Level 2 - Kyle de Vos (KD23040008523)

L2T13 – Capstone Project I

Date: 5 July 2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Software Requirements Document**

**Bookstore Database Management System for Book Clerk for Stock Control**

Table of Contents

1. **Preface**
   1. Version History
   2. Intended Audience
   3. Versions Overview
2. **Project Introduction**
   1. Purpose
   2. Function and Business Integration
   3. Project Scope

2.4 References

1. **Glossary**
2. **User Requirements Definition**
   1. Functional Requirements
   2. Non-Functional Requirements
   3. Product Standards
3. **System Requirements Specification**
   1. Function Requirements
   2. Non-Functional Requirements
4. **System Architecture**

6.1 Architecture Model Overview

1. **Overall Description**
   1. Operating Environment
   2. Classes and Interfaces
2. **Testing**

8.1 Unit Tests

1. **System Evolution**
   1. Hardware Evolution
   2. Anticipated Feature Modifications
   3. Anticipated Feature Additions
   4. Security Additions
   5. Interface Additions
2. **Appendices**
3. **Index**
4. **Preface**
   1. Version History

This is the first designed system of the *Book Management System* with no prior system in place. Version History is none.

* 1. Intended Audience

This document and its accompanying project form a Capstone Project simulating a *Book Store Management System* for the HyperionDev Software Engineering Course. It is not intended for real-world use. The audience of this project are the HyperionDev markers and reviewers. Docstrings for Modules, Classes and Methods are worded to describe thought processes or design styles in addition to function.

* 1. Versions Overview

stock\_manager.py – First Version. Includes full CRUD operations and functionality to interact with database, retrieve and validate user input and return data to UI. Functions allowing user to use the ‘Main Menu’ are included here with setup of the console UI. This version includes the modules and classes required for the above.

1. **Project Introduction**
   1. Purpose

The simulated purpose of this project is a Book Stock Management System for a bookstore with an included database. This project aims to allow access to this database and perform CRUD operations on its entities, utilising a terminal as its UI. Finance Tracking Functionality has not been incorporated. The project has elements of dependency de-coupling and the SOLID Principles to maximise re-use and scalability, but is designed to meet requirements of task for a table handling only ‘book’ elements.

* 1. Function and Business Integration

This project is simulated (and assumed) to assist a business with no prior software-based system. It is assumed that the business has traditional ‘paper’ systems in place. Requirements Elicitation did not show evidence of a current system.

This project aims to replace a potential ‘paper’ stock management system allowing for simple CRUD operations only. The simulated business would be intended to perform financial operations utilising data from this project (more specifically stored within its database).

* 1. Project Scope

This project will look to manage Book Stock for a simulated Book Store Business. A database is included and will allow a user to perform addition, manipulation and deletion of Books stored in the database. Additional classes and interfaces will be added to ensure this project is scalable and follows the SOLID Principle [1].

* 1. References

[0] – Software Requirements Document compared to style from KrazyTech.

Ravi Bandakkananavr, Software Requirements Specification document with example. Last Accessed on 4 July 2023.

Available from: <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

[1] – Stephen Watts. The Importance of SOLID Design Principles.

Last Accessed on 4 July 2023

Available from: <https://www.bmc.com/blogs/solid-design-principles/#:~:text=SOLID%20is%20an%20acronym%20that,some%20important%20benefits%20for%20developers>.

1. **Glossary**

* SOLID Principle - A set of 5 design principles used by developers to ensure a program is designed in an agile manner – allowing for easy modifications and scalability. [1]
* UI – User-Interface – Manner in which the application is presented to the user.

1. **User-Requirements Definition**

4.1 Functional Requirements

* Main Menu incorporated into terminal interface to allow user to select from options
* Add new books to the database
* Update (change) details for an existing book stored in database
* Delete a book from the database
* Search for a specified book within the database – assumed to allow search for book by title and/or author
* Menu Option to exit application

4.2 Non-Functional Requirements

* **Usability** – Main Menu is concise with clear menu options. Information returned from the system should be readable
* **Reliability** – System should validate inputs to ensure data written to database or accepted from user is correct (and in correct format). Program must be tested thoroughly ensuring invalid input does not crash program.
* **Performance** – Program should aim to minimise delays in data retrieval and processing of large amounts of information.
* **Security** – Current Project does not include requirements to authenticate user, but should be considered and recommended for future versions. Data must be checked for being entered into the database.
* **Implementation Constraints**: This project is a simulation business application that is created only using Python and SQLite.

4.3 Product Standards

This Project is designed to work with Books recording the attributes id, title, author and quantity. Data for each of these values is required for the database and books in stock must conform to these attributes at minimum. Additional attributes (such as an ISBN number or year of publication and version) are not included and would need to be added in later versions of this system – noting that data corrections for books present in database would be needed.

**5. System Requirements and Specification**

5.1 Functional Requirements