# Card Collector Application — Software Requirements Specification and Project Development Plan

Version 0
Group 9
COMPSCI 4ZP6A
Dr. Mehdi Moradi
October 10th, 2025

| | | |
|---|---|---|
| Tania Da Silva | dasilt2@mcmaster.ca | 400314513 |
| Norman Liang | liangl20@mcmaster.ca | 400377856 |
| Elite Lu | lue13@mcmaster.ca | 400364692 |
| Ishpreet Nagi | nagii@mcmaster.ca | 400394382 |
| James Nickoli | nickolij@mcmaster.ca | 400399883 |
| Kenneth Ong | salimk4@mcmaster.ca | 400395100 |
| Geon Youn | young2@mcmaster.ca | 400362097 |

# 0. Document Information

## 0.1 Contribution History

| Authors | Sections |
|---|---|
| Tânia Da Silva | 1.2, 1.3, 1.4, 1.5 |
| Norman Liang | 1.4, 1.5, 1.7, 2.1, 2.4 |
| Elite Lu | 1.4, 1.5, 2.1, 2.2 |
| Ishpreet Nagi | 1.3, 1.4, 1.5, 1.7, 2.1 |
| James Nickoli | 0.4, 1.1, 1.2, 1.4, 1.5, 1.6, 1.7 |
| Kenneth Ong | 1.4, 1.5, 2.3, 2.5, 2.6 |
| Geon Youn | 1.3, 1.4, 1.5, 1.6, 2.5 |

## 0.2 Revision History

| Version | Authors | Description | Date |
|---|---|---|---|
| 0 | Everyone | Initial document | Oct. 10, 2025 |

## 0.3 Glossary

| | |
|---|---|
| TCG | TCG is an abbreviation for “Trading Card Game”, which is a set of cards that are typically used in a turn-based strategy game where players build custom decks of cards to compete against each other. These cards are also often traded amongst players. |
| (Card) Collection | A group/collection of a set of TCG cards. |
| Pokémon Set | A collection of unique Pokémon cards released together, forming a specific set list, theme, and release date. |
| Raw/Graded | Cards can get sent to grading companies that will typically rank a card’s quality from 0 to 10. A raw card is a card that hasn’t been graded yet. |

## 0.4 Purpose of the Project

### 0.4.1 Background

Buying, selling, and trading used items is a great way to reduce the amount of energy and resources used to create new products. Selling owned items to make money to buy other items often leads to a loss in value in exchange for the convenience. However, a trade where both parties want each other's similarly valued items can take less time and preserve more value. One of the problems is that it traditionally takes a lot of time to look through items to find a trade in person. Another problem is offering trades on an online platform, when it's unclear what the other person wants or if they even want to trade.

### 0.4.2 Problem Statement

Tracking collections manually (e.g. using a spreadsheet) is inefficient, and it takes a large amount of time to find other collectors with whom to trade. Some collectors want to reduce the value they lose by selling old items to buy new items, and would rather trade directly. Additionally, it should be easier for smaller collectibles sellers that don't operate on a large enough scale to warrant using existing software geared towards full-time businesses to track financials and list items for sale on major platforms (example: eBay, TCGPlayer, Cardtrader, Cardmarket).

### 0.4.3 Proposed Solution

Our proposed solution is to:

- Allow users to add collectibles to the collection using their cameras, using computer vision with a classification model to automatically distinguish and organize the unique items.
- Provide collectors a way to more efficiently keep track of, search, manage their collections, and discover potential trades with other users.
- Sell business-oriented features to small businesses at an affordable rate, such as keeping track of prices paid and received for items, exporting subsets of their collection to list items for sale quickly, and generating financial reports to make it easier to report capital gains/business income.

# 1. Software Requirements Specification

## 1.1 Competitor and market analysis

### 1.1.1 Market Stats and Insights

Over 17 million sellers have accounts on eBay (source)—many potential customers that may pay for our app. The Pokémon Company has printed over 75 billion cards (source)—lots of cards in the world to be cataloged. eBay acquired a selling platform dedicated to TCG cards for ~$295 billion (source)—there is a large secondary market for these collectibles. 7% of Americans collect trading cards (source)—many potential users.

### 1.1.2 Competitors and Why They Don't Fill The Needs

- sortswift: Scanning is not continuous, high pricing ($200+ monthly).
- Shiny: Paid scanning, no selling tools.
- pokecard dex: Paid scanning, slow scanning.
- Withmascot: Purely for cross-platform listing.
- Card Dealer Pro: Desktop exclusive, developed for use with document scanners, monthly scan limit.
- CollX: Broken scanning.
- Collectr: Scanning is a premium feature; scanning workflow is slow.
- Card Camp: Desktop only, slow scanning.
- TCG Automate: Desktop only, high pricing.
- Tcgscan: Slow, paid.
- pricecharting app: Collection categorization is a paid, expensive business tier that only works with Shopify.
- Pokelenz: No selling tools.

### 1.1.3 Competitor Analysis

Broadly, the competing solutions are either dedicated towards higher volume sellers, who can justify spending $20-200 monthly to improve their business operations, or non-business-running collectors. We have identified a market need for an affordable solution dedicated to low-volume sellers who could justify only $5-10 a month for their business.

Out of the collectibles apps with a scanner, we found none that were free and had speed/accuracy enough to catalogue a whole collection faster than manual entry. If we can build a sufficient model that runs locally, we can provide it for free and beat the other competing solutions. No researched solutions had a feature to link users based on items they wish to trade. This will be a unique feature that sets us apart.

## 1.2 The Client and Other Stakeholders

The main internal stakeholders are all of us in the development team who will be designing and creating the project. The other internal stakeholders are the course professor and teaching assistants, who will take on an advisory/guiding role, and the Department of Computing and Software at McMaster University, which may provide funding or computing resources for the project.

The main external stakeholders will be the clients and users of the application. These stakeholders will be a variety of people who are interested in one or multiple of the following: tracking collections, searching collectibles visually and textually, trading with others, sharing collections with others, recording collection purchases/sales, generating reports for tax purposes, viewing pricing information, and listing items for sale on external platforms. Other external stakeholders will be affected by the project, but will not be directly interacting with it. This includes our peers, viewers of an early and final presentation of the application, anyone who comes to learn about the project at the Capstone Expo, and collectors who may interact with users of the project who used it to help decide on the price of an item to sell.

### 1.2.1 Personas

**NAME**: Sandra Smith Anne
**PERSONAL DETAILS/DEMOGRAPHIC**: 26 years old, full-time job, lives with friends for cheaper rent.
**RELEVANT SKILLS:** Uses her phone and computer regularly. Comfortable with apps and websites.
**BEHAVIOURAL DETAILS:** Collects cards as a hobby and is only aware of the grading system by proxy. Has never graded a card and does not care about the official grading system. Tends to hoard cards and is not willing to let go of duplicates easily. Willing to spend money on a card just because she likes it.
**FRUSTRATIONS:** Has multiple small binders to store cards and finds it hard to remember which binder has which cards. Has trouble keeping track of duplicates.
**GOALS:** Wants to keep track of her collection of cards. Her favourite Pokémon is Nihilego, which is not featured on many cards. Because of this, Sandra will collect multiples of the same card to fill out her binders. She wants to be able to indicate that she has multiples of a card in her collection. She also wants to be able to indicate the condition of each card, as some duplicates are in better condition than others. She wants to be able to differentiate the binders by what is on the cover and not their specific location in her room.

**NAME:** Sea Neumen
**PERSONAL DETAILS/DEMOGRAPHIC**: 30 years old, full-time job, lives with partner in a house.
**RELEVANT SKILLS**: Deep knowledge of Pokémon cards and the grading system. Comfortable with apps and websites. Has purchased cards online before.
**BEHAVIOURAL DETAILS**: Collects cards as a hobby. Gets his cards professionally graded and values the official grading system. Has a large collection of cards and is willing to spend money on rare cards. Likes to keep his cards in a box until he can sleeve them.
**FRUSTRATIONS**: Forgets which cards he has in his box and occasionally buys duplicates. Frustrated by how long it takes to find someone selling a specific card he wants.
**GOALS**: Wants to keep track of his cards and their grades. He likes collecting rare cards, such as misprints and error cards, and wants to be able to indicate this in his collection. Wants to easily be able to change the location of his cards once they get moved from the box to a binder or slab. Willing to sell and trade cards, but mostly wants to buy other cards.

**NAME:** Jimmy “Trudy” Evans
**PERSONAL DETAILS/DEMOGRAPHIC:** 22 years old, needs money for tuition, lives with parents.
**RELEVANT SKILLS:** Comfortable with apps and websites. Has purchased and sold cards online before.
**BEHAVIOURAL DETAILS**: Tends to be confrontational when people do not want to sell to him. Has a deep understanding of the grading system and knows how to find cards that are undervalued.
**FRUSTRATIONS**: Doesn't have a lot of money and wants to find a way to make money quickly.
**GOALS**: Does not care about starting a collection. Wants to be able to quickly find cards that he can resell for a profit.

## 1.3 Project Constraints

- Constrain the model to run locally on the user’s device rather than a dedicated server.
    - Ensure that the model is lightweight enough to run on a device as powerful as a phone while maintaining its performance in classification accuracy and speed.
    - This constraint will allow us to maintain security and the privacy of the user, as it ensures all images and related data are stored locally on the user’s device.
- Constrain the application so that it does not mandate the user to share their collection online with the application’s user base.

- Constrain messaging and payment handling of the trading process to outside our application.
- Constrain this app to only house a collection of Pokémon cards.
- For this capstone, we'll be constraining the app to the English language.
    - All UI elements will be in English, but will be coded to support future language additions. Only English Pokémon cards are supported.
- Constrain the app to only store taken images locally for non-paid users, and only upload taken images for items up for trade, or for paid users.
    - These real pictures will be required for tradeable items to provide trust between users and incentivize users to take good pictures.

## 1.4 Functional Requirements

### 1.4.1 Priority 0 (Minimum Viable Product)

- Identify and classify a Pokémon card from an image.
    - An object detection model to identify Pokémon card locations in an image.
        - Users should be able to provide the image either (1) manually or (2) in real-time through their camera (option 2 is preferred).
    - A perceptual hashing algorithm to classify English Pokémon cards from the identified Pokémon cards by the object detection model.
    - A way to perform this task locally on the user's device.
- A way to manually identify a card if the system can't classify it.
- A way to select the card's condition (raw condition or graded) and type (ex. holo/reverse holo) and enter additional notes.
- A way for users to search through the Pokémon TCG database either directly, by describing their card with natural language, or by set.
- Basic web and mobile GUI:
    - A way for users to add cards to their collection.
        - Automatically through their camera or an uploaded image.
        - Manually by searching through the Pokémon TCG database.
    - A way for users to view their added cards.
    - A way for users to search through their collection.
    - A way for users to delete cards from their collection.

### 1.4.2 Priority 1 (Next Feature Set)

- Users should be able to create a public profile that can be viewed by other users who have an account.
- Users should be able to add their contact information to their profile.
- Users should be able to mark certain items in their collection to be available for trade.
- Users should be able to rate each other after completing a trade.

- The system should have a mechanism that can deal with logging duplicate entries.
- The system should allow users to register for an account.
- The system should be able to back up users' collections to a central server.
- The system should be able to create trade recommendations between users' collections.
- The system should allow users to share their collections between each other.
- Users should be able to highlight/"showcase" up to three cards on their public profile.

### 1.4.3 Priority 2 (Non-critical Features)

- The system should allow users to organize their collection of cards into groups and subgroups.
- The system should allow users to label their groups and subgroups to reflect their real-world location.
- The system should allow creating tags and assigning tags to items in the collection.
- The system should allow users to verify cards by displaying their scanned card rather than a stock image of the card.
- Users should be able to put cards on hold when they're being traded/sold.
- Users should be able to remove cards on hold after they've been traded/sold or cancelled.
- The system should automatically move cards from one user's collection to the other's after a trade/sell.
- Users should be able to enable an option to view pricing information.
- Users should be able to enter the price paid for or sold at for collection items.
- The system should allow scanning for multiple cards at a time when the user is trying to add a card to their collection.
- Users should be able to report other users for scam, inappropriate behaviour, etc.

### 1.4.4 Priority 3 (Future Application Features)

- Expand the computer vision model to other collectible items such as K-pop albums, photocards, comics, coins, stamps, etc., and other language Pokémon cards.
- Allow users to have different collection categories in their profiles.
- Users can enter shipping and tax costs to amortize across a sub-collection of items.
- Users can pay for a monthly subscription that gives them access to business-oriented features.

- Paid users can export collections to external selling platforms, see more up-to-date prices, import sales data exports from external platforms to update collections, and generate reports for taxes and insights.

## 1.5 Non-functional Requirements

### 1.5.1 Performance Requirements

- The card identifying mechanism should complete within 2000 milliseconds.
- Users should be able to retrieve their collection from the database within 1000 milliseconds after submitting a query.

### 1.5.2 Security, Privacy, and Legal Requirements

- The application will adhere to the Personal Information Protection and Electronic Documents Act (PIPEDA).
- User authentication should support secure log-ins.
- The user's contact information should not be publicly displayed by default and will only be shared with other users they wish to trade with.
- Users should not be able to view others' card collections that have not been explicitly made public.
- Users shall not be able to create usernames with profanities or otherwise inappropriate language.

### 1.5.3 Reliability Requirements

- The card identifying mechanism should have at least a 98% accuracy rate.
- The machine learning model and database should be able to be updated to include newly released sets and changes for existing cards' accuracy
- The server should have an average uptime of at least 98%.

### 1.5.4 Portability Requirements

- The application will run on the latest iOS (iOS 26) and Android version.
- The UI should scale to all commonly used aspect ratios without losing information or functionality.
- The machine learning model should operate locally on the user's device.

### 1.5.5 Usability Requirements

- Should be at minimum WCAG Level A compliant, with Level AA as our goal.
- The application should be usable with no previous formal training.
- The user should be able to change the labels assigned to the cards in at most four clicks (collection → specific collection → card → edit labels).

### 1.5.6 Look and Feel Requirements

- The application shall have a cohesive brand identity that is consistent across all menus.

- Design should be simplistic and intuitive, focusing on readability and usability.
- The font used across the application should be dyslexia-friendly to ensure easy user readability.
- Colour choice should not be jarring, focusing on ease of viewing and accommodating common visual impairments.

## 1.6 Data and Metrics

The goal of the ML aspect of the project is to identify Pokémon cards from the camera feed and classify which card it is. We will attempt to approach this with various techniques (CNNs, object detection, OCR models, and perceptual hashing). We will use the high-resolution scan images of cards and potentially their labelled metadata. We will use https://docs.pokemontcg.io for Pokémon cards. We will measure both the speed and accuracy of identification. Our goal is to have on-device identification run in under 2000 milliseconds on low-end devices. We will generate ROC curves to measure the performance of our model by analyzing different subsets of cards, such as cards with unique artwork and cards that share artwork with other cards from different sets. We will aim for an ROC AUC of 0.95.

## 1.7 Predicted Risks and Issues

### 1.7.1 Risks

- Constraining the model to run locally on the user's device, compared to a dedicated server.
    - If we cannot achieve this goal, it will result in requiring a constant connection to a server to run the model, which can become impractical for our scope in terms of price for such a heavy task.

### 1.7.2 Issues

- The accuracy of our computer vision model relies entirely on the user's camera quality.
    - Many differences between the trading cards can be subtle, such as holographic cards, fine prints on the card, shadows and shading differences, etc.
    - Glare and unideal image quality can hinder the model's accuracy.
    - Instructions will have to be created for the user for card and camera positioning to obtain the most optimal picture for classification, paired with the ability to correct mistakes made by the model.
- Users can leave insincere ratings for other users they may have traded with.
    - Address this manually with a reporting and human review system, or automatically with a reputation system.
- Users entering inaccurate prices for what their items sold for to attempt to manipulate market data

    - Address this by withholding outliers and shadowbanning users who submitted outlying data that doesn't get confirmed by other users and data sources.
- Accounts can be stolen, server data can be hacked/leaked, spam attacks, vulnerability exploits, and social engineering attacks.
    - Address this by implementing account recovery and/or two-factor authentication, proper encryption of private data, rate limiting, routine updates of libraries/frameworks, and protocols for how we will interact with external communications.

# 2. Project Development Plan

## 2.1 Team Meeting and Communication Plan

After thorough deliberation among ourselves as a team, we have decided upon the following tools and applications for the use of communication, planning, work sharing, and meetings.

### 2.1.1 Discord

- Primary communication channel for meetings and virtual discussions.
- We created a capstone Discord server with separate messaging/voice chat channels for the various subteams to allow for organized and easy communication among sub-team members.
- Discord also allows members to reach out to each other directly in private, one-on-one messaging conversations.

### 2.1.2 GitHub

- Main platform for code/document sharing.

### 2.1.3 Google Drive/Docs

- Google Docs and, by extension, Google Drive will be our platform for document creation, like the SRS and Design Document, and meeting notes.

## 2.2 Team Member Roles

These roles are the responsibility of the individual, but every member will aid in other roles not specified.

| Member | Main Role | Responsibilities |
|---|---|---|
| Tânia | Front-end Lead | Implement the UI/UX of the application.<br>Write unit tests for various components to ensure quality in isolation. |
| Norman | Data Lead | Deal with database management and SQL calls. |

| | | |
|---|---|---|
| | | Manage the accounts database and data associated with each account. |
| Elite | Design Lead | Design the UI and the application interactions. Work mainly in Figma and support the front-end lead with the implementation of the application. Consider additional items such as accessibility (A11y). |
| Ishpreet | Back-end Lead | Link the code between the database and the front-end. Write unit tests for various components to ensure quality in isolation. Deal with account security.<br>Write unit tests and integration tests for various back-end components to ensure model and front-end code are properly linked. |
| James | Project Manager | Track team milestones and define the scope of the project. Hold scrum meetings weekly to determine the weekly deliverables, scope, etc. |
| Kenneth | QA Lead | Have the “final say” for code being pushed into production. Design various tests, code, or interaction-based, to ensure deliverables meet or exceed quality standards. Test the various interactions between components to ensure interactions work as intended |
| Geon | ML Lead | Find and prune through the dataset for the cards and other collectables. Design a model to learn the various characteristics of cards and identify the cards. |

## 2.3 Workflow Plan

### 2.3.1 GitHub

- GitHub will be the primary location where all code will be stored.
- We have created a new organization called “collectiblescapstone” and a repository called “CollectiblesApp”.
- We will use the monorepo pattern, where the source code for the web and mobile application, along with other files used to train the ML model, are all stored in a single repository.
- The main branch corresponds to the development environment and is protected by a branch rule that prevents any merges to it.

- We have set it up such that any changes to the main branch need to go through a pull request and should be approved by at least one team member who is not also the author of the pull request.

### 2.3.2 JIRA/Issue Tracking

- JIRA will be used to track issues and development items.
- An EPIC will be created for each phase/scope of the feature development
- In each epic, backlog items will be created that reflect exactly what will be done by the developers.
- Backlog items will be created such that it is small enough to be completed by a single developer over a single sprint.
- We aim to have 2-week sprints, where the first week is focused more on the development side and the second is more on the testing side.

### 2.3.3 Figma

- Figma will be used to design the user interface and user experience of the application.
- We plan to use reusable components in Figma that will also reflect the reusable components in React.
- If we choose to use a UI library, then we can use the Figma extension for that UI library to design the UI/UX of the app.
- Developers will refer to the Figma designs when creating the UI in React.

### 2.3.4 Machine Learning Related Workflow

- The dataset will be uploaded to GitHub, where every team member will have access to it.
- Training will be done either locally or using the resources provided by the faculty.
- Once we have a trained model, we can use it in our application using Tensorflow.js/JS-PyTorch.

## 2.4 Proof of Concept Demonstration Plan

We plan for our proof-of-concept demonstration to include a minimal mobile/web user interface, with a focus on showcasing the card identification feature. The card identification demonstration will involve using a camera and computer vision technology to identify which card is scanned through the application. We will also demonstrate a basic implementation of the card collection portion of the application using the identified card, which will be stored under the user's list of collected cards.

One potential risk with the proof of concept demonstration is that we may not have sufficient resources or time to achieve the level of speed/accuracy that we strive for in the AI card identification implementation. To overcome this, we will still show the progress we've made up to that point, but we may have to edit the video to match our vision of the final product or show a prototype instead. We also need to account for the potential complexity of integrating the card database with the machine learning model. Suppose we find it infeasible for our demonstration to identify every card in the database. In that case, we may focus solely on one card, which we will train our model to identify, isolated from the database.

## 2.5 Technology

### 2.5.1 Web/Mobile Application

- The web application will be built using the Next.js framework, which allows us to leverage React.js and Tailwind CSS for our frontend and JavaScript (Node.js/Next.js) for our backend.
- Jest will be used to write unit tests for both frontend and backend to ensure code quality and catch any bugs/side effects early.
- Capacitor.js will be used to port the web application to mobile platforms and take advantage of native features.

### 2.5.2 Database

- The Pokémon card data, application data, and user data will be stored on a PostgreSQL database server (possibly hosted on Supabase).
- Prisma ORM will be used to manage and simplify database workflows.

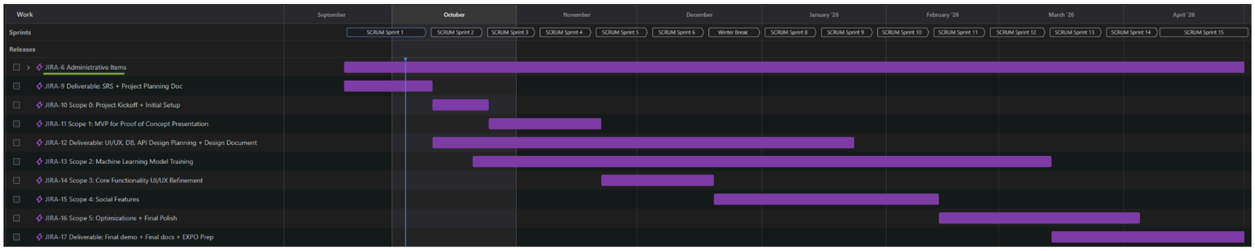
### 2.5.3 Machine Learning

- We will train our models using Python (through Jupyter Notebooks) and use the following ML libraries: PyTorch, HuggingFace, and fast.ai.
- Because training models is quite computationally expensive, we will use the CAS department GPUs.

## 2.6 Project Scheduling

### 2.6.1 Gantt Chart

JIRA Timeline


Work
September
October
November
December
January '26
February '26
March '26
April '26
Sprints
SCRUM Sprint 1
SCRUM Sprint 2
SCRUM Sprint 3
SCRUM Sprint 4
SCRUM Sprint 5
SCRUM Sprint 6
Winter Break
SCRUM Sprint 8
SCRUM Sprint 9
SCRUM Sprint 10
SCRUM Sprint 11
SCRUM Sprint 12
SCRUM Sprint 13
SCRUM Sprint 14
SCRUM Sprint 15
Releases
JIRA-6 Administrative Items
JIRA-9 Deliverable: SRS + Project Planning Doc
JIRA-10 Scope 0: Project Kickoff + Initial Setup
JIRA-11 Scope 1: MVP for Proof of Concept Presentation
JIRA-12 Deliverable: UI/UX, DB, API Design Planning + Design Document
JIRA-13 Scope 2: Machine Learning Model Training
JIRA-14 Scope 3: Core Functionality UI/UX Refinement
JIRA-15 Scope 4: Social Features
JIRA-16 Scope 5: Optimizations + Final Polish
JIRA-17 Deliverable: Final demo + Final docs + EXPO Prep