PES UNIVERSITY, BANGALORE

Department of Computer Science and Engineering

B. Tech (CSE) – 5th Semester – Aug-Dec 2024

UE22CS341A - Software Engineering

PROJECT PLAN DOCUMENT
EXAM CENTRE MANAGEMENT SYSTEM

Team #: 25

PES2UG22CS003	AAKANKSH SEELIN
PES2UG22CS032	ADITI ROOPESH MIRJI

Life-cycle followed

The chosen lifecycle is a waterfall model as the requirements for the project are clear and well defined.

1. Product Certainty (High):

i. The project requirements and functionality are well-defined.
ii. The system will manage books, members, and transactions, with clear CRUD operations on each entity.
iii. The objectives of the system, including member registration, book management, borrowing, and returning operations, are standard functionalities that are predictable and well understood.
iv. There are no ambiguous or experimental features in the system, ensuring a high level of confidence in the product.

2. Process Certainty (High):

i. The chosen Waterfall model provides a structured approach with clear phases: requirements, design, development, testing, and deployment. Since each phase has clear goals and deliverables, it eliminates uncertainties.

3. **Resource Certainty (High):**
   i. The tools and technologies needed for the project, such as MySQL Workbench, Python, Streamlit, and version control systems, are widely available and accessible.
   ii. We have access to the required resources, including development tools, databases, and libraries, ensuring no significant gaps in expertise or tooling.
   iii. Since this is a known system with no high-level complexity, the resource allocation is predictable and manageable.

The degree of certainty for **Product, Process and Resource is – high,high,high.**

Tools Used for this Project

i. **Planning Tool**: Google Sheets- for initial planning, task distribution, and creating simple timelines.

ii. **Design Tool:** smartdraw – for creating flowcharts, ER diagrams, and other design diagrams; Figma – for creating a wireframe of the UI if required.

iii. **Version Control:** Git & GitHub- for managing our source code, maintaining versions, and collaboration.

iv. **Development Tool:** Python (in VS Code) with Streamlit for quick iterations on the frontend; MySQL for backend development.

v. **Bug Tracking:** GitHub Issues (for tracking bugs and feature requests as they emerge).

vi. **Testing Tool:** PyTest for unit testing in Python; Selenium for web application testing (if required).

Deliverables classified as reuse/build components

1.Reuse Components:

i. ER diagrams for reference.
ii. Using MySQL as the back-end database system to manage records like books, members, loans, etc.,
iii. MySQL connectors, frameworks.
iv. Python and Streamlit libraries
v. Basic frontend components .

Reusing existing, reliable libraries and connectors allows us to save development time and ensure the robustness of our system.

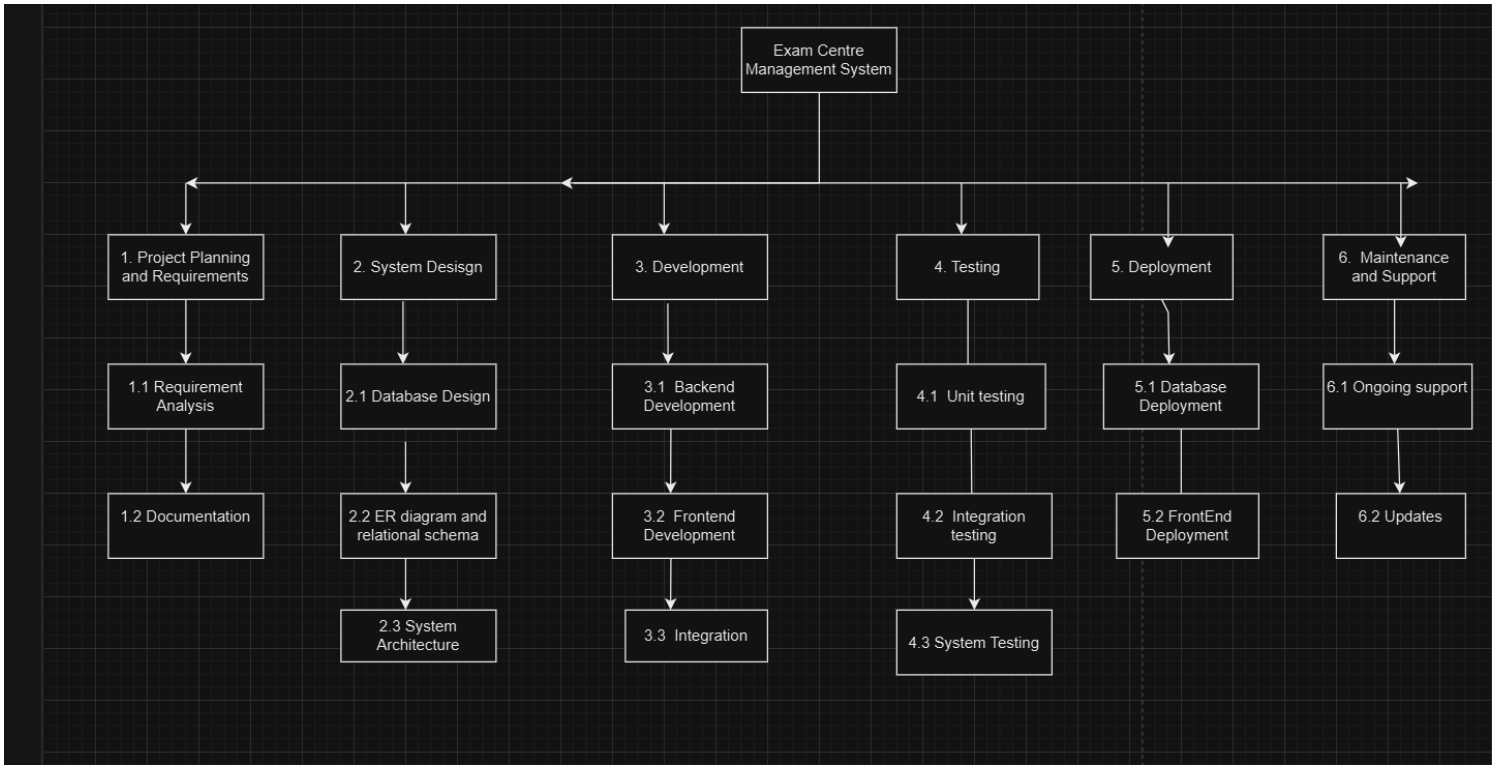
2.Build Components:

i. Custom user authentication
ii. CRUD operations for the Exam Centre Management System.
iii. A custom front-end interface for library administrators and users to manage books, members, and transactions.
iv. The database schema for books, members, transactions, and overdue fees designed specifically for this EMS.
v. The core functionality such as issuing and returning books, calculating late fees, tracking availability.

These components need to be custom-built as per project-specific requirements, like member management, book transactions, etc.

Work Breakdown Structure

This project follows a deliverable-based WBS, as each phase produces a specific outcome (e.g., database setup, UI development, deployment). We have four deliverables in the form of SRS, Project Plan, Testing details and final report. Hence, this project follows a deliverable based WBS.



1. Project Planning and Requirements
   - 1.1 Requirement Analysis
     - 1.1.1 Identify functional and non-functional requirements.
     - 1.1.2 Define scope (CRUD operations, book management, member management, transactions).
   - 1.2 Documentation
     - 1.2.1 Create Software Requirements Specification (SRS).

    - 1.2.2 Review and finalize project plan with timeline and deliverables.

2. System Design
- 2.1 Database Design
    - 2.1.1 Design the Books Table (fields: book_id, title, author, genre, availability_status).
    - 2.1.2 Design the Members Table (fields: member_id, name, contact_info, membership_id).
    - 2.1.3 Design the Transactions Table (fields: transaction_id, book_id, member_id, borrow_date, return_date).
- 2.2 ER Diagram and Relational Schema
    - 2.2.1 Create an ER diagram to model relationships between books, members, and transactions.
    - 2.2.2 Develop the relational schema for MySQL database.
- 2.3 System Architecture
    - 2.3.1 Define system architecture (frontend in Streamlit, backend in MySQL).
    - 2.3.2 Plan API structure for frontend-backend communication.

3. Development
- 3.1 Backend Development
    - 3.1.1 Set up MySQL database with defined schema.
    - 3.1.2 Implement CRUD operations for Books Table (add, update, delete, view).

    - 3.1.3 Implement CRUD operations for Members Table (register, update, delete, view).
    - 3.1.4 Implement transaction management (borrow, return, and view transactions).
- 3.2 Frontend Development
    - 3.2.1 Build user interface in Streamlit for book management.
    - 3.2.2 Build UI for member registration and management.
    - 3.2.3 Build UI for transaction management (borrowing and returning books).
- 3.3 Integration
    - 3.3.1 Connect frontend with MySQL backend.
    - 3.3.2 Test integration of all CRUD functionalities.

4. Testing
- 4.1 Unit Testing
    - 4.1.1 Test individual CRUD operations for books, members, and transactions.
- **4**.2 Integration Testing
    - 4.2.1 Test integration between frontend and backend.
    - 4.2.2 Ensure data consistency and smooth flow between components.
- 4.3 System Testing
    - 4.3.1 To validate entire flow .

5. Deployment
- 5.1 Database Deployment

    - 5.1.1 Deploy MySQL database on server.
- 5.2 Frontend Deployment
    - 5.2.1 Deploy Streamlit frontend to web server.

6. Maintenance and Support
- 6.1 Ongoing Support
    - 6.1.1 Monitor system performance.
    - 6.1.2 Resolve any post-deployment bugs.
- 6.2 Updates
    - 6.2.1 Plan for future system updates and enhancements.

Effort Estimation (in person-months)

This project is to be completed in the months of August-November 2024.

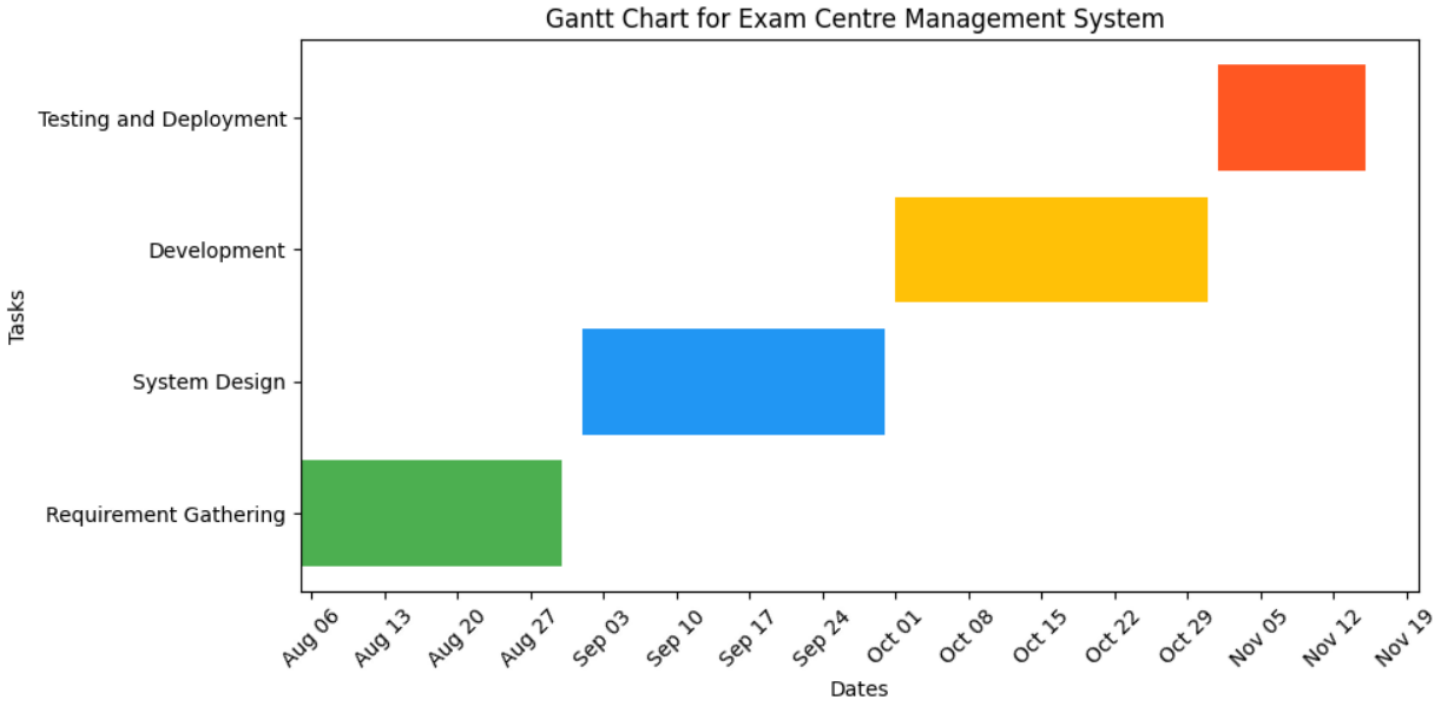
No of months =4, no of persons = 2.

No of working days = (260/12) * 4 = 86.6 working days.

Effort for one person would be 7/86.6 = 0.080 P-M

Effort for two persons= 0.161 P-M

Gantt Chart


Gantt Chart for Exam Centre Management System
Testing and Deployment
Development
System Design
Requirement Gathering
Tasks
Aug 06
Aug 13
Aug 20
Aug 27
Sep 03
Sep 10
Sep 17
Sep 24
Oct 01
Oct 08
Oct 15
Oct 22
Oct 29
Nov 05
Nov 12
Nov 19
Dates