COVER PAGE

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

TEST PLAN

1.0 Introduction

This document outlines a structured approach to testing the Health Care Hospital System to ensure it meets the defined functional, non-functional, and usability requirements. It includes the testing strategy, test objectives, testing environment, tools, and detailed test cases.

1.1 Goals and Objectives

Goal: Validate the Health Care Hospital System's functionalities, ensuring accurate handling of hospital administrative tasks such as patient registration, staff management, room management, and invoicing.

Objectives:

- ➢ Verify system reliability and accuracy in handling administrative processes.
- ➢ Ensure user-friendly and responsive design across different devices.
- ➢ Detect and address potential security, usability, and functional issues.

1.2 Statement of Scope

The testing scope includes core functionalities:

Functionality/Features to be Tested:

- ➢ Patient Management (registration, view, edit, delete)
- ➢ Staff Management (add, edit, delete, search)
- ➢ Room Management (assignment, availability, tracking)
- ➢ Invoice Management
- ➢ User Interface and Access Control

Features Not Tested:

- ➢ Integration with external systems
- ➢ Performance under heavy load (e.g., multi-server deployment)

2.0 Test Plan

2.1 Software to be Tested

Software Name: Health Care Hospital System

Exclusions: Any incomplete or partially developed features, and external integrations.

2.3 Testing Tools and Environment

Environment:

- ➢ Web-based testing on different browsers: Chrome, Firefox, Safari.
- ➢ Local deployment on WAMP/XAMPP server with MySQL database.

2.4 Test Schedule

The test schedule for the Health Care Hospital System provides a structured timeline for completing the testing process to ensure the application's quality before its deployment.

- ➢ **Week 1** - Initial Setup and Environment Configuration: During this first week, the team configures the testing environment, installs the necessary testing tools (such as Selenium and MySQL Workbench), and prepares the application on the testing server. This phase includes ensuring that browsers, simulators, or devices align with target environments for testing compatibility.
- ➢ **Weeks 1** - Functional Testing: The focus of these weeks is validating the functionality of each module—such as Patient Management, Staff Management, Room Management, and Invoice Management. Test cases will be executed to confirm that each component behaves as expected, including form submissions, data retrieval, search functionality, and updates.
- ➢ **Week 2** - Integration Testing: Integration testing will focus on verifying interactions between different components. Here, the testing team will check if the modules (e.g., patient assignments to rooms or generating invoices based on patient and room data) work together smoothly and data flows correctly between them.
- ➢ **Week 3** - User Interface Testing: This week is dedicated to assessing the user experience across different devices, screen sizes, and browsers. Testing will ensure that the system's user interface is responsive and accessible, following best practices for design consistency, accessibility, and usability.
- ➢ **Week 4** - System and Acceptance Testing: In this final testing phase, system tests evaluate the complete, integrated system's performance, functionality, and reliability under typical and high-load conditions.

This structured schedule allows for focused testing on each aspect of the Health Care Hospital System, helping ensure its stability and functionality by project end.

3.0 Test Cases

The Test Cases section of the test plan provides a comprehensive list of individual tests designed to verify that each component of the Health Care Hospital System meets its specified requirements. Each test case includes specific information, like unique IDs, test inputs, expected outputs, and descriptions of what each test will validate. This detail helps ensure accuracy in testing, promotes consistency in the process, and allows testers to track the functionality and behavior of each part of the system. Following are the test cases identified for this project:

a) <u>Patient Management</u>

ID	Test Input	Expected Output	Description
TC-01	Enter valid patient details, click Submit	Patient record saved, displayed on patient list	Tests successful patient registration
TC-02	Leave required fields blank, click Submit	Error message indicating required fields	Tests form validation
TC-03	Enter patient Details, click Delete	Patient record removed from database	Tests patient deletion functionality
TC-04	Edit patient details, click Submit	Updated patient details displayed on list	Tests update functionality

b) <u>Staff Management</u>

ID	Test Input	Expected Output	Description
TC-05	Enter valid staff details, click Submit	Staff member added, visible in staff list	Tests staff addition functionality
TC-06	Enter duplicate staff D, click Submit	Error message for duplicate ID	Validates unique staff ID requirement
TC-07	Edit staff details, click Submit	Updated staff information displayed on list	Tests staff update functionality
TC-08	Search by staff name	Relevant staff records displayed	Tests search functionality

c) <u>Room Management</u>

ID	Test Input	Expected Output	Description
TC-09	Assign room to patient, click Submit	Room status updated to assigned, patient-room link created	Tests room assignment
TC-10	Leave fields blank, click Submit	Error message indicating required fields	Tests form validation

d) <u>Invoice Management</u>

ID	Test Input	Expected Output	Description
TC-11	Enter invoice details, click Generate	Invoice generated and displayed on the list	Tests invoice creation functionality

e) <u>User Interface and Access Control</u>

ID	Test Input	Expected Output	Description
TC-12	Login as admin	Dashboard and all pages accessible	Tests admin access control
TC-13	Login as general staff	Limited page access, based on role permissions	Tests role-based access control
TC-14	Access restricted page without login	Redirected to login page with error message	Tests unauthorized access handling
TC-15	Simulate incorrect login	Error message indicating invalid credentials	Tests login validation

f) Usability and Responsiveness Testing

ID	Test Input	Expected Output	Description
TC-16	Open app on different screen sizes	UI adjusts and maintains usability	Tests responsiveness on devices
TC-17	Simulate slow network for page load	Page loads appropriately, or error message displayed	Tests error handling for connectivity

g) Database and Data Integrity

ID	Test Input	Expected Output	Description
TC-18	Check data entry in the Patient table	Records correctly saved with all fields populated	Ensures data is correctly stored
TC-19	Validate total_fee calculation	Total fee matches sum of charges, doctor, and medicine fees	Tests calculation accuracy

These test cases cover various aspects of the system: basic functionality, edge cases, and complex interactions across modules like Patient Management, Staff Management, Room Management, and Invoice Management. For example, a test case for patient registration might include verifying input fields and ensuring data integrity, while a room assignment test case checks that patient details correctly link to available rooms and update statuses in the database. Each test aims to confirm the system's ability to handle both typical and unexpected scenarios, such as missing data entries or incorrect formats.

Additionally, the test cases allow for easier identification of issues by pinpointing where each failure occurs, ensuring efficient troubleshooting. This thorough approach ensures that the final product is robust, reliable, and user-ready.