**University Of Maryland Global Campus**

**Library Management System (LMS)**

**Project Plan**

**CMSC 495 7380**

**5 April 2023**

**Version1.10**

**Group 3: Brandon Durham, Benjamin Kus, Benjamin Ramos**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **REVISION** | **DATE** | **DESCRIPTION** | **Author** |
| 1.0 | 3/23/2023 | Initial | Ben Ramos |
| 1.1 | 3/26/2023 | Modification of Functional Requirements | Ben Kus |
| 1.2 | 3/28/2023 | Modification to the Specification Requirements | Brandon Durham |
| 1.3 | 3/28/2023 | Edit Overview and Objectives, Modification to Appendices, Addition of Test Schedule | Ben Ramos  Brandon Durham  Ben Kus |
| 1.4 | 4/2/2023 | Removal of Requirements Table | Brandon Durham |
| 1.5 | 4/2/2023 | Modification to the Software and Hardware Requirement | Ben Ramos |
| 1.6 | 4/2/2023 | Addition of Project Schedule | Ben Kus |
| 1.7 | 4/2/2023 | Addition of Software Management Plan | Brandon Durham |
| 1.8 | 4/4/2023 | Document Overview and Objectives Revision | Ben Ramos |
| 1.9 | 4/5/2023 | Software Management Addition | Brandon Durham |
| 1.10 | 4/5/2023 | Project Schedule Revisions | Ben Kus |

TABLE OF CONTENTS

1. Requirement Specifications
2. Objectives
3. Application Requirements
   1. Hardware
   2. Software
4. Software Management
5. Project Schedule
6. Appendix
7. **Requirement Specifications**
   1. A library management system is a project which aims in developing an application to maintain the book catalog of a library. The purpose of this is to provide administrators with the ability to conduct daily maintenance to its book catalog. It will provide an easy and administrator-friendly way to track books, add and remove users, as well as track overdue books and numerous other functions. It will also allow an administrator to see a list of books issued along with their issue date and return date. This system will store all the books and user information along with book numbers, book titles, and author names in the system database. The administrators can handle admin functions such as creating a new LMS user account and other administrative matters. Overall, this project is being developed to help the administrators with cataloging in the best possible way possible and reduce human strain.
8. **Objectives**
   1. By using a library management system, the operation of borrowing and managing inventories is paperless. This system provides an administrator-friendly data entry with a dropdown button menu, list box, and checkbox in purpose to make the input entry easier to understand and use. It is also created to ensure that the library items are stored properly to maintain their security. This system will store all the books and members' information that consist of book numbers, book titles, author names, and racks in the system database. Administrator and will handle administrative functions such as creating a new LMS user account and deciding the number of days allowed for the borrowed books. Administrators need to enter the correct password and admin ID before they can access this function. From here, they can add, delete, or update the book and user database.
9. **Application Requirements**
   1. Hardware Requirements
      1. These are the initial software requirements that and subject to change as the application is developed.
      2. Developmental Requirements
         1. Intel Core 2 Duo 1.8 GHz,
         2. 2 Gigabytes memory
         3. 150 GB hard disk
         4. Windows 10
      3. Operational Hardware Requirements
         1. Intel Core 2 Duo 1.8 GHz,
         2. 2 Gigabytes memory
         3. 150 GB hard disk
         4. Windows 10
   2. Software Requirements
      1. These are the initial software requirements that and subject to change as the application is developed.
      2. Developmental Software Requirements
         1. Windows 10/Linux OS
         2. Eclipse
         3. Java JRE 7
         4. Front-end frameworks like Angular/React/Vue for the client.
         5. Relational DBMS like MySQL, PostgreSQL, etc.
         6. Containers and orchestration services like Kubernetes (for a large setting like a national library)
      3. Operational Software Requirements
         1. Windows 10/Linux OS
         2. Relational DBMS like MySQL, PostgreSQL, etc.
10. **Software Management**
    1. Our group will be using GitHub as a software management system. We will use numerical system with a +1-version number i.e., 1,2,3,4, etc. to reflect the different versions of the application.
11. **Project Schedule**

**Test Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Duration** | **Start Date** | **End Date** | **Person** |
| **1. Project Requirements and Project Plan** | 7 | 3/22 | 3/28 |  |
| 1.1 Writing | 4 | 3/22 | 3/25 | Ben Ramos |
| 1.2 Self Review | 2 | 3/26 | 3/27 | Ben K. Ben R., Brandon |
| 1.3 Revision | 1 | 3/28 | 3/28 | Ben R. |
| **2. Project Analysis** | 7 | 3/29 | 4/4 |  |
| 2.1 Analyzing | 4 | 3/22 | 3/25 | Ben K. |
| 2.2 Self Review | 2 | 3/26 | 3/27 | Ben K. Ben R., Brandon |
| 2.3 Revision | 1 | 3/28 | 3/28 | Ben K. |
| **3. Project Design** | 7 | 4/5 | 4/12 |  |
| 3.1 Class Diagram | 4 | 4/5 | 4/8 | Ben R. |
| 3.2 Sequence Diagrams | 4 | 4/5 | 4/8 | Brandon |
| 3.3 Self Review | 2 | 4/9 | 4/10 | Ben K. Ben R., Brandon |
| 3.4 Revision | 1 | 4/11 | 4/11 | Brandon |
| **4. Project Test Plan & ICD** | 7 | 4/12 | 4/18 |  |
| 4.1 Writing Test Plan | 4 | 4/12 | 4/15 | Ben R. |
| 4.2 ICD | 4 | 4/12 | 4/15 | Brandon |
| 3.3 Self Review | 2 | 4/16 | 4/17 | Ben K. Ben R., Brandon |
| 3.4 Revision | 1 | 4/18 | 4/18 | Ben R. |
| **5. Sprint 1** | 7 | 4/19 | 4/25 |  |
| 5.1 Application Development | 4 | 4/19 | 4/22 | Ben K. |
| 5.2 Database Development | 4 | 4/19 | 4/22 | Brandon |
| 5.3 Self Review | 2 | 4/23 | 4/24 | Ben K. Ben R., Brandon |
| 5.4 Revision | 1 | 4/25 | 4/25 | Ben K. |
| **6. Sprint 2** | 7 | 4/26 | 5/2 |  |
| 6.1 Application Testing | 4 | 4/26 | 4/29 | Ben R. |
| 6.2 Application Revisions and Clean Up | 4 | 4/26 | 4/29 | Ben K. |
| 6.3 Self Review | 2 | 4/30 | 5/1 | Ben K. Ben R., Brandon |
| 6.4 Revision | 1 | 5/2 | 5/2 | Brandon |
| **7. Final Delivery** | 7 | 5/3 | 5/9 |  |
| 7.1 Final Modifications and Testing | 4 | 5/3 | 5/6 | Ben K. |
| 7.2 Application/GUI Verification | 4 | 5/3 | 5/6 | Brandon |
| 7.3 Self Review | 2 | 5/7 | 5/8 | Ben K. Ben R., Brandon |
| 7.4 Revision/Finalize | 1 | 5/9 | 5/9 | Ben R. |

1. **APPENDIX A**

**Group Members**

|  |  |
| --- | --- |
| Member | Ben Ramos |
| Member | Benjamin Kus |
| Member | Brandon Durham |