Library Application

Kekoa Kubli

CST-451 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Donna Jackson

Revision: 1.0

Date: September 3, 2023

**ABSTRACT**

The Books application is a library book management application. Using basic CRUD operations Users will be able to manage books that are available within a library. This application will be to allow users to add books to the libraries inventory, get a list of books based on different search criteria, and remove books from the inventory when books are checked out. The goal of this application is to help libraries maintain its inventory and to allow users to see what book are available.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| August 2, 2023 | Kekoa Kubli | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 4

Detailed High-Level Solution Design 5

Detailed Technical Design 6

Appendix A – Technical Issue and Risk Log 7

Appendix B – References 8

Appendix C – External Resources 9

Design Introduction

**High-Level Design:**  
A diagram of a cloud

Description automatically generated

* The purpose of this flowchart is to display all the components of the application and how they are logically connected with one another. Starting from the users running the application from a browser. Non-functional requirement that is out of scope would have some sort of login for user authorization to the application. Once user have access to the application the GUI is used to interact with all data. Then to the HTTP Calls that are used to Create, Read, Update, and Delete information between users and Databases.
* Implementation of this design and proposed solution will remain in line with the tools already selected from previous planned stages. Other Non-functional requirements that would get implemented would be authorization and security concepts to protect the user’s data, such as implementing a user login. Performance can be enhanced on the fly with cloud solutions and scaling up resources on demand as needed.

**Mockup Screen Shots:**

*Search Page*

A screenshot of a computer

Description automatically generated with low confidence

*Add New Book Form:*A screenshot of a computer

Description automatically generated with medium confidence

**Logical UML Diagram:**A picture containing text, diagram, plan, line

Description automatically generated

**Database ER Diagram:**

A screenshot of a book

Description automatically generated with low confidence

* Books are the main data Object in this application. These book object have different fields that can be used to cater to different search criteria. More field can be added in the future to aid in further filtering out search criteria. In the Relational Database the id is the Primary key that is unique to each book object and all other data fields are required.

1. Use the template to list the project deliverables that are to be included external to this Design Specification (Data Dictionary, API Design, etc.).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | API – Node.js/Express | Middleware created to handle API calls from the controllers to the database. | internal | Complete | 8/5/2023 |
| 2 | Frontend Configuration – React.js | Free Open-source frontend design for quick web page response times. | internal | Pending | 8/5/2023 |
| 3 | Cloud Deployment | Application deployment and Database deployment | internal | Pending | 8/5/2023 |

Detailed High-Level Solution Design

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1. Cloud based Applications | Using cloud technology can increase the security and accessibility of data. | Costs are scalable and testing can be done in a free environment. |
| 2 - |  |  |
| Hardware and Software Technologies | | |
| 1 – React.JS Frontend Framework | | |
| 2 – Azure/AWS Cloud Application and database deployment | | |
| 3 - | | |
| 4 - | | |
| 5 - | | |

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| R | Application Delpoyment | This is out of the scope of this project. This can continue to affect the cost of the application as the project scales. | Continual testing with deployment and testing with different cloud services. | Kekoa Kubli | Low | 8/5/2023 |  |  |
|  |  |  |  |  |  |  |  |  |

]

Appendix B – References

*List all Project Documentation References*

*List all references using APA style*

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | <https://github.com/kkubli3246/libraryApp/tree/master> |
| **Hosting URL:** | *The Hosting URL (if applicable).* |