[**1. Summary**](#_o0wuneb5ph1c) **2**

[1.1. Project Overview](#_wsxz56m9aib9) 2

[1.2. Intended Audience](#_yoi5gfcnbd41) 2

[1.3. Team Members](#_eko7xsak1m1d) 2

[**2. Product Description**](#_vptaoxg1oadz) **2**

[2.1. Product Context](#_7d99e4zh4tg4) 2

[2.2. Project Timeline](#_oh2o0dopnnzd) 2

[2.3. Dependencies](#_mylfgi6wuk98) 3

[2.4. Similar Software](#_484j5yoxe8yd) 3

[**3. Top-Level Requirements**](#_t6p0vg4jke5) **4**

[3.1. Import/Export Speed](#_i7q1si7rvkvk) 4

[3.2. Library Capacity](#_4spgky3qmnx9) 4

[3.3. Implementation](#_k3z4ubmudgh7) 4

[3.3.1. File Format](#_ba6zyvbu9ne9) 4

[3.4. Security](#_sdimbcbzqzkw) 4

([sample project requirements template](https://csis.pace.edu/~marchese/CS775/Requirements%20Specification%20Template.doc))

([another sample requirements doc](https://www.hudexchange.info/resources/documents/FunctionalandTechnicalRequirementsTemplate.doc))

# 1. Summary

## 1.1. Project Overview

Virtual Game Shelf is a desktop Java application to organize your game library in a single convenient location. It strives to implement the best features from services like [Backloggery](http://www.backloggery.com/) and [My Game Collection](http://my-game-collection.tuyware.com/), without the need of an internet connection.

## 1.2. Intended Audience

Virtual Game Shelf is being designed for gamers with large game collections spanning multiple consoles. In some cases, these libraries span [1,000’s of games](http://www.backloggery.com/Stevoisiak) [or more](http://backloggery.com/ProtonJon).

## 1.3. Team Members

Steven will serve as project manager. He will host the projects public GitHub repository, reviewing and giving feedback on pull request changes made to the project. He will also be in charge of UI design and documentation. Documentation includes UI sketches, variable lists, file formats, and text documentation.

Morgan will be handling backend API functionality. He will research program API’s for Steam, Backloggery, Xbox Live, and PSN to determine the feasibility of each platform.

Arielle will be handling GUI implementation and programming. She will be the main team member working on user interface design. She will handle interactions between the main program GUI and Morgan’s backend API functionality.

# 2. Product Description

## 2.1. Product Context

Virtual Game Shelf is primarily designed to partially automate the task of game organizer site Backloggery.

## 2.2. Project Timeline

**Monday, February 27:** Groundwork

* File format for saved library decided on
* Basic functionality for Steam API integration
* Research on Backloggery integration

**Monday, March 6:** Milestone 1 (Functional Prototype)

* Can import Steam game libraries
* Allows manually adding/removing games
* Can save library to file
* User can view list of games in library
* If feasible, initial Backloggery integration
  + Morgan will finish researching whether Backloggery integration is feasible
* Basic user interface

**Monday, April 10:** Milestone 2 (Advanced Prototype)

* Can load library from file
* Library searchable by game title & console
* Exports to Backloggery
* Tracks game progress
* Polished user interface

**Friday, May 5:** Final polish/Presentation prep

**Friday, May 12:** Final Deliverable

* Full steam integration
  + Successfully imports [600+ game libraries](http://steamcommunity.com/id/Stevoisiak/games/?tab=all)
  + Determines unplayed/unfinished status based on time played

## 2.3. Dependencies

The GUI will require the installation [JavaFX for Eclipse](https://www.eclipse.org/efxclipse/install.html) for development and compilation.

Some [Steam Web API](https://partner.steamgames.com/documentation/webapi) calls may require the use of a developer API key.

## 2.4. Similar Software

[Backloggery](http://www.backloggery.com/): Popular online service for manually tracking game collection.

[bltool](https://github.com/ToxicFrog/bltool): Open source tool for command line interaction with Backloggery. Basic export functionality from Steam. Can possibly be implemented into our own project.

# 3. Top-Level Requirements

## 3.1. Import/Export Speed

Performance should be comparable to similar programs. [Bltool](https://github.com/ToxicFrog/bltool) is able to export a Steam library of [652 games](http://steamcommunity.com/id/Stevoisiak/games/?tab=all) to a [formatted text file](http://pastebin.com/00aRXCsj) in 10 seconds. Our program should aim for 30-60 seconds at most to import the same library.

## 3.2. Library Capacity

One of the largest known Steam libraries belongs to user [A\_dJ](http://steamcommunity.com/id/A_dJ) who, as of writing, owns over 10,000 games. To ensure support for all Steam users, our Steam importer must support libraries of at least this size.

## 3.3. Implementation

Virtual Game Shelf will be programmed primarily in Java using the [Eclipse](https://eclipse.org/) compiler. It will be hosted as a public GitHub repository. It will use the JavaFX library for its user interface.

### 3.3.1. File Format

File Format Examples:

* bltool ([sample](http://pastebin.com/00aRXCsj))
  + Exports as .txt
  + <region>, <platform>, <completion status>, <title>
* [steamcsv](https://github.com/jazzmind/steamcsv) ([sample](http://pastebin.com/MKasxda5))
  + Exports as .csv (comma separated variables)

## 3.4. Security

Private user information such as passwords must NEVER be stored without the user’s knowledge. In general, our program should try to avoid requesting passwords unless necessary. When possible, use official API’s (i.e., [Steam Web API](https://partner.steamgames.com/documentation/webapi)).

Whenever possible, allow users to retrieve publicly available account information without needing to log in or provide account passwords.

Users should not need to disable security features such as Two-Factor Verification when linking accounts with Virtual Game Shelf.