**Date:** 23-07-25

**University Library System (ULS)**

This document presents the development plan of the University Library System (ULS) following the **Software Development Lifecycle (SDLC)** methodology. The system is intended to streamline the management of library resources, improve user experience, and support the university’s library operations.

**Software Development Lifecycle (SDLC) Phases**

The development process of the ULS is structured into the following phases:

1. **Communication:**  
   Interaction with stakeholders to identify project goals and expectations.
2. **Requirements Gathering:**  
   Collection of detailed functional and non-functional requirements from stakeholders.
3. **Analysis:**  
   Examination of requirements to create models such as use case diagrams and identify system feasibility.
4. **Design:**  
   Planning the architecture, database, and interfaces of the system, supported by design diagrams.
5. **Implementation:**  
   Actual coding and integration of system modules based on the designs.
6. **Testing:**  
   Verifying the system’s functionality, performance, and compliance with requirements.
7. **Deployment:**  
   Installing and configuring the system in the live environment.
8. **Maintenance:**  
   Regular updates, bug fixes, and enhancements to ensure smooth operation over time.

**University Library System (ULS)**

**1. Communication**

The project begins by identifying all relevant stakeholders:

* **Admin:** Responsible for system configuration and overall management.
* **Staff:** Handles day-to-day operations such as issuing and returning books.
* **Members:** Users (students/faculty) who borrow and return books.

The objectives and scope are discussed to ensure the system addresses the needs of each stakeholder effectively.

**2. Requirements**

The requirements for ULS are categorized into the following:

**2.1 Software Requirements**

* Application software (custom ULS application)
* Operating system to host the application
* Database management system for data storage

**2.2 Hardware Requirements**

* Personal computers or laptops for staff and admin
* Server to host the system
* Network infrastructure for connectivity

**2.3 Functional Requirements**

* User account creation and management
* Book catalog management (add, remove, update books)
* Search functionality for books
* User authentication (login/logout)
* Book checkout and return
* Borrow history tracking
* Notifications (reminders, overdue notices)
* Fine calculation based on borrow period

**2.4 Non-Functional Requirements**

* High performance under load
* Secure data storage and access
* User-friendly interfaces
* Ability to scale with increased usage
* High system availability and reliability

**2.5 Domain Requirements**

* Manage library resources
* Manage user accounts
* Track book borrowing and return activities

**2.6 User Requirements**

* Three main user roles: Admin, Staff, Member

**2.7 System Requirements**

* Should support deployment as a:
  + Web application
  + Mobile application
  + Desktop application

**3. Analysis**

This phase involves analyzing the collected requirements to identify use cases, relationships, and possible system constraints.

**3.1 Use Case Diagram**

A use case diagram will be created to illustrate the interactions between system actors and their respective actions in the ULS.

**4. Design**

Designing the architecture and components of the system, both at high and low levels, to meet the specified requirements.

**Design components include:**

* **Entity-Relationship (ER) Diagram:** Defines database entities and their relationships.
* **Frontend Design:** User interface design for web, desktop, and mobile.
* **Backend Design:** Design of server-side logic and database access.
* **Code Implementation Plan:** Scheduling and organizing the development of system modules.

**Diagrams to be designed:**

* Use Case Diagram
* Sequence Diagram
* Collaboration Diagram
* Class Diagram
* Component Diagram
* State Diagram
* Activity Diagram
* Deployment Diagram
* Object Diagram

These diagrams help visualize system behavior, structure, and deployment.

**Use Case Diagram for ULS**

The use case diagram will visually depict the interactions of the three main actors (Admin, Staff, Member) with the system’s use cases, such as:

* Searching books
* Checking out books
* Returning books
* Managing user accounts
* Sending notifications

**Relationships in Diagrams**

The design phase also considers the types of relationships among system components:

1. **Association:**
   * **Single-Directional:** One actor or component initiates interaction with another without reciprocity.
   * **Bi-Directional:** Both entities interact mutually with each other.

USE\_CASE DIAGRAM:

