
PROPOSAL

Software Development 2
CMP020L004S

Proposal for **BookOrbit –** a web-based library management system (LMS) by**:**

Tolulope Gideon Babajide A00020050

Franklin Kaetochukwu Ikeh A00021448

Joshua Amiadamhen AM123619605
Bowas Joy Chakkalamuttath A00021346

BOOKORBIT **4**

PURPOSE 4
PROJECT TEAM 4
PROBLEM STATEMENT 5
PROJECT OVERVIEW 5
FUNCTIONAL REQUIREMENTS DOCUMENT (FRD) 5
NON-FUNCTIONAL REQUIREMENTS DOCUMENT (NFRD) 6
KEY STAKEHOLDERS 6
PERSONAS 7
IMPLEMENTATION STRATEGY 8
TIMELINE 8
TASK BOARD 9
GITHUB PROJECT REPOSITORY 9
CODE OF CONDUCT 10
CONCLUSION 13

BookOrbit

Version: 1.0
Date: (01/02/2025)

Purpose

The purpose of this proposal is to present an end-to-end plan for developing BookOrbit - a **Library Management System (LMS)** web application that addresses the inefficiencies of manual library operations. By digitizing cataloging, borrowing, and user management, the LMS will streamline administrative tasks, reduce errors, and improve the overall user experience.

Scope

The proposal covers the project team structure, the solution's objectives, core functionalities, requirements, code of conduct, key stakeholders, user personas, and more. At the end, links to our task board and GitHub repository are also provided for transparency and collaboration.

Project Team

Team Name: The Innovators

1. **Project Manager:**
    - Name: Joshua Amiadamhen
    - Responsibilities: Overseeing project timelines, resource allocation, and stakeholder communication.
2. **Lead Developer**
    - Name: Tolulope Babajide
    - Responsibilities: System design, core back-end development, infrastructure decisions.
3. **Front-End Developer:** Bowas Joy
    - Name:
    - Responsibilities: UI/UX design, responsive interface, integrating design with back-end.
4. **Database Administrator:**
    - Name: Franklin Ikeh
    - Responsibilities: Database schema design, performance tuning, backup strategies.
5. **QA Engineer and Tester:**
    - Name: Dr Mamoona Humayun
    - Responsibilities: Writing test cases, manual and automated testing, ensuring quality standards.

Problem Statement

Traditional library operations often rely on **manual processes** for tasks such as tracking borrowed books and handling overdue returns. This approach leads to:

- **Time Delays:** Manual checking and record-keeping can slow down borrowing and returning procedures.
- **Record Inaccuracies:** Paper-based logs risk misplacement and errors in book status updates.
- **Limited Access:** Users may need to visit physically during specific hours to check availability or request a book.
- **Overdue Fines Mismanagement:** Manual fine calculations can be error-prone and time-consuming.

Goal: To develop a web-based LMS that **automates** and **digitizes** these operations, allowing for real-time updates, automated fine calculation, and self-service access for library members.

Project Overview

Objective: To create a web-based solution for comprehensive library management, addressing catalog maintenance, user roles, real-time availability, notifications, and reporting.

Target Users:

- **Admin:** Has top-level authority to manage users, system settings, and generate high-level reports.
- **Staff:** Manages the book catalog, supervises borrowing and returning, and administers fines.
- **Library Users (Students/Public):** Uses the system to search, borrow, and return books with minimal friction.

Functional Requirements Document (FRD)

1. **User Authentication & Role Management**
    - Secure login for admin, staff, and user roles.
    - Access levels vary based on assigned roles.
2. **Book Catalog Management**
    - Admin/Staff can add, update, or remove category/book entries.
    - Supports book details like ISBN, author, category, and availability status.

3. **Borrowing & Returning System**
    - Tracks checkout activity with automated due date calculations.
    - Updates real-time book availability and borrowing history.
4. **Fine Management**
    - Automatically calculates overdue fines based on library rules.
    - Sends payment reminders and updates user records.
5. **Search & Filter Books**
    - Allows searching by title, author, genre, and other filters.
    - Delivers quick, accurate results.
6. **Notifications & Reminders**
    - Sends out emails/SMS for upcoming due dates or overdue books.
    - Customizable frequency and channels of communication.
7. **Reports & Analytics**
    - Administrators can view usage metrics, borrowing trends, and fine statistics.
    - Provides data export options for deeper analysis.

Non-Functional Requirements Document (NFRD)

These define the **quality attributes** of the LMS.

1. **Fast Performance:** Optimize the database and use caching to ensure quick searches.
2. **Scalable Infrastructure:** Use cloud services and scaling tools to handle many users.
3. **Strong Security:** Hash passwords and encrypt data to protect user information.
4. **High Availability:** Use backup systems and reliable hosting to keep the app running.
5. **User-Friendly Design:** Focus on simplicity and gather feedback to improve the interface

Key Stakeholders

1. **Administrators:** Oversee system usage, manage high-level reports, and handle user escalations.
2. **Staff:** Manage book inventory, transactions, and user requests.
3. **App Users (Students/Public):** Primary users for borrowing and returning resources.
4. **IT & Development Team:** Provide technical maintenance, updates, and bug fixes.

Personas

Below are two detailed **personas** representing typical BookOrbit users.

Persona 1: Franklin Eze (College Student)

- **Age:** 29
- **Role:** Library User
- **Background:**
  Franklin is a postgraduate student at Roehampton University. He frequently needs research materials and likes to keep track of his due dates to avoid fines. He also prefers digital solutions that minimize the need for physical library visits due to his busy schedule of classes and extracurricular activities.
- **Goals & Motivations:**
  1. Quickly locate academic and reference materials.
  2. Receive automated due date reminders to avoid late fees.
  3. Simplify the borrowing and returning processes.
- **Frustrations & Pain Points:**
  - Manual processes that require him to sometimes stand in long lines.
  - Lack of real-time updates on book availability.
  - Inconsistent or missing notifications for due dates.
- **Usage Scenario:**
  - Franklin logs into BookOrbit from his dorm room, searches for a required textbook, and checks if it's available. He borrows it digitally or reserves a physical copy, then receives an email confirmation with the due date of when he is to return the book. When it is near the due date, he gets an SMS reminder and other few popup notifications, and these helps her keep to the deadline and she returns the book on time.

Persona 2: Emelda Okafor (Head Librarian)

- **Age:** 45
- **Role:** Library Staff
- **Background:**
  Emelda has been working at the institution's library for over 15 years. She oversees book acquisitions, manages library staff schedules, and ensures the library runs smoothly.
- **Goals & Motivations:**
  1. Maintain an up-to-date catalog with accurate information.
  2. Efficiently track overdue books and manage fines.
  3. Generate meaningful reports to help improve library services.
- **Frustrations & Pain Points:**
  - Inaccurate catalog data leading to user complaints.
  - Manual tracking of fines, which can be time-consuming.
  - Limited data insights into which books are most popular.

- **Usage Scenario:**
  - Emelda logs into BookOrbit each morning, checks the dashboard to see overdue books, and sends out fine reminders. She updates the catalog with any new arrivals and runs a monthly report to see which categories are being borrowed the most.

Implementation Strategy

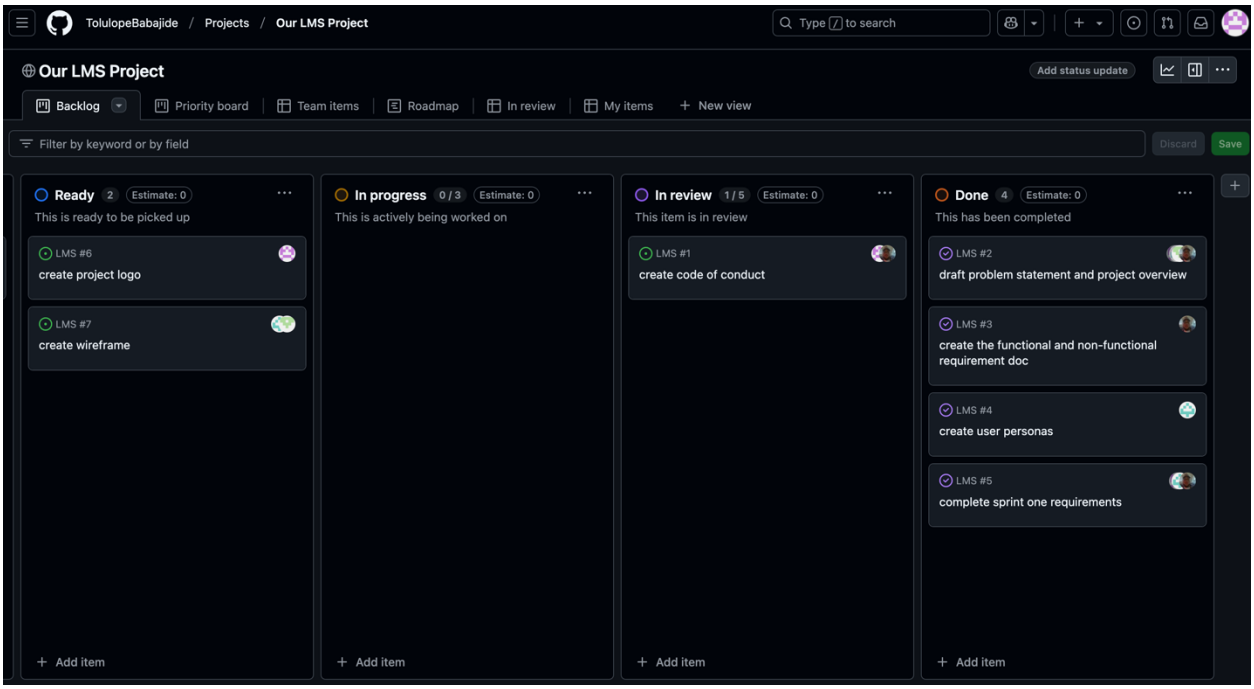
1. **Planning & Analysis:**
   - Finalize requirements and confirm the data model.
   - Define sprints and milestones on the task board.
2. **Design:**
   - Create UI wireframes and architectural diagrams.
   - Set up the database schema.
3. **Development:**
   - Implement core functionalities (authentication, book management, borrowing/returning).
   - Integrate notifications and fine management modules.
4. **Testing:**
   - Conduct unit, integration, and user acceptance testing.
   - Fix bugs and optimize performance.
5. **Deployment:**
   - Deploy on a cloud service for high availability (e.g. AWS, Azure).
6. **Maintenance & Support:**
   - Ongoing bug fixes, user feedback incorporation, feature enhancements.

Timeline

Phase	Duration	Key Deliverables
Requirements Analysis	1 week	FRD & NFRD finalized
System Design	1 week	UI mockups, architectural design
Development	3 weeks	Functional LMS modules: Borrow/Return, Fines
Testing	1 week	Bug-free, stable release candidate
Deployment	1 week	Production launch, documentation
Maintenance	Running	Continuous support & enhancements

Task Board

For project tracking, we are utilizing a Github Kanban board to manage tasks, user stories, and sprints. Click here to view our task board.



GitHub Project Repository

All code, issues, and version control activities are managed via GitHub here.

Code of Conduct

1. Respect and Professionalism

- Treat all group members with respect, kindness, and professionalism.
- Value diverse opinions and perspectives. Disagreements should be handled constructively.
- Avoid offensive language, personal attacks, or disrespectful behavior.

2. Communication Guidelines

- All group members should use the agreed-upon communication channels for project-related discussions.
- Respond to messages within **24 hours** or notify the group if you will be unavailable.
- Be clear, concise, and respectful in all communications.
- Raise concerns or issues promptly to avoid delays.

3. Attendance and Participation

- All group members must attend all scheduled meetings (in-person or virtual) on time. If you cannot attend, inform the group in advance.
- Actively participate in discussions, brainstorming sessions, and decision-making processes.
- Contribute equally to the project and take ownership of assigned tasks.

4. Task Management and Deadlines

- All group members must use the agreed-upon project management tool – Github Kanban board, to track tasks and progress.
- Complete assigned tasks by the agreed deadlines. If you anticipate a delay, inform the group immediately.
- Help other group members if they are struggling with their tasks and ask for help when needed.

5. Collaboration and Teamwork

- Share ideas, resources, and knowledge openly with the group.
- Be open to feedback and constructive criticism. Use feedback to improve your work.
- Avoid working in silos. Keep the group updated on your progress and challenges.

6. Accountability

- Each group member must take responsibility for their assigned tasks and deliverables.
- If a group member makes a mistake, he must own up to it and work with other group members to find a solution.

- Group members will hold each other accountable in a respectful and supportive manner throughout the duration of the project.

7. Academic Integrity

- While carrying out assigned tasks, kindly ensure you follow the institution's guidelines for academic integrity and ethical behavior.
- Do not share project details or code with individuals outside the group without permission.

8. Work-Life Balance

- Respect each other's time and personal commitments. Avoid scheduling meetings or setting deadlines outside reasonable hours.
- Take breaks when needed to maintain productivity and mental well-being.

9. Continuous Improvement

- As a group, we'll regularly reflect on the project's progress and identify areas for improvement.
- Be open to adapting processes or workflows to enhance efficiency and collaboration.

Conflict Resolution Process

Conflicts are natural in group projects, but they must be resolved constructively to maintain a positive and productive environment. These steps will be followed to address conflicts:

Step 1: Open Communication

- Address conflicts as soon as they arise. Do not let issues fester.
- Discuss the issue privately with the involved group member(s) in a calm and respectful manner.

Step 2: Seek Mediation

- If the conflict cannot be resolved privately, involve a neutral third party from the group to mediate the discussion.
- The mediator will facilitate a constructive conversation to identify the root cause of the conflict.

Step 3: Focus on Solutions

- Avoid blaming or personal attacks. Focus on finding a solution that works for everyone.
- Brainstorm possible solutions and agree on a compromise or action plan.

Step 4: Escalate if Necessary

- If the conflict remains unresolved after mediation, we'll escalate the issue to the course supervisor for guidance and resolution.

Step 5: Learn and Improve

- After resolving the conflict, we'll reflect on the situation and identify ways to prevent similar issues in the future.
- Use the experience to strengthen group communication and collaboration.

All group members commit to upholding these principles throughout the project. Failure to adhere to these guidelines may result in a discussion with the team and, if necessary, escalation to the course supervisor.

Conclusion

BookOrbit aims to streamline library operations, reduce overhead, and significantly enhance the user experience. By embracing a modern, scalable, and secure architecture, it addresses the main challenges faced in traditional library environments.