

# **Magic: The Gathering Price-Checking Bot**

**An Automation Solution Using Robotic Process Automation (RPA) Technology**

**STUDENTS:**

**Pavaloiu Bianca-Anastasia**

**Gradinariu Rares-Ioan**

**Andrei Mihai Alexandru**

**GROUP: 1241A**

# Table of Contents

1. Executive Summary
2. Introduction
  - 2.1. Project Background
  - 2.2. Objectives and Benefits
3. Problem Definition
4. Current Approaches
  - 4.1. Manual Methods
  - 4.2. MtgDesktopCompanion
  - 4.3. Online Platforms
5. Project Scope
  - 5.1. Project Objectives
  - 5.2. Business Objectives
  - 5.3. Project Constraints
  - 5.4. Critical Success Factors
6. System Requirements
  - 6.1. Financial Requirements
  - 6.2. Human Resources
  - 6.3. Technological Requirements
7. Technological Approach
  - 7.1. High-Level Architecture
  - 7.2. Input Module
  - 7.3. Web Scraping Module
  - 7.4. Output Module
  - 7.5. Error Handling
8. Development Process
  - 8.1. Feasibility Study
  - 8.2. Technical Design
  - 8.3. Implementation
  - 8.4. Testing
  - 8.5. Deployment
9. Risk Management
  - 9.1. Risk Identification
  - 9.2. Risk Analysis
  - 9.3. Mitigation Strategies
  - 9.4. Monitoring and Control
10. Workings of the Project
11. Stakeholders
  - 11.1. Primary Stakeholders
  - 11.2. Secondary Stakeholders

## 12. Quality Assurance

### 12.1. Plan

### 12.2. Metrics

## 13. Future Enhancements

## 14. Conclusions

## 15. References

## 16. Annexes

### 16.1. Annex 1: Workflow Diagram (State-Based System)

### 16.2. Annex 2: Requirements Hierarchy

### 16.3. Annex 3: Risk Management Table

### 16.4. Annex 4: Class Diagram

### 16.5. Annex 5: Sequence Diagram

### 16.6. Annex 6: Gantt Chart

### 16.7. Annex 7: Development Process Flow

### 16.8. Annex 8: Main sequence

## 1. Executive Summary

Magic: The Gathering (MTG) is one of the world's most popular trading card games, with a vibrant community of collectors and players. Card prices evolve constantly, driven by rarity, market interest, and tournament results.

This project proposes an MTG Price-checking Bot built in UiPath, leveraging Cardmarket.com as a data source. By automating the retrieval of real-time price information and updating an Excel-based decklist, this bot saves time, reduces errors, and simplifies collection management for MTG enthusiasts.



## 2. Introduction

### 2.1. Project Background

Magic: The Gathering's global popularity has spawned numerous online marketplaces and tools to help players buy, sell, and track card values. European traders largely prefer Cardmarket.com. However, manually searching for each card's price on Cardmarket and updating spreadsheets can be tedious and prone to errors.

Robotic Process Automation (RPA), particularly UiPath, provides an elegant solution for automating repetitive tasks, effectively offloading the burden of manual data entry. This synergy between a popular trading card game and modern RPA technology forms the basis for this project.

### 2.2. Objectives and Benefits

- **Objectives:**
  - Provide an automated workflow that reads card names from an Excel file.
  - Retrieve current pricing data from Cardmarket.com.
  - Update the Excel file's "Price" column with real-time values.
- **Benefits:**
  - **Time Savings** – Drastically reduce the manual effort of searching and recording prices.
  - **Accuracy** – Eliminate human error in data transcription.
  - **Scalability** – Handle larger decks or collections without significantly increasing processing time.

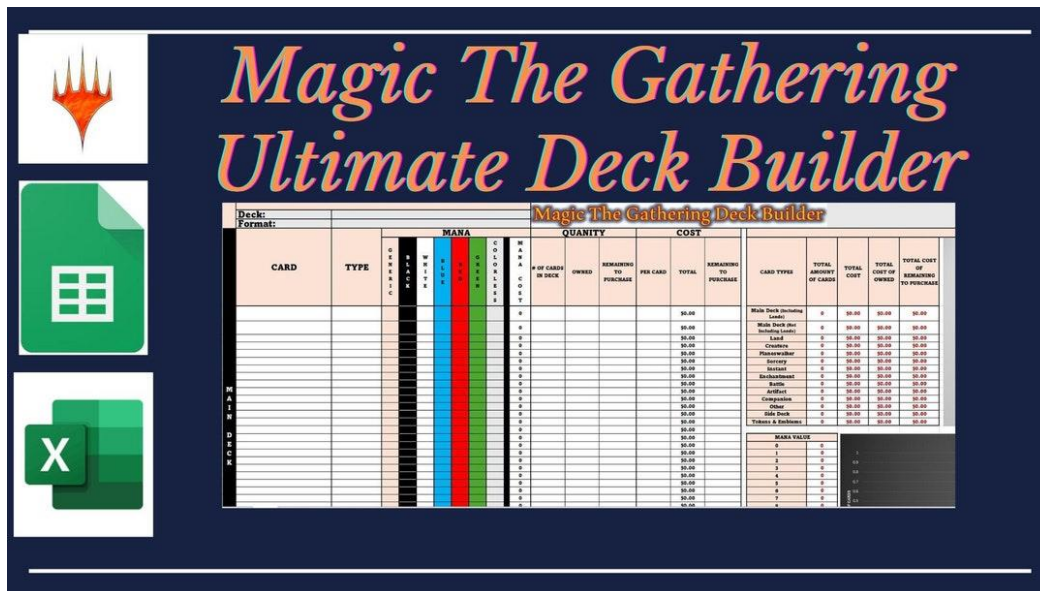
## 3. Problem Definition

Collectors often rely on slow, manual methods to keep track of card values, especially if their preferred source is Cardmarket.com. Repeatedly searching multiple individual cards is laborious, and spreadsheets become outdated quickly. An automated solution addresses these inefficiencies by periodically updating an Excel decklist and providing up-to-date, reliable pricing information for better collection and trading decisions.

## 4. Current Approaches

### 4.1. Manual Methods

Users often look up individual cards on Cardmarket or MTGGoldfish, then input the data into spreadsheets. This approach is direct but highly susceptible to transcription errors and time-consuming re-checks.



## 4.2. MtgDesktopCompanion

This is an open-source desktop tool available on GitHub (developed by nicho92). It offers features such as collection management, deck building, and price tracking. The tool integrates multiple data sources, including TCGPlayer and Cardmarket.com

However, it has several drawbacks:

- **Hard to Install:** Lacks a proper installer and requires manual setup.
- **Requires Java Knowledge:** Users must be familiar with Java Runtime Environment (JRE) and dependencies.
- **Console Configuration:** Requires command-line setup, which can be intimidating and error-prone for non-technical users.
- **Clunky Interface:** The user interface and workflow are cumbersome, making it less accessible.
- **Prone to Errors:** Bugs and misconfigurations can lead to unreliable results.

The screenshot shows the MTG Desktop Companion interface. The main window lists cards with columns for Card, Expansion, Price, and Daily%. Below this, there are several smaller panels: 'Trends' showing price history for selected cards, 'Editions Prices' showing prices for different card sets, and 'Winners/Losers' showing results for specific tournaments. The interface is designed for tracking card prices and tournament performance.

## 4.3. Online Platforms

Platforms like MTGGoldfish, EchoMTG, or Deckbox streamline price tracking but may not fully integrate with Cardmarket.com or may prioritize the U.S. market. Users wanting direct European price data often find themselves resorting to more manual solutions or incomplete automation.

The screenshot shows the MTGGoldfish website interface. The main section displays a Modern League 2025-01-02 deck page. It includes the deck name, format, event, and date. Below this, there is a table showing the deck list with columns for Place, Deck, Pilot, Tabletop Price, MTGO Price, and Toggle Deck. The deck is named 'Naya Energy' and is piloted by '300'. The table shows the deck's performance in various tournaments, including a win in the Modern League 2025-01-02. The right side of the page features a 'Metagame Summary' table showing the percentage of decks using various cards in the tournament.

## 5. Project Scope

### 5.1. Project Objectives – Annex 2

1. **Automate Price Retrieval**  
Develop a UiPath bot that navigates Cardmarket.com, searches for each card, and extracts its price.
2. **Real-Time Updates**  
Populate the “Price” column in an Excel file dynamically, ensuring availability of current market values.
3. **Reduce Manual Effort**  
Limit user intervention to simply running the bot, achieving at least an 80% reduction in time spent on price updates.

### 5.2. Business Objectives

- **Competitive Advantage:** Possibly extend this bot for broader data analytics or multi-market support.
- **Improved User Experience:** Provide collectors with a fast, easy way to monitor the monetary value of their decks.
- **Cost Savings:** Free up time that would have been spent on repetitive tasks.
- **Easy to use:** Friendly to use and set-up by new users with a focus on non-technical persons.

### 5.3. Project Constraints

- **Adherence to Cardmarket.com TOS:** The bot must not exceed site scraping limits or violate any policies.
- **Performance:** Even large decklists (100–500 cards) should ideally complete scraping in a matter of minutes.
- **User-Friendly:** Interface and setup must be straightforward for both technical and non-technical audiences.

### 5.4. Critical Success Factors

- **Accurate Card Matching:** The bot should handle slight naming discrepancies or expansions without scraping the wrong listing.
- **Error Handling:** Robust logging is essential when a card cannot be found or if the site is temporarily unavailable.
- **Documentation & Support:** Clear manuals, tutorials, or user guides will drive adoption and correct usage.



## 6. System Requirements

### 6.1. Financial Requirements

- **License:** UiPath Community Edition can be used to avoid extra license costs.
- **Data:** Cardmarket.com provides free, publicly available listings. Additional APIs or premium features could be explored in the future if required.

### 6.2. Human Resources

- **Developer:** Skilled in UiPath, web scraping, and basic Excel automation is sufficient to build and maintain the project.
- **Maintenance:** Post-launch, minimal technical support may be required to fix issues triggered by major UI changes on Cardmarket.com.

### 6.3. Technological Requirements

1. **UiPath Studio** (Community or Enterprise edition).
2. **Microsoft Excel** (for input file and reading/writing decklist data).
3. **Stable Internet Connection** (necessary to access Cardmarket.com).

## 7. Technological Approach

### 7.1. High-Level Architecture – Annex 1

The system follows a state-based workflow, ensuring robustness and error handling throughout the process. Each key operation is represented as a state, with transitions based on success or failure conditions.

**Key states include:**

- **Idle:** The initial state where the system is waiting for the user to load a deck file.
- **LoadingDeck:** Reads the user-provided deck file, verifying its format and contents.
- **DeckLoaded:** Confirms that the deck file was successfully loaded, transitioning to the next state to fetch card prices.
- **FetchingPrices:** The core operation where the bot searches Cardmarket.com for card prices. Success or failure transitions are handled at this stage.
- **PricesUpdated:** Represents the successful fetching of all prices, transitioning to saving the updated file.
- **SavingDeck:** Updates the deck file with retrieved prices and saves it.
- **DeckSaved:** The final state, indicating that the deck was successfully updated and saved.
- **Error:** Handles any issues that occur during the process, such as invalid file formats, failed searches, or file save errors.

## 7.2. Input Module - Annex 4

### 1. Load Deck File:

- Users provide an Excel-based deck file containing card names and any additional metadata.
- The bot validates the file format, ensuring it meets the expected structure.
- If the format is invalid, the system transitions to the **Error** state, providing feedback to the user.

## 7.3. Web Scraping Module - Annex 4

### 1. Fetch Card Prices:

- The bot enters the **FetchingPrices** state, where it searches for each card on Cardmarket.com.
- For each card:
  - If a price is found, it transitions to the **PricesUpdated** state.
  - If a card cannot be found (due to a search logic error or mismatch), it logs the error but continues processing.
- If the process encounters critical issues (e.g., network errors or unexpected site structure changes), it transitions to the **Error** state.

## 7.4. Output Module - Annex 4

### 1. Save Updated Deck:

- After all prices are successfully fetched, the bot transitions to the **SavingDeck** state.
- The updated prices are written back to the deck file, and the total deck cost is calculated.
- If the file save operation succeeds, the system transitions to the **DeckSaved** state, indicating the process is complete.
- If an error occurs during file saving, the system transitions to the **Error** state.

### 2. Handle Errors:

- Any failure (e.g., invalid file format, search errors, or file save errors) transitions the system to the **Error** state.
- The user is provided with a detailed error message and the option to retry or reset the workflow.

## 7.5. Error Handling - Annex 4

The bot incorporates robust error-handling mechanisms to ensure reliability:

- **Invalid File Format:** If the input file does not match the expected format, the bot transitions to the **Error** state with a descriptive message.
- **Search Failures:** Cards not found during the fetching process are logged, and the bot continues to process other cards.
- **File Save Errors:** If the updated deck cannot be saved, the bot provides the user with options to retry or investigate the issue.

## Workflow Overview

1. **Idle** → Await user action to load a deck file.
2. **LoadingDeck** → Verifies the input file format.
  - If valid: Transition to **DeckLoaded**.
  - If invalid: Transition to **Error**.
3. **DeckLoaded** → Begins fetching card prices.
4. **FetchingPrices** → Attempts to fetch prices for all cards.
  - On success: Transition to **PricesUpdated**.
  - On partial success: Log errors and continue.
  - On failure: Transition to **Error**.
5. **PricesUpdated** → Proceeds to save the updated deck.
6. **SavingDeck** → Attempts to write updates to the deck file.
  - If successful: Transition to **DeckSaved**.
  - If unsuccessful: Transition to **Error**.
7. **DeckSaved** → Workflow completed successfully.
8. **Error** → Logs the issue and provides options for retry or reset.

This state-based design ensures a clear and logical flow, making the system robust and user-friendly while maintaining flexibility to handle errors gracefully.

## 8. Development Process

### Annex 6

#### 8.1. Feasibility Study – Annex 7

- **Technical Viability:** UiPath can reliably navigate Cardmarket.com's structure and read the relevant HTML elements.
- **Performance Tests:** A sample deck of ~50 cards completes the scraping in 1–2 minutes on a typical home network.

#### 8.2. Technical Design

- **Modular Workflow:** Separate code segments handle reading Excel, scraping prices, and writing results back.
- **Exception Handling:** The system gracefully logs or flags cards that are unresolvable (e.g., unusual tokens or misprints).

#### 8.3. Implementation

- **UiPath Activities:**
  - **Read Range** to load Excel data.
  - **Type Into** and **Click** to navigate Cardmarket.
  - **Get Text** to extract the card price.
  - **Write Cell** to populate the Excel output.
- **Optimization:** Minimizing browser refreshes and reusing sessions for multiple cards.

#### 8.4. Testing

- **Unit Testing:** Single-card checks to confirm the correct prices are pulled.
- **Integration Testing:** Multiple decks of varying sizes to ensure correct totals and identification of missing cards.
- **User Acceptance Testing (UAT):** Gathering feedback from actual MTG players on usability, clarity, and reliability.

#### 8.5. Deployment

- **Package Distribution:** The workflow is published from UiPath Studio.
- **Documentation:** Straightforward guide on how to set up Excel columns, run the bot, and interpret results.

## 9. Risk Management

### Overview - Annex 3

Risk management is a crucial aspect of the MTG Deck Price-Checking Bot project, addressing potential challenges and ensuring the project progresses smoothly. The identified risks are categorized by their triggers, probability, impact, and the strategies to mitigate or handle them. Here's a breakdown of the key points:

### 9.1. Key Risks Identified

The project has identified five major risks:

1. **Changes in Website Structure or Anti-Scraping Measures (R01)**
  - *Summary:* Cardmarket.com may introduce changes to its website layout or security measures, such as rate-limiting or CAPTCHA, which can disrupt scraping operations.
  - *Impact:* High likelihood of delays and additional development time (10 days impact, 70% probability).
  - *Mitigation:* Develop flexible scraping logic that adapts to changes.
2. **Inconsistent File Formats (R02)**
  - *Summary:* Errors can arise when processing deck files due to mismatched column headers or unsupported Excel formats.
  - *Impact:* Moderate impact (7 days, 50% probability).
  - *Mitigation:* Standardize input formats and validate the structure of deck files.
3. **Integration Bugs Between Modules (R03)**
  - *Summary:* During the integration phase, dependencies between modules may cause delays if bugs or incompatibilities arise.
  - *Impact:* Low impact (5 days, 30% probability).
  - *Acceptance:* The team will accept delays and allocate additional time for integration testing.
4. **Dynamic Website Updates or Data Retrieval Failures (R04)**
  - *Summary:* Challenges in retrieving data due to rate-limiting mechanisms, IP blocking, or incomplete price data caused by frequent updates to the website.
  - *Impact:* Moderate impact (6 days, 40% probability).
  - *Mitigation:* Use retry mechanisms, proxy rotation, and user-agent strategies.
5. **Insecure Data Handling (R05)**
  - *Summary:* Potential issues with sensitive data handling, such as unencrypted storage or improper user-agent configurations, which could lead to security vulnerabilities.
  - *Impact:* Low impact (4 days, 20% probability).
  - *Avoidance:* Use environment variables and encryption to secure sensitive configurations.

## 9.2. Risk Analysis

The risks were evaluated based on **probability**, **impact**, and their overall risk level:

- **High Priority Risks:**
  - **R01** is critical due to its high likelihood and significant impact.
- **Moderate Priority Risks:**
  - **R02** and **R04** are less likely but can still disrupt project progress.
- **Low Priority Risks:**
  - **R03** and **R05** are less likely to occur and have lower impact.

## 9.3. Mitigation Strategies

Each risk has a corresponding strategy to minimize its impact:

- **R01 (Website Changes):** Flexible scraping logic to adapt to structural changes.
- **R02 (File Formats):** Input standardization ensures consistency across all deck files.
- **R03 (Integration Bugs):** Extend integration testing to identify and fix dependency issues.
- **R04 (Data Retrieval Failures):** Retry mechanisms, proxy usage, and user-agent rotation to handle dynamic updates.
- **R05 (Data Security):** Encrypt sensitive data and use secure environment variables for configurations.

## 9.4. Monitoring and Control

The project team has defined a monitoring process to regularly evaluate risks:

- **R01:** Weekly monitoring of website updates to address changes promptly.
- **R02:** Ongoing validation of input file structures during weekly testing.
- **R03:** Bi-weekly reviews of integration progress to catch inter-module issues early.
- **R04:** Weekly evaluations of scraping reliability, with adjustments made as needed.
- **R05:** Monthly reviews of encryption and sensitive data handling practices.

## Explanation

The Risk Management section ensures that potential issues are proactively addressed and do not disrupt the project's timeline or deliverables. Risks like **website changes** (R01) and **inconsistent file formats** (R02) are critical because they directly affect the core functionality of the bot. By planning for such scenarios, the team can adapt quickly and efficiently.

For example:

- Developing **flexible scraping logic** allows the bot to continue functioning even if the target website changes.
- **Retry mechanisms** ensure data retrieval remains reliable, even in the presence of rate limits or CAPTCHA mechanisms.

By categorizing risks into high, moderate, and low priority levels, the project team focuses its resources on the most impactful risks while still addressing lower-priority ones. Regular monitoring ensures that risks are evaluated continuously, and necessary actions are taken to avoid delays.

## 10. Workings of the Project

### Annex 5

This section provides a detailed walkthrough of the UiPath workflow. The following code snippets and descriptions illustrate how each step accomplishes the goal of reading card data, scraping Cardmarket.com, and updating the Excel decklist.

	A	B	C	D	E	F	G
1	Card Name	Price		Deck Total Cost		No. of Missing Cards	
2	Otter Token (Blue and Red 1/1)	0,02 €		9.21		1	
3	Sauron, the Dark Lord	2,00 €					
4	Relic of Sauron	4,45 €					
5	Frodo, Sauron's Bane	0,02 €					
6	Kitsa, Otterball Elite	2,70 €					
7	Otterball Antics	0,02 €					
8	This is a test of Missing card	Not Found					
9							
10							

### Annex 8

#### 1.1 Main Sequence (Sequence)

##### - Variables:

decklistDataTable (DataTable)  
 searchResults (String)  
 cardPrice (String)  
 index (Int32)  
 DecimalCardPrice (Int32)  
 DeckTotalCost (Double)  
 MissingCards (Int32)  
 ExcelPath (String)

##### - Activities Overview:

1. Assign -> ExcelPath = "C:\Users\haira\Desktop\SysEngProject\CardList.xlsx"
2. ReadRange (Open Excel file with Magic: the Gathering decklist)
  - DataTable = decklistDataTable
  - File = ExcelPath
  - SheetName = "Decklist"
  - AddHeaders = True
3. Assign -> DeckTotalCost = 0
4. Assign -> MissingCards = 0
5. For Each card in decklistDataTable.Rows (index = index)
  - Body (Sequence):
    - a. Navigate to cardmarket.com (Open/Attach Browser)
      - Type card name into search field

- Press Enter to search
- Delay (small pause)
- b. Check if the "Search Results" page appears:
  - If "Search Results" appear:
    - \* Attempt to find the price element
    - \* If "Sorry, no matches..." appear, then cardPrice = "Not Found"
    - \* Otherwise, extract the first available "From x,xx €" text
  - Else if the single card's "Info." tab appears:
    - \* Extract price from the "From x,xx €" text
  - If none appear, cardPrice = "Not Found"
- c. If (cardPrice = "Not Found"):
  - MissingCards = MissingCards + 1
 Else:
  - DeckTotalCost = DeckTotalCost + the numeric portion of the cardPrice
- d. WriteLine (for debugging): Print cardName + " " + cardPrice
- e. WriteCell (Update Excel "Price" column with cardPrice)
  - Cell = "B" + (index + 2).ToString
  - SheetName = "Decklist"
  - File = ExcelPath
- 6. WriteCell -> D2 = DeckTotalCost.ToString  
(writes total deck cost in cell D2)
- 7. WriteCell -> F2 = MissingCards.ToString  
(writes count of missing/unfound cards in cell F2)

## Explanation of Key Steps

### 1. Reading the Excel File

- The bot uses a **Read Range** activity to load the decklist into a DataTable. Each row represents a card with at least a "Card Name" column.

### 2. Initializing Variables

- DeckTotalCost starts at 0. Each card's price will be added to this variable.
- MissingCards is also set to 0; it increments whenever a card price is "Not Found."

### 3. Scraping Cardmarket.com

- The bot types the card name into the search bar and presses Enter.
- If the site shows "Search Results," it attempts to read the text (e.g., "From 2,50 €").
  - If the site says "Sorry, no matches for your query," the price is set to "Not Found."
  - Otherwise, the first matching card price is scraped.
- If the site shows a single card's info page, the "From x,xx €" text is scraped directly.
- The logic uses **Check App State** or other conditional activities to branch based on the page's content.



#### 4. Updating the Excel File

- The final price (or “Not Found”) is written back to Column B of the corresponding row.
- The loop continues until all cards are processed.
- After the loop, the bot writes the total cost in cell D2 and the missing card count in cell F2.

#### 5. Error/Exception Handling

- If at any point the card is not found or the site structure is unexpected, the code assigns “Not Found” to cardPrice.
- This not only prevents crashes but also clearly flags the user about which cards need attention.

In practice, this setup allows the user to keep an Excel decklist with minimal manual overhead. By running the automation on demand (or scheduling it in UiPath Orchestrator), the deck’s total value and the count of any missing cards remain current.

## 11. Stakeholders

### 11.1. Primary Stakeholders

- **End Users (MTG Players/Collectors):** Direct beneficiaries who need real-time price updates.
- **Developer/Implementation Team:** Builders and maintainers of the UiPath workflows.

### 11.2. Secondary Stakeholders

- **Cardmarket.com:** As the data source, ensuring scraping compliance is vital.
- **Potential Commercial Partners:** Companies interested in integrating or selling this bot.

## 12. Quality Assurance

### 12.1. Plan

- **Requirement Traceability:** Each project requirement corresponds to specific test cases.
- **Periodic Retesting:** Regular checks to verify the workflow whenever Cardmarket modifies its website layout.
- **Code Reviews:** Another UiPath developer might review the workflow for improvements.

## 12.2. Metrics

1. **Accuracy:** The fraction of correct prices fetched compared to manual checks (goal: 99+%).
2. **Execution Time:** How long the bot takes for an average deck of 100 cards (goal: under 3 minutes).
3. **Reliability:** Frequency of system crashes or unhandled errors (goal: fewer than 1 in 100 runs).

## 13. Future Enhancements

1. **API Integration:** Should Cardmarket offer an official API endpoint, hooking into it may yield more stable, faster results.
2. **Extended Marketplaces:** Scrape TCGPlayer, Star City Games, or Hareruya for expanded price comparisons.
3. **Multi-Currency Support:** Convert “€” to “\$” or other currencies in real time.
4. **Historical Tracking:** Store daily or weekly snapshots of deck prices for trend analysis.

## 14. Conclusions

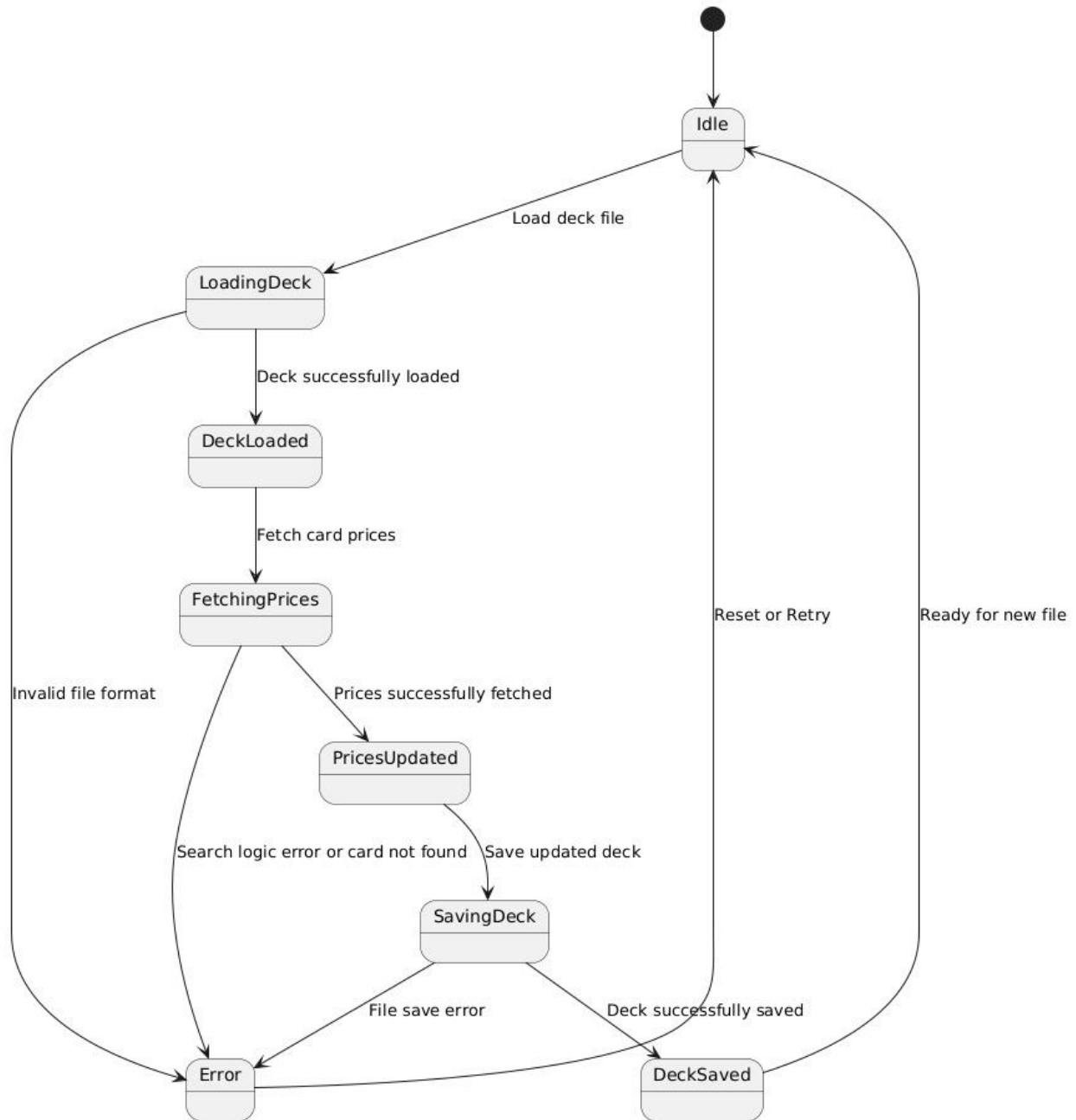
The Magic: The Gathering Price-checking Bot project demonstrates how even niche tasks like card price tracking can benefit from Robotic Process Automation. By combining UiPath’s straightforward activity-based workflows with Cardmarket’s readily accessible data, users gain a powerful, time-saving tool. From an operational standpoint, the bot significantly reduces manual data entry, improves accuracy, and can scale effortlessly to larger collections. Future expansions—API integrations, multi-currency support, or advanced analytics—promise to further streamline MTG collection management for a global audience.

## 15. References

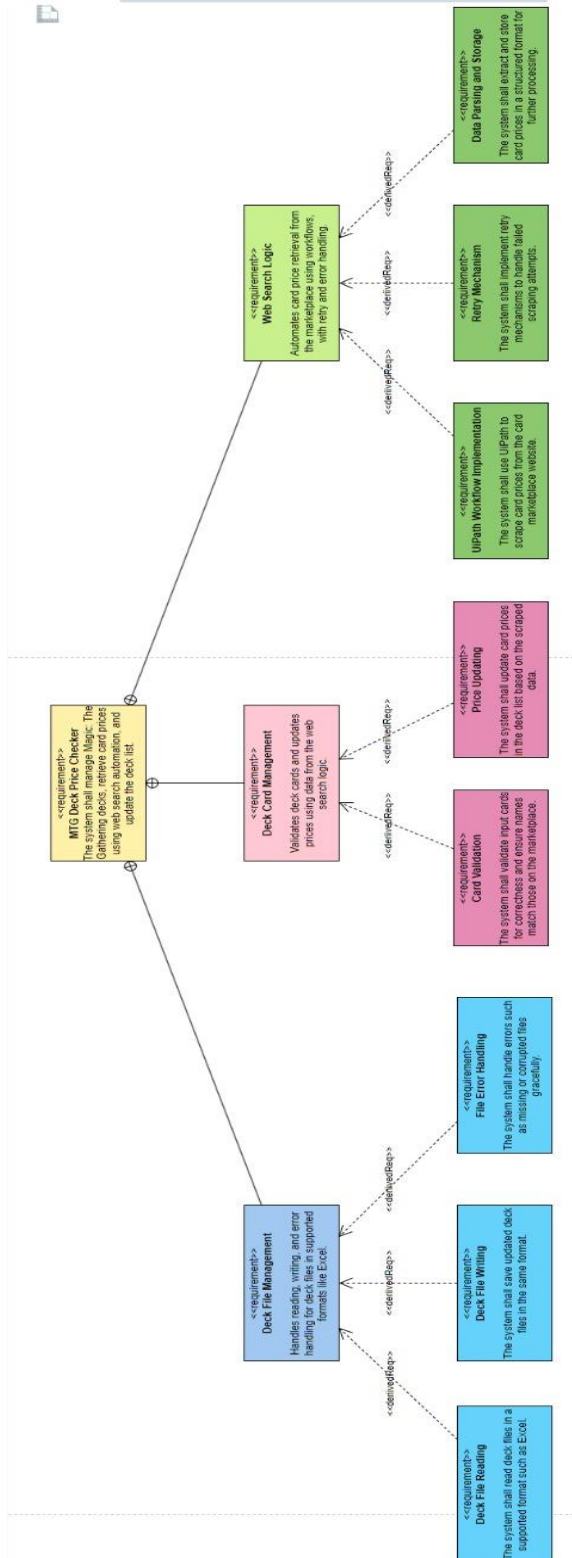
1. **Cardmarket.com** – <https://www.cardmarket.com>
2. **MTGGoldfish** – <https://www.mtggoldfish.com>
3. **UiPath** – <https://www.uipath.com>
4. **EchoMTG** – <https://www.echomtg.com>
5. **Deckbox** – <https://deckbox.org>

## 16. Annexes

### Annex 1: Workflow Diagram (State-Based System)



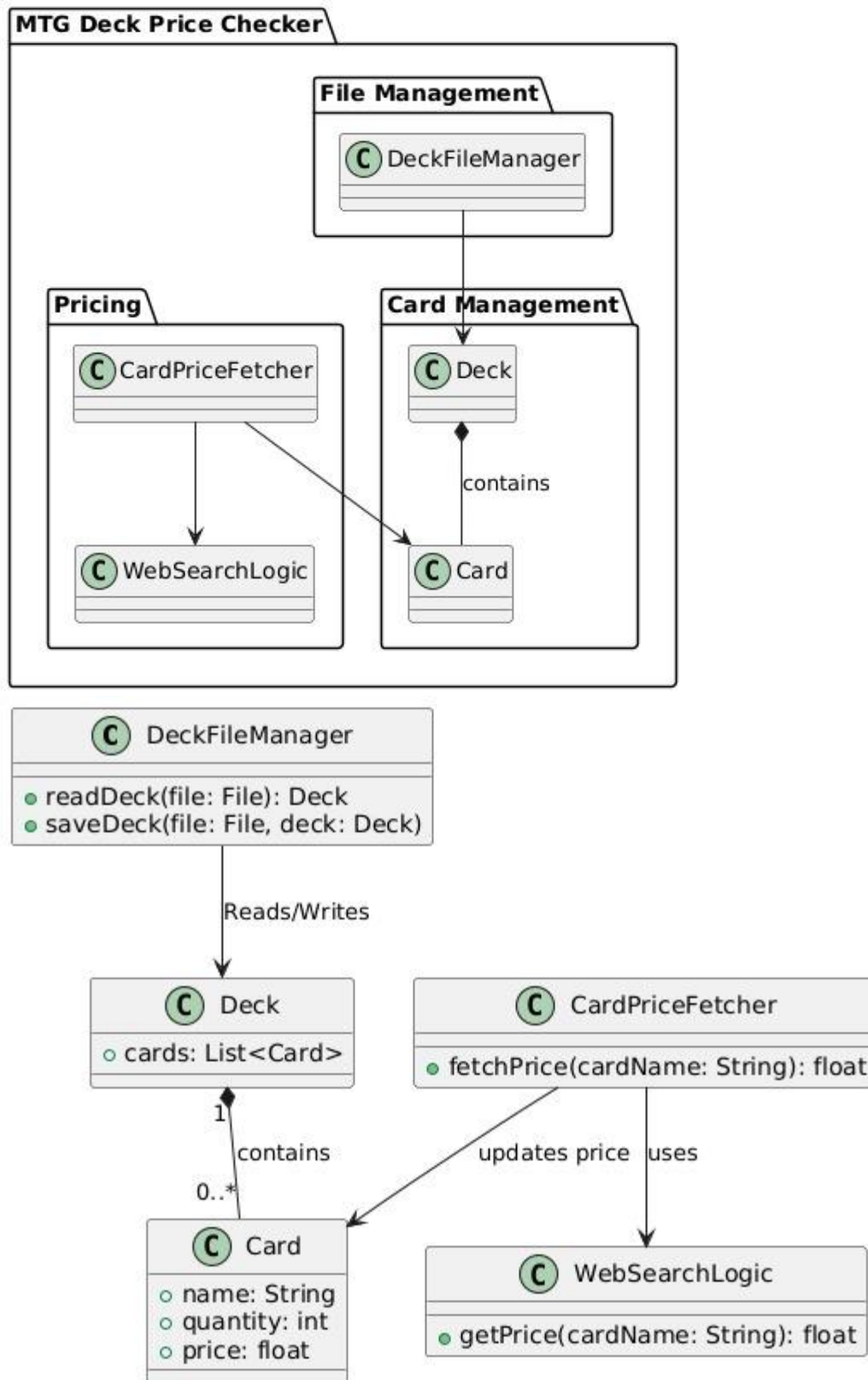
## Annex 2: Requirements Hierarchy



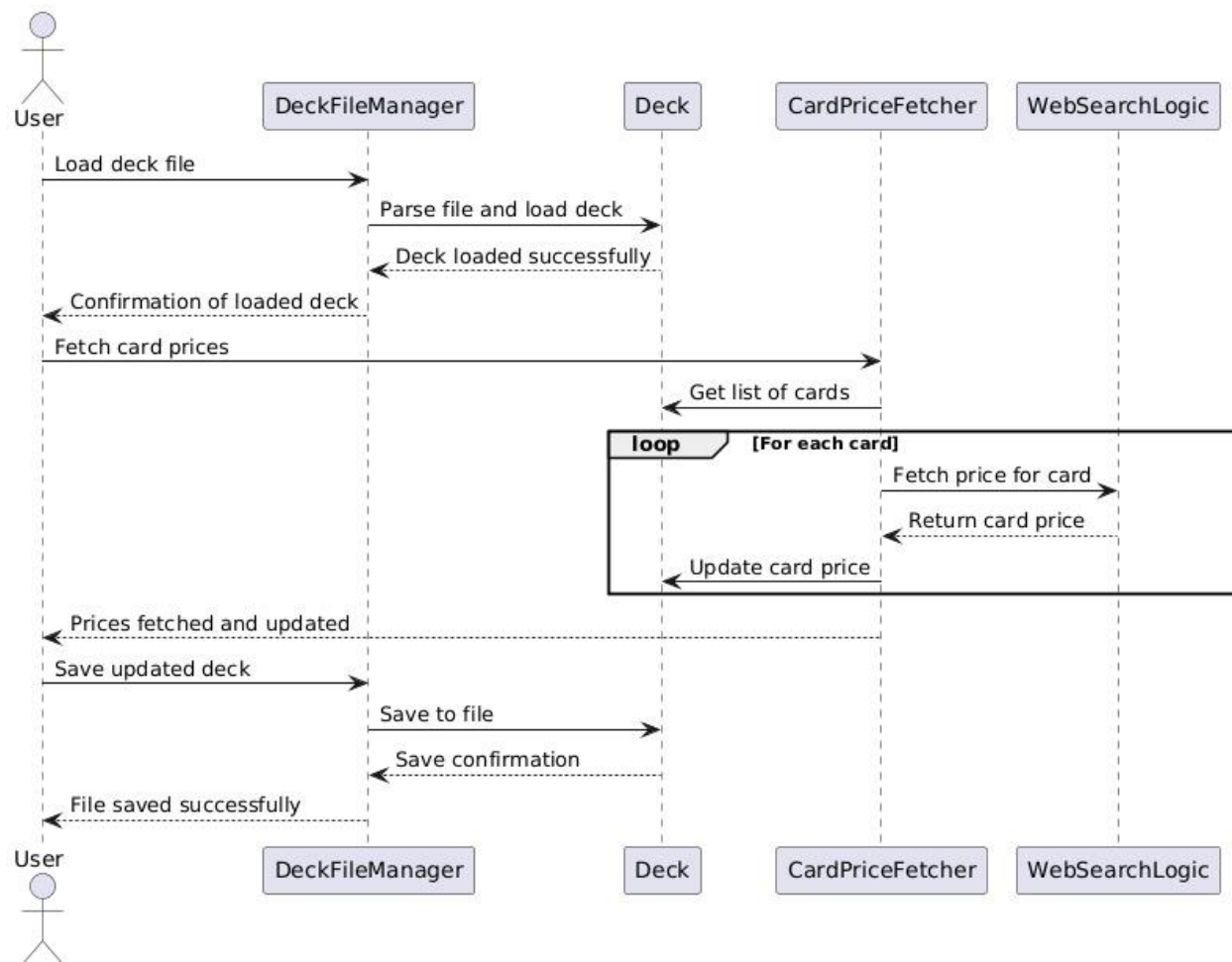
Annex 3: Risk Management Table

Date		Project Mngr		Name														
PROJECT RISK MANAGEMENT PLAN																		
Identification				Qualitative Analysis			Quantitative Analysis			Response Strategy		Monitoring and Control						
ID#	Status	Date Identified	Functional Assignment	Threat/Opportunity Event	Risk Trigger	Type	Probability	Impact	Risk Matrix	Probability (%)	Impact (8 or 13)	Effect (9 or 13)	Response Actions including advantages and disadvantages	Affected WBS Tasks	Responsibility (Task Manager)	Status Interval or Milestone Check	Comments	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)-(15)	(16)	(17)	(18)	(19)	(20)
1	Active	10/10/2024	Development Team	Web scraping requires additional time to implement due to dynamic website changes	Unanticipated changes in the card marketplace website structure or anti-scraping measures.	Schedule	High	High	<div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div><div>L</div><div>M</div><div>H</div><div>VH</div></div><div><div>VL</div>&lt;</div></div>									

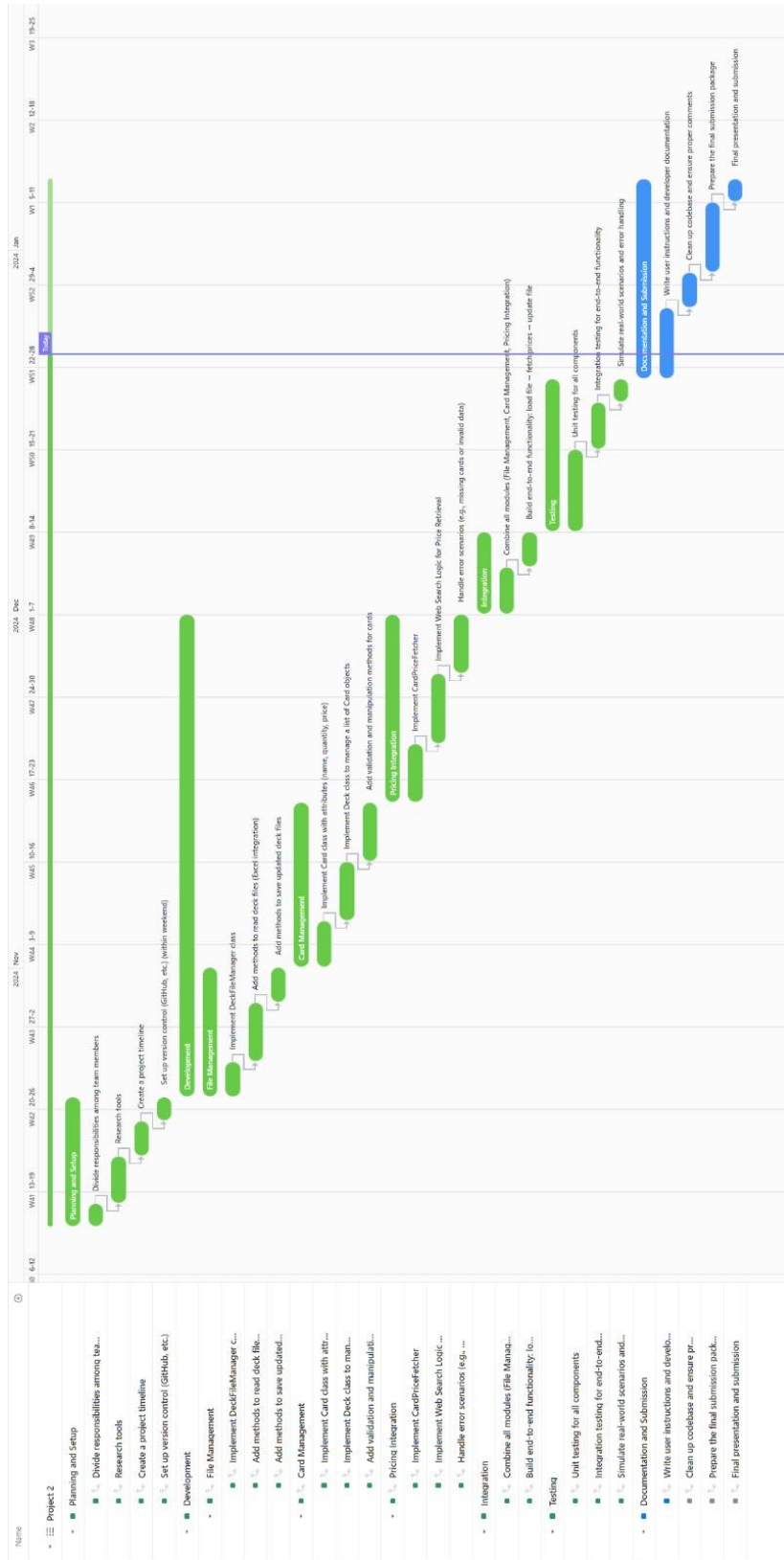
## Annex 4: Class Diagram



## Annex 5: Sequence Diagram

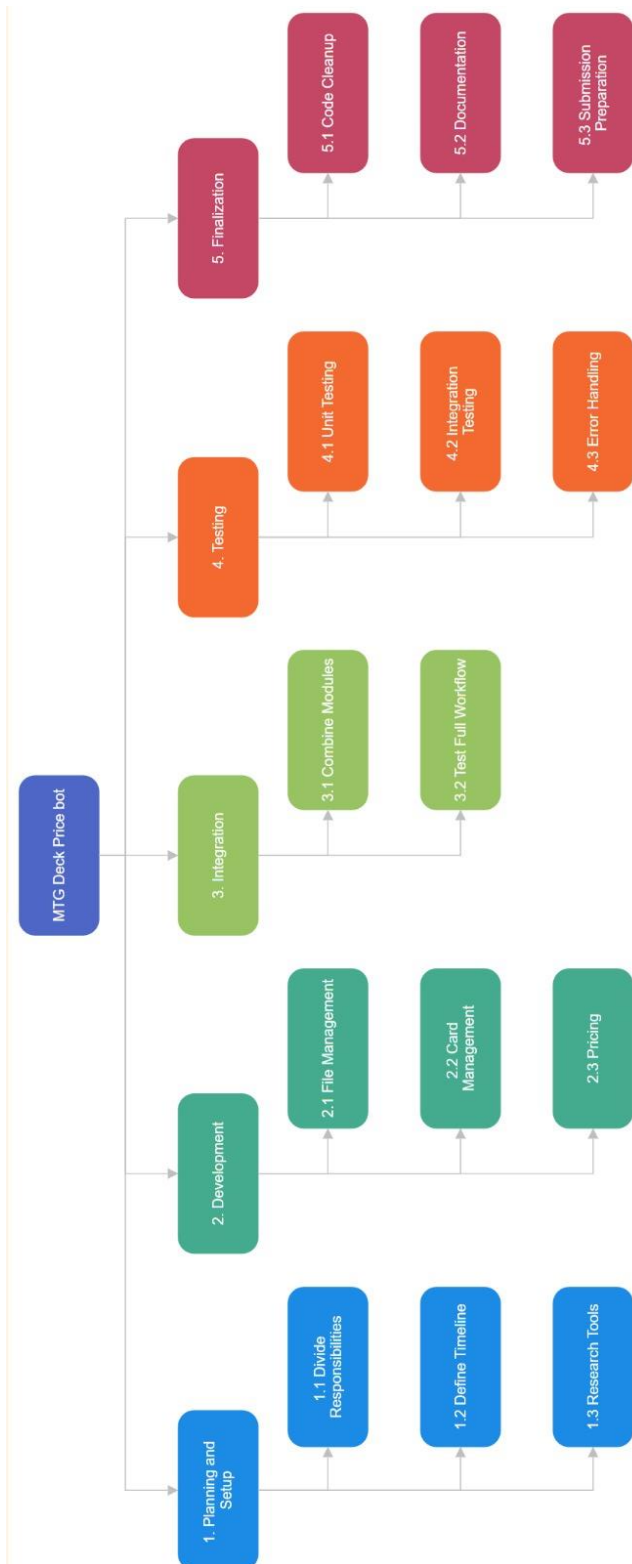


## Annex 6: Gantt Chart





## Annex 7: Development Process Flow



## Annex 8: Main sequence

(B) A bot that accesses an excel file with a Magic: the Gathering decklist and computes the price column for each individual card taking the price from cardmarket.com

