Project Documentation for HamzaProject

# 1. Problem Statement

Title: HamzaProject - Library Management System  
  
 Problem Statement:  
 Managing a library involves keeping track of books, students, and the borrowing process. Traditional methods of maintaining records using physical ledgers or simple spreadsheets are prone to errors, inefficiency, and data loss. There is a need for a robust, efficient, and user-friendly digital system to manage library operations, including adding and viewing books, maintaining student records, and tracking book borrowing and returns.  
  
 Objectives:  
 - Develop a system that allows easy addition, editing, and viewing of books.  
 - Create a comprehensive student record management system.  
 - Implement a streamlined borrowing process that tracks which student has borrowed which book and manages the return and cancellation processes.  
 - Provide a user-friendly interface for librarians and administrators.

# 2. Project Plan and Schedule

## Project Plan:

- Phase 1: Requirements Gathering (1 week)  
 - Conduct interviews with stakeholders (librarians, administrators).  
 - Gather requirements and document them.  
  
 - Phase 2: System Design (2 weeks)  
 - Design database schema for storing books, students, and borrowing records.  
 - Create wireframes and mockups for the user interface.  
  
 - Phase 3: Development (4 weeks)  
 - Set up project environment.  
 - Develop backend using ASP.NET Core.  
 - Implement frontend using Razor Pages.  
 - Integrate with the database using Entity Framework Core.  
  
 - Phase 4: Testing (2 weeks)  
 - Conduct unit testing, integration testing, and system testing.  
 - Fix bugs and issues identified during testing.  
  
 - Phase 5: Deployment (1 week)  
 - Deploy the system to a production environment.  
 - Conduct user training and system handover.  
  
 - Phase 6: Maintenance (Ongoing)  
 - Provide ongoing support and maintenance.  
 - Implement additional features and improvements based on user feedback.

## Project Schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Duration | Start Date | End Date |
| Requirements Gathering | 1 week | 01/05/2024 | 07/06/2024 |
| System Design | 1 week | 08/05/2024 | 15/05/2024 |
| Development | 2 weeks | 16/05/2024 | 29/05/2024 |
| Testing | 1 week | 30/05/2024 | 06/06/2024 |
| Deployment | 1 day | 07/06/2024 | 08/06/2024 |
| Maintenance | Ongoing | 09/06/2024 | - |

# 3. System Analysis and Design

## System Analysis:

Actors:  
 - Librarian: Manages books and student records, handles borrowing and returns.  
 - Administrator: Oversees the entire system, manages librarian accounts, and generates reports.  
  
 Use Cases:  
 - Add/View Books  
 - Add/View Students  
 - Borrow Book  
 - Return Book  
 - Cancel Borrowing

## System Design:

Architecture:  
 - The system follows a Model-View-Controller (MVC) architecture using ASP.NET Core and Razor Pages.  
 - The backend is developed in C# with Entity Framework Core for database interactions.  
 - The frontend is designed using Razor Pages, HTML, CSS, and Bootstrap for styling.  
  
 Database Design:  
 Tables:  
 - Books: Stores book information (ID, Name, Author, ISBN, etc.).  
 - Students: Stores student information (ID, Name, Number, Email, etc.).  
 - StudentBorrows: Tracks borrowing records (ID, BookID, StudentID, Status, BorrowDate, ReturnDate).  
  
 Class Diagram:  
 Book  
 ———————  
 - Id: int  
 - Name: string  
 - Author: string  
 - ISBN: string  
  
 Student  
 ———————  
 - Id: int  
 - Name: string  
 - Number: string  
 - Email: string  
  
 StudentBorrow  
 ———————  
 - Id: int  
 - BookId: int  
 - StudentId: int  
 - Status: int  
 - BorrowDate: DateTime  
 - ReturnDate: DateTime  
  
 Relationships:  
 Book 1---\* StudentBorrow \*---1 Student

## Wireframes:

- Landing Page:  
 - Welcome message, description of goals, and navigation links to manage books, students, and borrowing.  
  
 - Books Page:  
 - Form to add new books and a table to display existing books.  
  
 - Students Page:  
 - Form to add new students and a table to display existing students.  
  
 - Borrow Page:  
 - Form to select a student and a book for borrowing, and a table to display current borrow records with status and cancel button.