agile approach is iterative and suited for low- to medium-sized projects such as the collection manager project. It is also collaborative; team input is sought at the end of each iteration, unlike the traditional approach that evaluates the project at the end of the development process.

The foundation of agile software development is the concept of incremental and iterative development, in which the steps that comprise a development life cycle are revisited on multiple occasions. It does this by iteratively improving software and using customer feedback to converge on solutions (Leau et al. 162). The common agile development approaches are crystal
methodologies, dynamic software development methods, feature-driven development, lean software development, Scrum, and extreme programming (Highsmith and Cockburn 120)

No single "best" agile methodology can be applied to any and all kinds of projects and businesses. The precise requirements and limitations of the project, as well as the capabilities and preferences of the team and the stakeholders, all play a role in determining which strategy is most appropriate.

Certain agile methodologies are used more frequently and are a better fit for specific categories of projects than others. Take, for instance:

The framework known as Scrum is used for managing and completing complicated projects. It works exceptionally effectively for projects that have requirements that are extremely emergent or are changing at a quick rate.

Extreme programming, sometimes known as XP, is an approach to software development that emphasises teamwork while maintaining a strict work ethic. It is a set of concepts and practices. It works especially well for initiatives with strict deadlines and a primary focus on delivering results as quickly as possible.

In developing software, "lean development" is a methodology that emphasises maximising value while decreasing waste. Projects that primarily emphasises efficiency and consistent improvement are an ideal fit for this methodology.

Crystal is a family of agile methodologies that may be adapted to meet the particular requirements and restrictions of the team and the project at hand. A focus on people and communication distinguishes them, and they are especially suited to projects that must satisfy complicated social and technical criteria.

The nature of the project, the expertise and talents of the team, the culture of the company, as well as the demands and expectations of the stakeholders are all elements that will ultimately play a role in determining which strategy will be the most effective for a given project.

The collection manager project will involve the creation of a platform that can support a wide range of collectibles and allow users to create their modules or modify existing ones. In addition, the project will involve a significant amount of uncertainty because it will involve the potential for users to create their modules. In this scenario, an agile methodology well-suited to evolving requirements and quick iteration, like Scrum or Extreme Programming (XP), could be a good fit.

Under the Scrum methodology, the overall project would be divided into manageable chunks of labour referred to as "sprints," The team would work together to produce functional software after each sprint. The team agreed to get together regularly to discuss the progress that had been made and make any necessary adjustments to the plan. This strategy would allow for rapidly adapting to changing requirements and responding flexibly to new issues.

Extreme Programming, also known as XP, is another agile methodology that has the potential to be useful for completing this project. It emphasises the prompt delivery of working software, close collaboration between developers and clients, and a focus on simplicity and ongoing development. The team would approach their work in a highly iterative and incremental fashion, with the primary goal being the consistent delivery of small, workable software increments as they were developed.

Our work style will require writing easily maintained and thoroughly documented code. Because of this, it will be much simpler for each team member to comprehend the code and modify it as the project progresses.

By adhering to sound software design principles, we will be able to build a codebase adaptable to shifting requirements and new features since it will be flexible and easy to extend.

We will also perform exhaustive testing on the codebase: there will be a much easier time locating and fixing errors early on in the development process as a result of this, as well as ensuring that the code is of high quality overall.

Similarly, the codebase will be tracked using GitHub and git version control at the end of each iteration. By utilising version control, we will be able to work more effectively with other team members and track changes made to the codebase.

We shall have protected branches (such as the main branch) and only allow merges through pull requests. In this procedure, a distinct branch is created for each set of changes, and then a pull request (PR) is created to request that the changes be merged into the primary development branch. After that, a member or members of the team can have a look at the pull request before it is merged.

We will also configure automated testing using helpful GitHub apps such as

Access Lint: This includes automated website accessibility testing. Accessing will examine the changes and comments when a pull request is opened for any new accessibility issues. It will then provide you with prompt, timely, and targeted feedback before the code is made public.

GitGuardian: It offers the highest possible level of protection for software engineers.

It can uncover any hard-coded secrets within commits and repositories and assist with rectification and avoidance.

Codecov: This provides a hosted service that gives metrics about the percentage of the source code validated by the automated tests that are a component of the software build process.

Furthermore, the work style will incorporate test-driven development.

Test-driven development, sometimes known as TDD, is a process for developing software in which the tests for a piece of code are created before the code itself is ever developed (Beck, 2002). This strategy can be helpful in several ways, including helping to enhance the code's design and maintainability, ensuring that the code is correct and meets the criteria, and ensuring that it does fulfil the requirements.

We have many different TDD methodologies to choose from if you want to use them for the collection management project, including the following:

The initial technique for test-driven development (TDD) was called "classic TDD," It involved a developer writing a test that failed, implementing the code to make the test pass, and then refactoring the code to enhance its design. This approach is repeated for each piece of functionality being put into effect (Beck).

The term "behaviour-driven development" (abbreviated as "BDD") refers to a subset of test-driven development (TDD) that emphasises defining the behaviour of a system from the point of view of the end user (Smart). Writing tests in a common language and using a simple syntax for non-technical stakeholders to comprehend is an essential part of behaviour-driven development (BDD).

Acceptance Test-Driven Development, or Acceptance TDD (ATDD), is a subtype of Test-Driven Development (TDD) that emphasises the drafting of tests that reflect the acceptance criteria for a user story or feature (Osherove). Writing tests that define the system's intended behaviour is one of the steps involved in ATDD. The next step is to write the code to ensure the tests are successful.

Lastly, to achieve an agile work style, we will also ensure that there are:

1. Engagement of early-stage users
2. Development that takes place in cycles or iterations
3. Teams that are responsible for their organisation
4. The ability to adapt when confronted with change

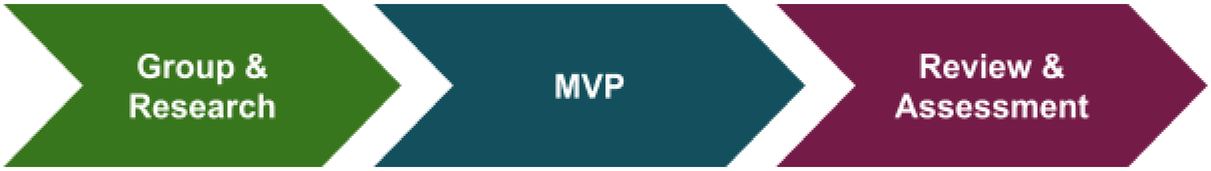
Timescale

This project involves multiple steps, with various complexity - where each step could be done by 1 or more persons of the team.
This means we need to keep track of the project schedule to ensure we can deliver a product within the time scheduled for the project.

High-level

The work on the project is divided into 3 big parts:

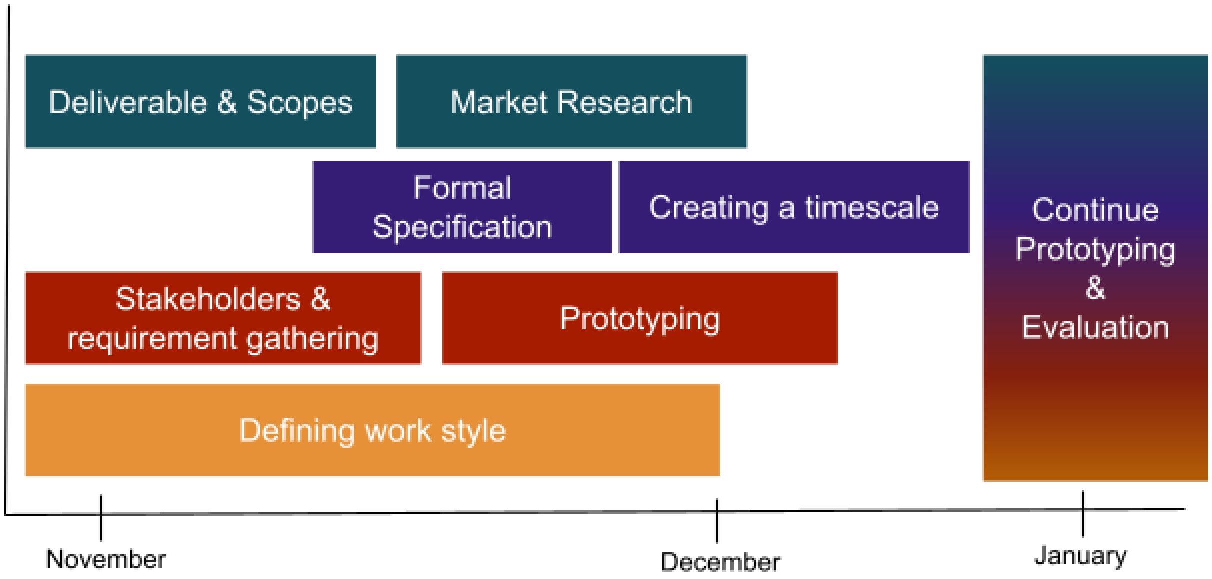
1. Forming the group and starting research
2. Building the MVP product
3. Reviewing, summarising, and assessment



For longer projects, where the participants invest more time, this process (made of 3 parts) will be done again and again (Research -> development -> review).

In more depth

The work in the research phase (first phase) will be separated this way among the team members this way:



Group members by colour:

Juan Valdivia

Guy Moses

Nicholas Novelle

Olawale Michael Juwon

Feature Set

I. Features – Wish List

- A. Basic CRUD operations
    - 1. Arbitrary Fields
    - 2. Arbitrary Tags
    - 3. sold prices
- B. Collection viewing options
    - 1. See your own collection breakdown
        - a) By categories
        - b) By price
        - c) By date of purchase
    - 2. Compare your collection to different user’s collections
- C. Filter and Searching
- D. Scan To Add
- E. Online Backups
    - 1. Integrate with Google Drive, and Dropbox, for user-owned backups
    - 2. Host our own server so we can analyse the data
- F. Buying and Selling
    - 1. Start by supporting external sources
    - 2. Build internal marketplace
    - 3. Is this where we can monetize the application
- G. Image Storing (Insurance purposes)
- H. Prices
    - 1. Current Price Information
        - a) Generic: ebay + specific: ScryFall, sources
        - b) When possible, add regional pricing for existing markets
    - 2. Historic Price Data
        - a) Data cached on our servers
        - b) Reports showing current and past value
- I. Data Export/Import
    - 1. Allow users to switch to a different application if desired
    - 2. Aid in the creation of a standard to make it easy to change tools

Colour map:
High priority
Medium priority
Low priority
Lower priority

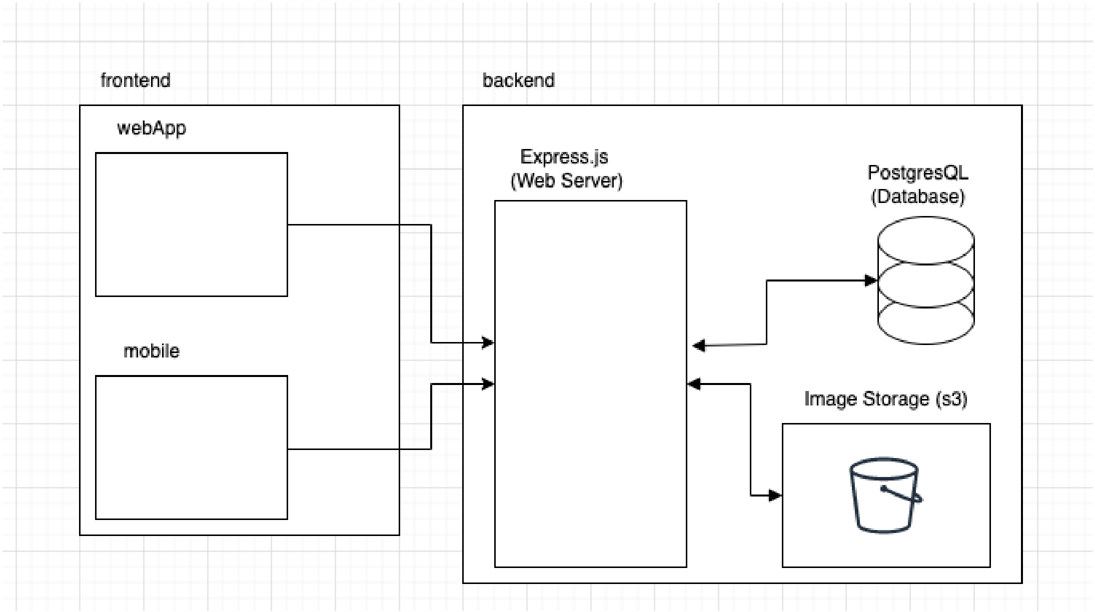
Formal Spec

According to our execution plan (timescale) we will divide the work into several phases.
Each phase will end with a product with a defined feature set - we can deliver.
In the first phase, we will deliver the 3 features from the following table:

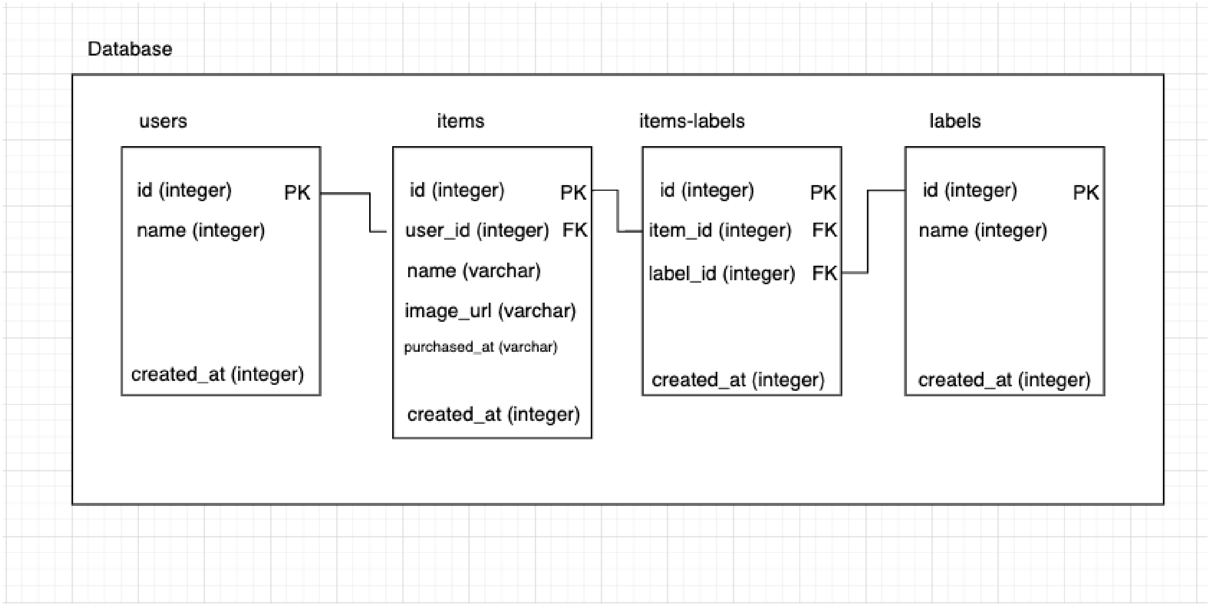
Feature set (First phase)

feature	Priority	must-have / nice-to-have
Managing a collection/ wallet ● Insert new items ● View items ● update/ delete items	1	must-have
Advanced collection viewing features ● Sort by category ● Sort by date of purchase	2	must-have
Filters and Querying features ● Search items by name ● Search items by labels	3	must-have

High-level Architecture
From: (diagrams.net)



Database UML (first phase):



App Backend API Endpoints

/user

GET:id (user)
POST (new user)
PATCH (existing user)

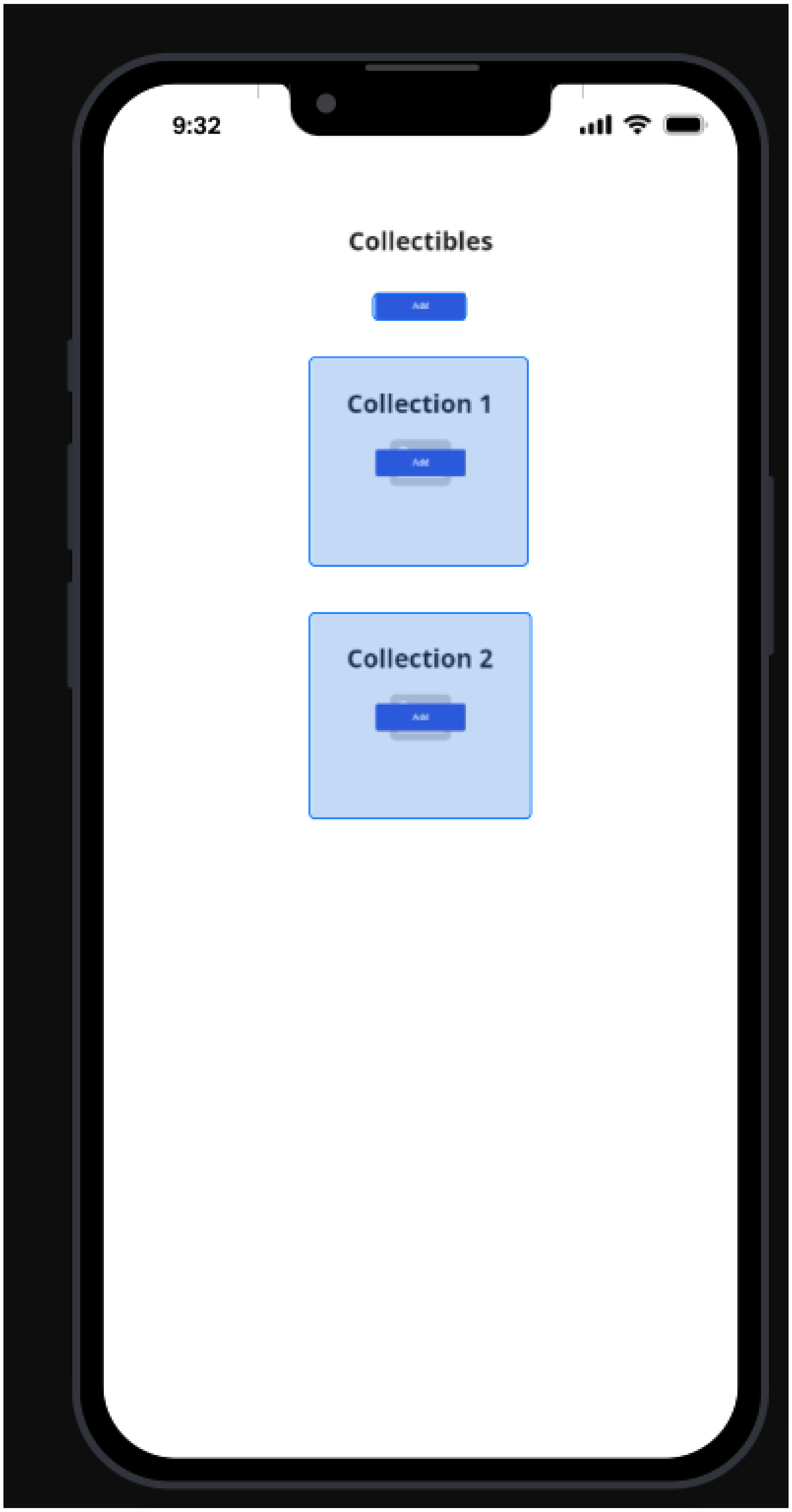
/item

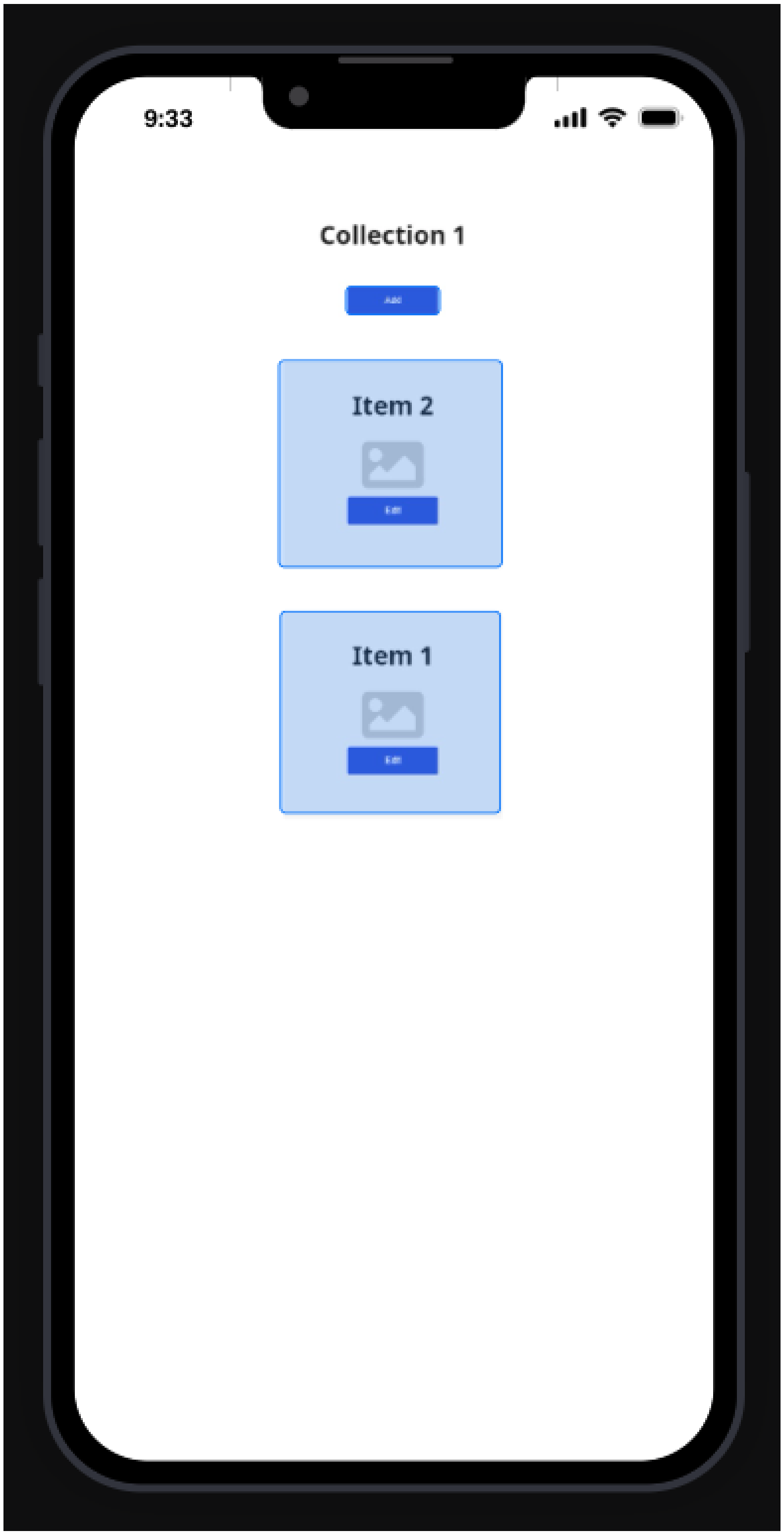
GET (list of items)
GET:id (item)
POST (new item)
PATCH (existing item)
DELETE (item)

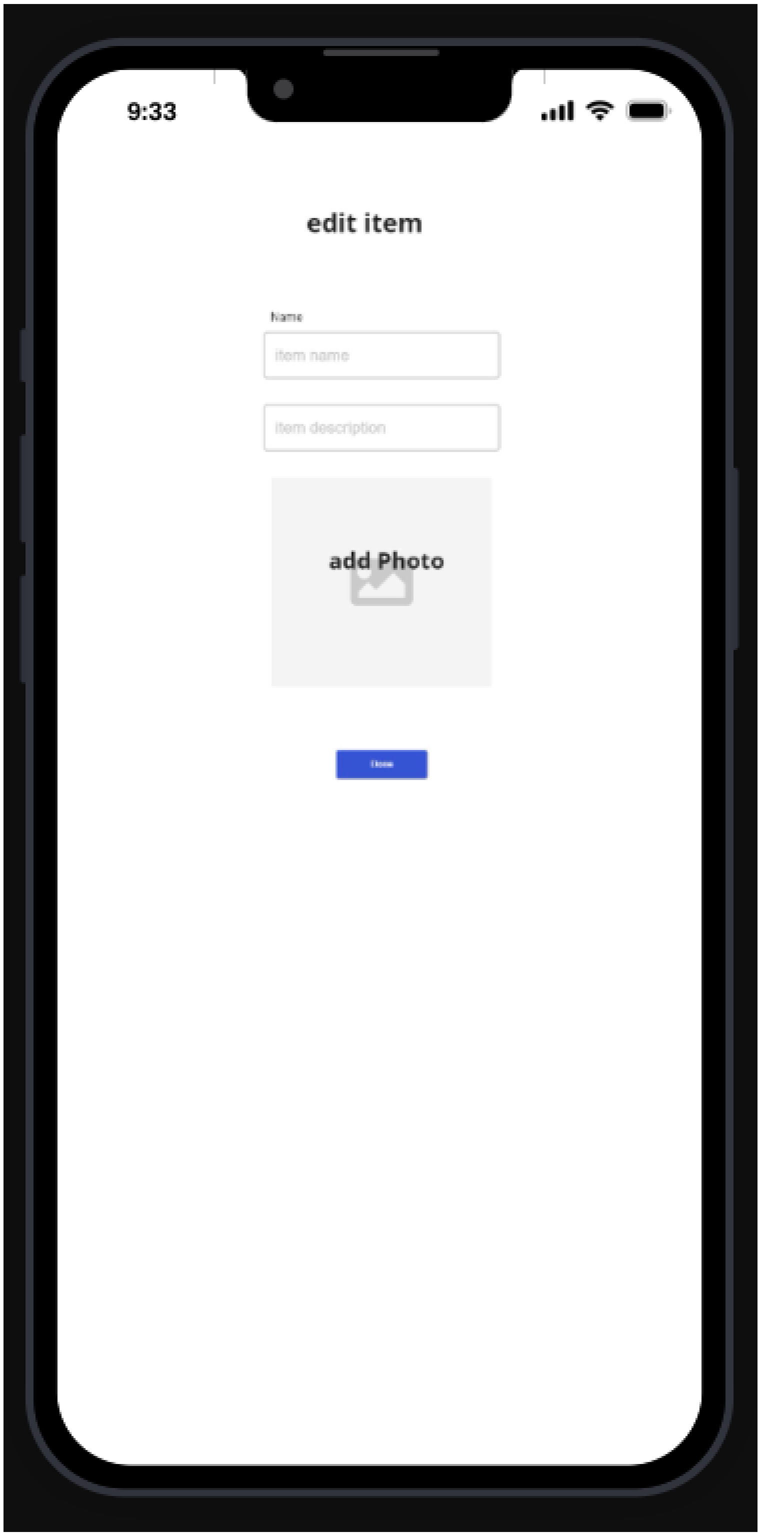
/label

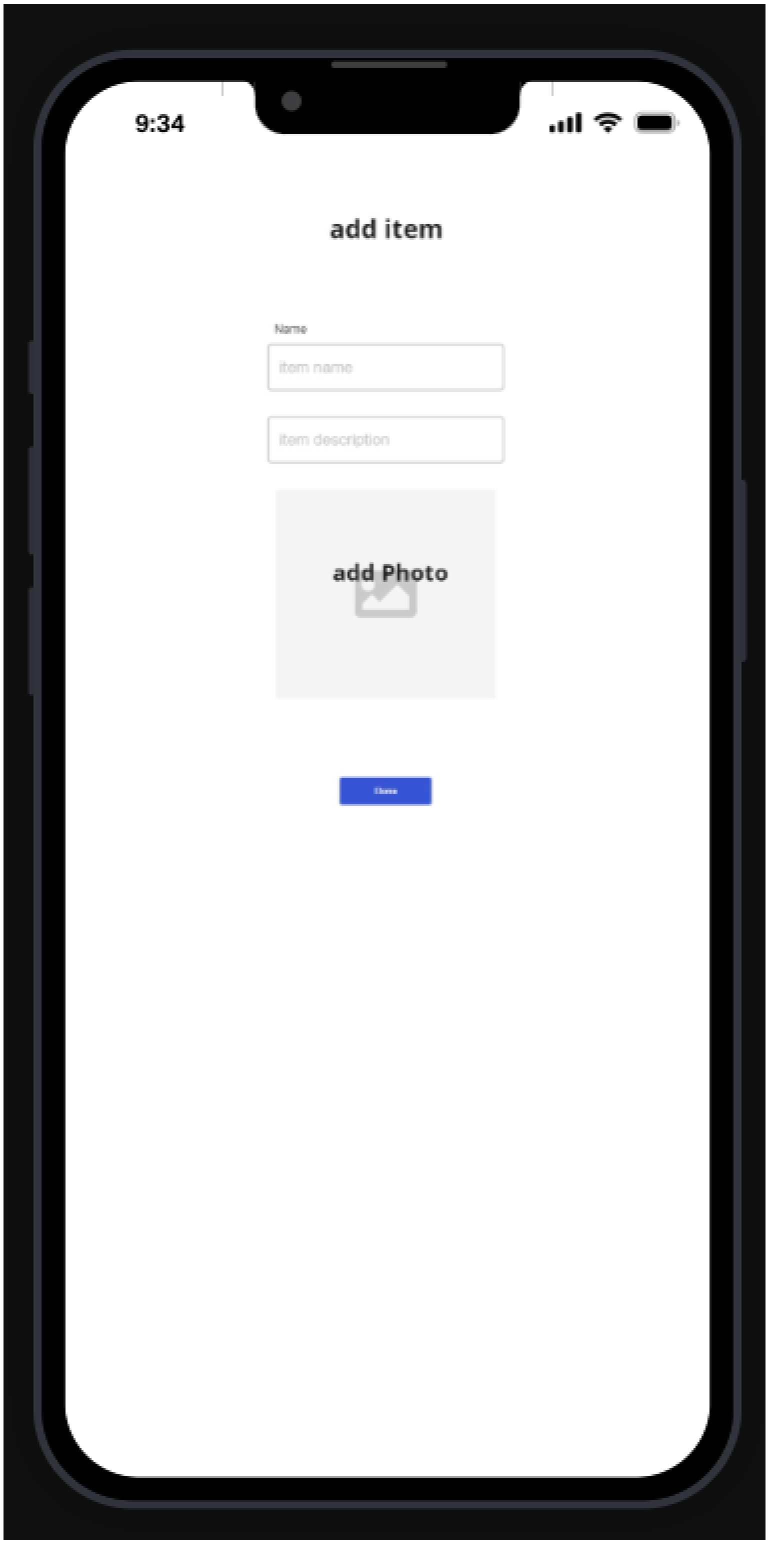
GET (list of labels)
POST (new label)

Design




9:33
Collection 1
Add
Item 2
Edit
Item 1
Edit


9:33
edit item
Name
item name
item description
add Photo
Done



from: https://nikinov359868.invisionapp.com/console/share/MV7HKTRPCQD/979052422

Conclusion

Our proposal identifies a growing investment area that is currently underserved by existing technology. The approach we are taking we feel will fill a niche that will continue to grow as younger investors feel more comfortable with the market. By identifying a minimum feature set, we can focus on getting a completed product out as soon as possible, allowing us to quickly grow with our user base and reevaluate our assumptions. Research into the market also shows that it is ripe for innovation. With such a large cap, we can expect that even a small success at the start will bear fruit very quickly.

The outcomes of this initial planning stage have given us a strong base to build on. Unfortunately, we don't have a lot of useability feedback from our initial designs–mostly having updated them ourselves as we created them, however, they support the features both we and our stakeholders identified as important. The early phases of coding will be focused on getting an alpha version of the site into the hands of users as quickly as possible.

The main concerns going forward are the difficulty of creating our marketplace. Many technologies can help us create the service, but the most important challenge to overcome will be getting people to use it. After all, a market without vendors and shoppers is just a bunch of empty stalls. It is essential that our pre-market offering instil a sense of confidence and appreciation in our users, making it likely that they will want to use us as their primary source for buying, trading, and selling their collectibles.

Works Cited

Beck, Kent. *Test Driven Development*. Addison-Wesley, 2003. Accessed 7 January 2023.

Highsmith, J., and A. Cockburn. “Agile software development: the business of innovation.” *IEEE Xplore*, vol. 34, no. 9, 2001, pp. 120-127, https://ieeexplore.ieee.org/abstract/document/947100.

IBIS World. “Online Antiques & Collectibles Sales in the US - Market Size.” *IBISWorld*, 20 October 2022, https://www.ibisworld.com/industry-statistics/market-size/online-antiques-collectibles-sales-united-states/. Accessed 7 January 2023.

Leau, Yu Beng, et al. “Software Development Life Cycle AGILE vs Traditional Approaches.” *IACSIT Press, Singapore*, vol. 37, no. 37, 2012, pp. 162-167, http://ku-fpg.github.io/files/agile-traditional.pdf.

Osherove, Roy. *The Art of Unit Testing: With Examples in C#*. Manning, 2013. Accessed 7 January 2023.

Raj, Rahul. “The Impact of Covid and Climate Change on the Collectibles Market.” *Rolling Stone*, 30 September 2022, https://www.rollingstone.com/culture-council/articles/impact-of-covid-and-climate-change-on-the-collectibles-market-1234602334/. Accessed 7 January 2023.

Smart, John Ferguson. *BDD in Action: Behaviour-driven Development for the Whole Software Lifecycle*. Manning, 2014. Accessed 7 January 2023.