**COVER PAGE**

A logo of a university

Description automatically generated

EEC 521/CIS 534: Software Engineering

Professor: Dr. Yongjian Fu

|  |  |  |
| --- | --- | --- |
| **Team Members** | | |
| Team Member 1 | LIKITH REKAPALLI | CSU ID: 2845684 |
| Team Member 2 | KRISHNA SAI SALLA | CSU ID: 2887720 |

**PROJECT PLAN**

**1.0 Introduction**

This project plan outlines the development of the "Health Care Hospital System," a web-based application designed to manage patient and staff information, invoices, and room details in a hospital environment. The plan provides an overview of the project's scope, major functions, performance requirements, management constraints, risk management, scheduling, and technical details.

**1.1 Project Scope**

The Health Care Hospital System is a web application built using PHP and MySQL for the backend, with HTML, CSS, JavaScript, Bootstrap, and Ajax for the frontend. It is designed to facilitate the management of key hospital operations, allowing administrators and staff to efficiently handle various tasks.

**Inputs:**

* Patient and staff data (via registration forms)
* Patient invoice data
* Room allocation data

**Processing:**

* Manage patient and staff information (CRUD operations)
* Generate and manage invoices for outpatient (OPD) and admitted patients
* Track room availability and assignments

**Outputs:**

* Admin dashboard displaying summary reports
* Updated patient and staff information
* Generated patient invoices
* Room availability and assignment updates

**1.2 Major Software Functions**

The system's primary functions are divided into several modules:

**Admin Dashboard:**

* Manage patient and staff data
* Oversee patient invoices and room allocation
* View summary information

**Patient Management:**

* Add, edit, delete, and search patient information
* View patient history and details

**Staff Management:**

* Add, edit, delete, and search staff information
* View active staff members

**Invoice Management:**

* Generate and view invoices for OPD and admitted patients
* Track and manage patient billing

**Room Information Management:**

* Monitor room availability
* Assign doctors to morning/evening shifts
* Track the number of rooms and their ID’s

**1.3 Performance/Behavior Issues**

* Response Time: System should respond within 2 seconds for data retrieval and updates.
* Scalability: Should support up to 100 simultaneous users without performance degradation.
* Security: Patient and staff information must be securely handled, and data access should be limited based on user roles (admin/staff).

**1.4 Management and Technical Constraints**

* Time Constraints: Project must be completed by the end of the academic term, with all milestones adhered to.
* Technical Constraints:
* Compatibility with all major browsers (Opera, Chrome, Mozilla, IE8).
* Deployment using WAMP, XAMPP, or other similar local server environments.
* Resource Constraints: Limited access to paid tools for testing and security implementation.

**2.0 Risk Management**

Effective risk management is crucial to identify, assess, and mitigate potential issues that could impact the project's success.

**2.1 Project Risks**

The following risks have been identified for the "Health Care Hospital System" project:

* Incomplete Features: Some modules of the system are not fully implemented or may contain bugs.
* Security Vulnerabilities: Limited security features may expose sensitive data.
* Cross-Browser Compatibility Issues: Older browsers may not fully support the modern technologies used.
* Time Overrun: Enhancing the incomplete features within a limited timeframe could be challenging.
* Database Issues: The system may encounter issues with database management or SQL errors.

**2.2 Risk Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk Name** | **Probability** | **Impact** | **RM3 Pointer** |
| Incomplete Features | High | High | Allocate more time for testing and bug fixes |
| Security Vulnerabilities | Medium | Medium | Implement basic security (Input validation, Access controls) |
| Compatibility Issues | Medium | Medium | Test on all supported browsers early |
| Time Overrun | High | High | Prioritize core functionalities for completion |
| Database Issues | Medium | Medium | Ensure SQL syntax and database configurations are correct |

* 1. **Overview of Risk Mitigation, Monitoring, Management (RM3)**

**Mitigation:**

* Focus on the most critical parts of the project, starting with incomplete features. Introduce basic security features and ensure cross-browser testing early in the process.

**Monitoring:**

* Regularly track the progress of each module and identify areas that need immediate attention.

**Management:**

* Prioritize time-sensitive tasks such as fixing incomplete features and conducting basic security tests to ensure the project is ready for submission.

**3.0 Project Schedule**

Given that the project is already partially developed, the following tasks will focus on completing, testing, and improving the existing functionalities.

**3.1 Project Task Set**

The following are the primary tasks associated with the Waterfall Model process:

1. **Requirements Analysis**

* Understand the existing functionality of the project.
* Define the project’s requirements and validate how much of it has been completed.
* Review the provided GitHub documentation and source code to identify functional and non-functional requirements.

1. **System Design Review**

* Perform a review of the existing system design.
* Identify how the functional and non-functional requirements have been addressed in the existing design.
* Create a design document (UML diagrams, Data Flow Diagrams) based on the existing codebase.

1. **Test Plan Development**

* Create a comprehensive test plan for the system based on the requirements and design review.
* Develop test cases to validate the functional requirements, even if actual testing won’t be performed.
* Document expected outcomes for each test case.

1. **Risk Analysis and Management**

* Identify project risks such as security vulnerabilities, data integrity issues, or architectural flaws.
* Develop a risk management strategy, including risk mitigation, monitoring, and contingency plans.

1. **Project Documentation and Report**

* Compile all the work done (requirements analysis, design review, testing plan, and risk management) into a final project report.
* Provide conclusions about the strengths and weaknesses of the system from a software engineering perspective**.**

**3.2 Functional Decomposition**

Each major module will be decomposed into smaller tasks to ensure comprehensive coverage of all system functionalities. Key functional breakdowns are:

1. **Patient Management Analysis**

* Document how the existing system handles patient CRUD operations.
* Verify if the system design supports expected behaviors.

1. **Staff Management Analysis**

* Evaluate the functionality for staff management, particularly the ability to add, edit, and search staff information.
* Invoice and Room Management Analysis

1. **Invoice and Room Management Analysis**

* Review the code and database structure for invoice and room information management.
* Each of these modules will be broken down into specific requirements, system design components, and test cases.

**3.3 Task Network**

The task network shows the dependencies between various project tasks. Tasks like "System Design" must be completed before "Implementation" can begin.

1. **Requirement Analysis 2. System Design Review 3. Testing Plan Development**

**4. Risk Analysis and Management 5. Project Documentation & Reporting**

**3.4 Timeline Chart**

The following Timeline chart provides a breakdown of the tasks and estimated duration of each activity:

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Start Date** | **End Date** | **Duration** |
| Requirement Analysis | 09-25-2924 | 10-08-2024 | 2 weeks |
| System Design Review | 10-09-2024 | 10-22-2024 | 2 weeks |
| Testing Plan Development | 10-23-2024 | 11-05-2024 | 2 weeks |
| Risk Analysis and Management | 11-06-2024 | 11-12-2024 | 1 week |
| Project Documentation & Reporting | 11-13-2024 | 11-19-2024 | 1 week |

**3.5 Schedule Compliance**

To ensure that the project stays on track, the following methods will be used for schedule compliance:

* Weekly progress meetings: Discuss progress, roadblocks, and task updates.
* Milestone reviews: Assess the completion of major project milestones.
* Task reporting: Regular updates on task progress and completion status.
* Issue tracking: Keeping a record of all the issues in an Excel file and acting accordingly.

**4.0 Appendix**

**4.1 Supporting Documents**

* Requirements specification document
* Wireframes and UI design mockups
* Database schema design
* Original GitHub project repository details
* Local installation steps and system requirements (XAMPP, WAMP, etc.)
* Testing results and issue logs

**4.2 References**

* Development tools: XAMPP, Sublime Text, PHPMyAdmin
* Frameworks: Bootstrap, JQuery
* Code repository: GitHub (original project source)