**Software Requirements Specification**

**for**

C\*A\*R\*D

(Collectibles Aggregation and Retail Determination)

**Version 1.0 approved**

**Prepared by Benjamin Klybor, Nick Tulio, Parker Fairchild, Jahkel Robin**

**CS-275**

**10/23/2017**

**Table of Contents**

**Table of Contents 1**

**Revision History 2**

**1. Introduction 3**

1.1 Purpose 3

1.2 Document Conventions 3

1.3 Intended Audience and Reading Suggestions 3

1.4 Product Scope 3

1.5 References 3

**2. Overall Description 4**

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Classes and Characteristics 4

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements 5**

3.1 User Interfaces 5

3.2 Hardware Interfaces 6

3.3 Software Interfaces 6

3.4 Communications Interfaces 7

**4. System Features 7**

4.1 System Feature 1: Two Step Verification 7

4.2 System Feature 2: Collection Management 8

4.3 System Feature 3: Keep Collection Up To Date 8

4.4 System Feature 4: Communication Between Users 9

**5. Other Nonfunctional Requirements 9**

5.1 Performance Requirements 9

5.2 Safety Requirements 10

5.3 Security Requirements 10

5.4 Software Quality Attributes 10

5.5 Business Rules 10

**6. Other Requirements 11**

**Appendix A: Glossary 11**

**Appendix B: Analysis Models 11**

**Appendix C: To Be Determined List 11**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

This document purpose is to provide a clear understanding of the Collectable Aggregation and Retail Determination system (hereafter referred to as the Software). We will explain the function of the system as well as the targeted audience and the how the Users will interact with the Software. The information contained in the document is intended to provide details about all the functions of the system, the interfaces and use cases. This document also includes goals that the team may or may not implement, depending on the constraints.

## Document Conventions

For the remainder of the document any words or phrases highlighted in yellow will describe a part or whole of a stretch goal for the project. A stretch goal is something that is not crucial to produce the project, but that is desirable and relevant to improve the Software.

## **Intended Audience and Reading Suggestions**

The intended audience for this document includes the developers, project manager, and any testers. The remainder of the document will provide an overall description of the Software followed by a description of the external interface requirements and system features. There are also a number of nonfunctional requirements descriptions as well.

## **Product Scope**

There are two main goals for this project. First is the academic goal of following the software design cycle in order to implement proper software design techniques and gain experience designing software. The second goal is to create a unique product that will outlive the completion of the project. We hope to complete both these goals in tandem, however the academic goal takes precedence over the creational one. While the creation of a novel product is the general end-goal of software design, since the project is meant for academic purposes, function will take precedence over form.

## **References**

Project Description - [*https://docs.google.com/document/d/131tBCgyBNPsnF9xqcMO0JNIddpupetWT9suek8amvTM/edit?usp=sharing*](https://docs.google.com/document/d/131tBCgyBNPsnF9xqcMO0JNIddpupetWT9suek8amvTM/edit?usp=sharing)

# **Overall Description**

## **Product Perspective**

This product is a new idea, not based on any pre-existing product or system. The product will rely on gathering data from the internet. Otherwise, all of its functionality is self-contained.

## **Product Functions**

* User must be able to sign in to their account
* Must run on Windows
* User must be able to enter and save a list of collectibles
* The system must update the price of the collectibles

## **User Classes and Characteristics**

This product is most focused on collectors. Typically, the most Users will most likely be “Adult Collectors”. Specifically, people of working age, which would be around 14 years or older, who regularly purchase collectible items. “Adult Collectors” will be the system’s core demographic. This is because this group will be the most likely to regularly purchase collectibles and check information about each collectible using the C\*A\*R\*D product. Another possible demographic would be “Young Collectors”. However, they seem to be less likely to use the app, as they may not be able to buy collectibles as often as an adult with the ability to freely spend their own money.

## **Operating Environment**

The system will be able to run on Windows 10.

## **Design and Implementation Constraints (Optional)**

This project must be completed by the end of the semester. This is a very heavy time constraint. The team has experience with C++ and SQL, so the project will most likely be restricted to these languages. Additionally, since this is a school project, the team does not have any budget. Therefore, the team cannot rely on any tool that must be purchased.

## **User Documentation**

The documentation will include a User manual. The User interface for the system should not be very overwhelming, so a simple manual should be able to effectively explain to the User how the system works. This manual might be accessed through a “Help” button on the User interface.

## **Assumptions and Dependencies**

The project is dependant on being able to use third party databases, such as gathering information online. It is assumed that these databases will not be difficult to access through code.

# **External Interface Requirements**

## **User Interfaces**

User will be prompted to login or create a new account with a username/email and password. A verification code will be sent to the User’s email when a new account is created and every time a login attempt is successful, the User will have to enter this code in order to access his account. Creation of a new account will navigate Users to enter a list of collectibles to get started and navigate existing Users to their existing collection. The main User interface will contain the following components: list of current collection, a “watch list” of items of interest, a refresh button to update the current value of the collection, a menu for viewing and changing the collection in detail, the Collection menu, and a menu for viewing and changing User account information, the Account menu.

The Account menu will allow the User to update information such as linked emails and account password. The Collection menu will allow the User to view and change his collection in greater detail than the main user interface. Stretch goals will be to implement an internal messaging system and a community/forum system so Users can communicate with other Users as a whole. Ideally, a server will keep track of the value of a User’s Collection over time and the Software operating on the User’s machine will be able to generate a chart showing the value of the User’s Collection over time.

## **Hardware Interfaces (Optional)**

Hardware of Runnable Machines must be able to run some version of Microsoft Windows Professional 7 or later.

## **Software Interfaces**

The Software will communicate with the standard Microsoft Windows kernel root system and the standard network socket system on the Runnable Machine. Will need to communicate with collectibles websites to update item validity and price. The *Refresh Button* will query pertinent websites for up-to-date price information and check an item’s validity and return said information to the server. Will need to communicate with a server that will store historical price information about registered items. The Software will communicate with various websites that store up-to-date information of the value of a given item. The Software will determine which data storage formats the item information is stored in on a given website, retrieve that information using whatever communication standards used by the given website, and then

## **Communications Interfaces**

The Software will use the standard TCP/IP model for sending and receiving messages between networked computers, collectibles websites, and server. Communication with third party databases, the websites containing information on collectibles, will be determined at run time as the websites queried, their data storage methods, and their security systems might change over time, so this part of the Software must be flexible.

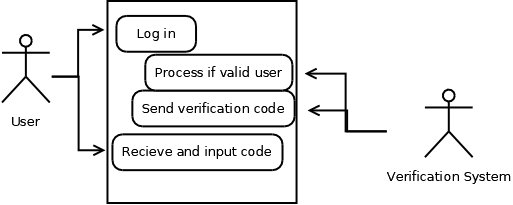
# **System Features**

## Two step verification

* + 1. Description and Priority

The two-step verification feature will allow for higher security for the User, in which the User will login and then be prompted by the Software to input a code sent via email from the Software. This feature is of medium-high priority.

* + 1. Stimulus/ Response Sequences

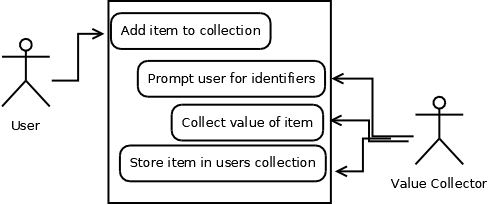
Use Case Diagram: 

* + 1. Functional Requirements
* The login screen will be hosted on the user’s machine
* An email will be sent from an address TBD that will contain a code the User needs to access his account; no access code will be sent if the User provides invalid login information
  1. **Collection Management**
     1. Description and Priority

This feature requires the User to input the name of an Item.Then depending on what the item is there will be further prompts (i.e for Magic the Gathering cards it will prompt for what type of collectible it is, what subset of the game it is, and then the actual name of the card). The Software will then use that info to collect the value of the item. The Collection Management will keep a list of the the User’s current Collection, including the name of each Item, a description of each Item, and the value of each Item. It will then compute the total value of the Collection and store this as well. This is a high-priority feature that will be implemented.

* + 1. Stimulus/Response Sequences

Use Case Diagram:

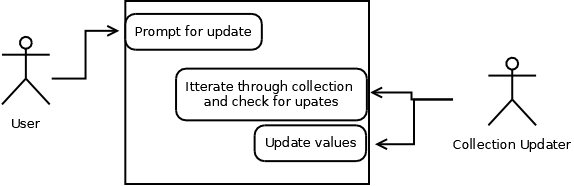


* + 1. Functional Requirements
* Will need to communicate with different third-party servers and, as a result, deal with different databases and different security requirements
* Invalid Items, defined as Item whose names cannot be verified by the Software, will be rejected by the Software and no value lookup will take place; rejected Items will still be stored in the User’s collection, but will not contribute to the value of the Collection
  1. **Keep collection up to date**
     1. Description and Priority

The Software will update the Collection’s values when prompted to by the use of an update button. This won’t require further input from the user past the button. This will be done with the observer pattern. This is a medium-high priority feature.

* + 1. Stimulus/Response Sequences

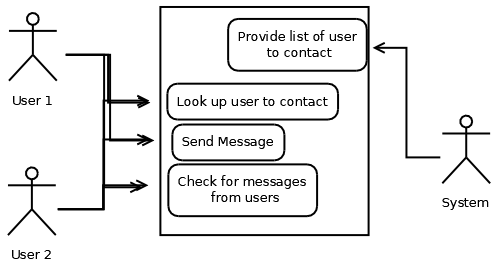
Use Case Diagram:

* + 1. Functional Requirements
* Will need to access the Collection, a list of some kind, stored locally on the User’s machine
* Will need to communicate with different third-party servers and, as a result, deal with different databases and different security requirements
* Invalid Items, defined as Item whose names cannot be verified by the Software, will be rejected by the Software and no value lookup will take place; rejected Items will still be stored in the User’s collection, but will not contribute to the value of the Collection
  1. **Communication between users**
     1. Description and Priority

There will be a rudimentary system for users to have discussions on. This will be for things relating to trading collectibles between their collections and will serve solely as a way for users to set up doing so, it will not have a means for them to exchange items out of the system. This is a low priority feature.

* + 1. Stimulus/Response Sequences

Use Case Diagram:

* + 1. Functional Requirements
* Keep a current list of the IP addresses of each User
* Be able to efficiently use the sockets and ports of the Runnable Machine

# **Other Nonfunctional Requirements**

## **Performance Requirements**

The Software will need to be able to communicate with various third-party servers and pull information from these servers upon request from the user. Calculation of the value of the User’s Collection will need to be done in a timely manner without any other action on behalf of the User beyond hitting the refresh button.

## **Safety Requirements**

No outstanding safety hazards are apparent at this time.

## **Security Requirements**

The two step verification system will be the highest form of security we will offer to users. Other than that the Software will use default security features of whatever external softwares are used.

## **Software Quality Attributes**

The initial Software will most likely require the user to have a decent degree of technical knowledge about each item in his Collection. Over time we hope to develop the Software to make it more intuitive, but a working program, over a beautiful program, is the end goal as of now. For now the Software will be restricted to run on Microsoft Windows.

## **Business Rules**

The team will meet as a group every Tuesday from 3-4 p.m. and every Wednesday from 6-7 p.m. Each goal set out for each individual team member must be completed by the time of the next team meeting. Goals set at the Wednesday meetings will be large enough to be completed by the Monday of the following week, and goals set at Tuesday meetings will be simple enough to be completed by the next day’s meeting.

# **Other Requirements**

**Appendix A: Glossary**

|  |  |
| --- | --- |
| Term | Meaning |
| User | The end-user of the Software. Includes anyone that will use the Software for either personal or academic needs. |
| The Software (Software) | Any software developed to implement the project |
| Runnable Machines | The types of machines that will be able to run the Software |
| Highlighted Sentences/Words | Highlighted words and phrases are stretch goals that are not necessary for completing the project, but that are desirable to implement. |
| Collection | The list of items the User submits to the Software. |
| Item | The name of the real world object the User has in his collection. |

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

* The email service that will be used to send the two-step verification code
* The type of data storage system to be used to store Collection information on the User’s machine