**Software Engineering**

**Software Requirements Specification**

**(SRS) Document**

**Game & Go**

**September 17th, 2022**

**v1.0**

**By: Justin Kim**

**Theodore Okamura**

**Nicholas Ray**

**<Honor Code>**

|  |
| --- |
| **Table of Contents** |

[1. Introduction 2](#_Toc113453891)

[2. General Description 3](#_Toc113453892)

[3. Functional Requirements 3](#_Toc113453893)

[4. Technical Requirements 4](#_Toc113453894)

[4.1 Operating System & Compatibility 4](#_Toc113453895)

[4.2 Interface requirements 4](#_Toc113453896)

[4.2.1 User Interfaces 4](#_Toc113453897)

[4.2.2 Hardware Interfaces 4](#_Toc113453898)

[4.2.3 Communications Interfaces 4](#_Toc113453899)

[4.2.4 Software Interfaces 4](#_Toc113453900)

[5. Non-Functional Requirements 4](#_Toc113453901)

1. Introduction
   1. **Purpose:** Our goal is to have a one stop shop for searching for new games, reading reviews and downloading games.
   2. **Document conventions:** The purpose of this Software Requirements Document is to give a comprehensive overlook at our project and the goals we wish to accomplish. We want our users to be greeted with a simple, yet effective front page that will be able to immediately narrow down what they are looking for. We have all be in a situation where we are presented with so many choices it becomes overwhelming, that is where we step in to alleviate that problem. When you first log-in you will be prompted to select which system you are shopping for, whether that be consoles, computer, or other devices. Then we allow the users to pick their genre of choice they wish to browse. Once these simple requests are input, we will pull from an API of assorted games and pick those which match the description they user has given.
   3. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition. Acronym, Abbreviation** |
| MongoDB | NoSQL database program |
| Spring boot | Java framework to create java based applications |
| DB | An abbreviation for Database. |

* 1. **Intended audience:** This project is the brain child of our very own, Justin, Theo and Nick. This Software Requirements Document will be useful for future users and used to show possible investors what we bring to the table and what we plan to deliver.
  2. **Project Scope:** Our overall goal is to allow people who are interested in new games a very simple shopping experience. Through this our business would be an online distribution of games from large companies possibly including, but not limited to: Activision, Blizzard, Bungie, and 343.
  3. **Technology Challenges:** Throughout development we have encountered little to no technological difficulties.
  4. **References:**  As of this moment we have no references since we are in infancy of our project.

## General Description

* 1. **Product perspective:** This project came to use when we all were able to relate and share less than ideal experiences with window shopping for games. The problem of having to go to multiple websites to find different pieces of information about a single triple A title can be downright frustrating, and that’s where we want to step in and be the solution to that problem.
  2. **Product features:** This program will be able to sort through an list of games sources from an API by genre, console, and hardware specifications. This will lead to a concise list of games that the user may enjoy purchasing.
  3. **User class and characteristics:** Our website application relies on the assumption that the user has a vague idea of what they are looking for, such as the console they wish to play it on whether it be Xbox, PlayStation, or on a Computer. Along side that we also give them the ability to input their preferred genre, ranging from first person shooters, massive online multiplayer, or horror! There will be no need to go to a multitude of different website that must individually perform these tasks.
  4. **Operating environment:**

Server: Java 17

Client: Web browser

* 1. **Constraints:** We are limited to Java and Spring boot.
  2. **Assumptions and dependencies:** Our external dependencies are SteamDB API, MongoDB, Spring boot as of now

## Functional Requirements

Functional requirements

Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.

* 1. **Primary**

We don’t have anything at this point in time.

* 1. **Secondary:**

We don’t have anything at this point in time

## Technical Requirements

* 1. Operating System & Compatibility
  2. Interface requirements
     1. User Interfaces

Not applicable at this time.

* + 1. Hardware Interfaces

Not applicable at this time.

* + 1. Communications Interfaces

Not applicable at this time.

* + 1. Software Interfaces

Not applicable at this time.

## Non-Functional Requirements

Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.

* 1. **Performance requirements**

Not applicable at this time.

* 1. **Safety requirements**

Not applicable at this time.

* 1. **Security requirements**

Not applicable at this time

* 1. **Software quality attributes**
     1. Availability

Software must be acceptable to the type of users for which it is

designed.

* + 1. Correctness

Software should run and execute the correct commands at any given time.

* + 1. Maintainability

Software should be written in such a way so that it can evolve to

meet the changing needs of customers.

* + 1. Portability

A set of attributes that bear on the ability of software to be transferred from one environment to another.

Detailing on the additional qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability etc.

* 1. **Process Requirements**
     1. Development Process Used

Plan Driven Process-Evolutionary Prototyping

* + 1. Time Constraints

This semester.

* + 1. Delivery Date

Sept. 20- SR Document Due

Oct. 4 Design Document Due

* 1. **Other requirements**
* All work subject to UNCG Academic Integrity Policy.

All SRS/SRD should be:

* **Correct:** A method of analysis that ensures that the software meets the requirements identified.
* **Unambiguous:** There is only one interpretation of what the software will be used for and it is communicated in a common language.
* **Complete:** There is a representation for all requirements for functionality, performance, design constraints, attributes, or external interfaces.
* **Consistent:** Must be in agreement with other documentation, including a systems requirements specification and other documents.
* **Ranked for Importance and/or Stability:** Since all requirements are not of equal weight, you should employ a method to appropriately rank requirements.
* **Verifiable:** Use measurable elements and defined terminology to avoid ambiguity.
* **Modifiable:** A well-defined organizational structure of the SRS document that avoids redundancies can allow easy adaptation.
* **Traceable:** Ability to trace back to the origin of development and move forward to the documents produced from the SRS.
* **Legible and Professionally Presented**: Must use a consistent font and style. Must have proper formatting of tables and charts. Must be grammatically correct. Use active tense and concise sentences.