K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch: A1	**Roll No.: 16010123012**
Experiment / assignment / tutorial No.: 07	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

Title: Designing test plan document for Mini Project

Aim: To learn and understand the way of developing the software by classical methods of software engg. Planning and monitoring, testing, validating of the project using tools and prepares a document for the same by using the concept of software engineering

---

Books/ Journals/ Websites referred:

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraq Hill, 2010 ,6th edition
2. Ian Somerville , Software Engineering , Addison Wesley,2011,9th edition
3 http://en.wikipedia.org/wiki/Software_requirements_specification

---

Test Plan Template:

MediSlot

Prepared by:

Aaryan Sharma

Aditey Kshirsagar

Aditya Baheti

Date: 08/10/2025

TABLE OF CONTENTS

2.0 OBJECTIVES AND TASKS
2.1 Objectives
2.2 Tasks

3.0 SCOPE

4.0 Testing Strategy
4.1 Alpha Testing (Unit Testing)
4.2 System and Integration Testing
4.3 Performance and Stress Testing
4.4 User Acceptance Testing
4.5 Batch Testing
4.6 Automated Regression Testing
4.7 Beta Testing

5.0 Hardware Requirements

6.0 Environment Requirements
6.1 Main Frame
6.2 Workstation

7.0 Test Schedule

8.0 Control Procedures

9.0 Features to Be Tested

10.0 Features Not to Be Tested

11.0 Resources/Roles & Responsibilities

12.0 Schedules

13.0 Significantly Impacted Departments (SIDs)

14.0 Dependencies

15.0 Risks/Assumptions

16.0 Tools

17.0 Approvals

1.0 INTRODUCTION

MediSlot is a smart, web-based platform designed to make healthcare more accessible, organized, and convenient for both patients and doctors. It simplifies the entire appointment process by allowing patients to quickly register, log in, search for doctors by specialization, check their availability, and book appointments online, all from one place, without the hassle of long waiting times or phone calls.
For doctors, MediSlot offers an easy-to-use dashboard where they can manage their schedules, update their availability, and keep track of upcoming appointments. One of the most useful features is the ability to view a patient's past medical history, which helps doctors provide more accurate diagnoses and personalized treatment based on previous health records.
The platform goes beyond just booking appointments. It keeps a record of all past and upcoming consultations, sends timely reminders and updates, and gives administrators the tools to manage doctors, patients, and system data efficiently.

2.0 OBJECTIVES AND TASKS

2.1 Objectives

The Master Test Plan for MediSlot is created to guide and organize all the testing activities for the project. It ensures that the appointment booking system, user dashboards, and all other features work correctly and meet user expectations. The main objectives are:

Define tasks and responsibilities:
It clearly states who is responsible for different testing tasks like checking patient registration, appointment booking, doctor availability, and medical history features so the work is done smoothly.

Improve communication:
The plan serves as a common reference for the whole team, including developers, testers, and project managers, helping everyone stay updated on what is being tested and when.

Set clear testing goals:
It explains what needs to be tested (e.g., login, appointment scheduling, notifications, past health records), how it will be tested, and what results are expected.

Provide a step-by-step roadmap:
The Master Test Plan shows the order of testing activities from unit testing of individual modules to complete system testing ensuring that no feature is missed.

Act as an agreement:
It works as a simple agreement between the development and testing teams about the quality standards that MediSlot must meet before it is launched.

Manage possible risks:
The plan helps identify potential problems in features like appointment scheduling, data handling, or notification delivery and prepares solutions in advance.

2.2 Tasks

The Master Test Plan lists all the important tasks that need to be carried out during the

testing process to ensure that MediSlot functions correctly and meets user requirements. These tasks include:
Functional Testing: Check all core features like patient registration, login, appointment booking, and viewing medical history.
UI/UX Testing: Ensure the interface is user-friendly, easy to navigate, and visually clear.
Security Testing: Verify that user data is safe and protected from unauthorized access.
Performance Testing: Check how quickly the system responds and handles multiple users.
Compatibility Testing: Make sure MediSlot works smoothly on different devices and browsers.

3.0 SCOPE

General
The scope of testing for MediSlot includes verifying all core functionalities and features of the system. This involves testing patient registration and login, doctor profile management, appointment scheduling and cancellation, viewing past medical history, notification delivery, and admin management modules. Additionally, it covers the testing of all existing interfaces, integration between different modules (like patient-doctor-appointment), and database interactions to ensure that the system works smoothly as a whole.

Tactics
To achieve the scope, the testing team will:

- Create and execute test cases for each core function, including user registration, appointment booking, and viewing medical records.
- Conduct integration testing to ensure smooth data flow between patient, doctor, appointment, and admin modules.
- Notify relevant team members (developers, testers, and project managers) about the testing schedule and areas requiring their input.
- Arrange review sessions and allocate time for collaborative testing activities, ensuring that all key stakeholders are available for support and feedback.
- Document and report any issues found during testing and track them until resolution.

4.0 TESTING STRATEGY

Our testing strategy focuses on breaking down **MediSlot** into smaller, manageable units and testing each one thoroughly to ensure that every component works correctly before integrating them into the full system. Once the individual modules are verified, they will be combined to perform system-level testing to ensure smooth end-to-end functionality.The approach emphasizes identifying issues early, ensuring stable integration, and confirming that the system meets both functional and performance expectations.

4.1 Unit Testing

Definition:
Unit testing will verify each small module like login, appointment booking, and medical history. All critical functions will be tested at least once, and coverage will be checked through test case execution and error frequency. Testing is considered complete when all unit test cases pass with minimal errors.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
Developers will write and run test scripts for individual modules using mock data. Test cases will cover valid, invalid, and boundary scenarios. Issues will be logged and fixed before integration testing.

4.2 System and Integration Testing

Definition:
System and Integration Testing will ensure that all modules of MediSlot such as patient, doctor, appointment, all work together smoothly as one system. It checks data flow, communication between components, and overall system behavior.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
Test scripts will be created by the testing team. After unit testing, modules will be gradually integrated and tested as a whole. Test cases will focus on workflows like appointment booking, notifications, and record retrieval. Any issues found will be logged, fixed, and re-tested to ensure proper integration.

4.3 Performance and Stress Testing

Definition:
Performance and Stress Testing will check how MediSlot behaves under normal and high load conditions. It ensures the platform remains stable, fast, and responsive when multiple users access features like booking appointments and viewing records at the same time.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
The testing team will prepare scripts to simulate multiple users and heavy loads. Key workflows like login, appointment booking, and data retrieval will be tested under peak conditions. System response time, stability, and error handling will be monitored. Any performance issues will be reported and optimized before final deployment.

4.4 User Acceptance Testing

Definition:
User Acceptance Testing (UAT) ensures that MediSlot meets the original requirements and is ready for real-world use. End users test the system to verify that it works as expected in practical scenarios.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
UAT scripts will be created based on key user workflows like registration, booking, viewing medical history, and notifications. End users will execute the tests in a near-live environment. Feedback and issues will be logged, reviewed, and resolved before the final release.

4.5 Batch Testing

Definition:
Batch Testing will ensure that multiple processes or scheduled tasks in MediSlot, such as appointment reminders and notifications, run correctly in sequence without errors.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
The testing team will create scripts to run automated or scheduled operations together. They will check for proper execution, order of processing, and data accuracy. Any failures or delays will be recorded and fixed before deployment.

4.6 Automated Regression Testing

Definition:
Automated Regression Testing ensures that after new changes or fixes are made in MediSlot, existing features like login, appointment booking, and notifications still work correctly without introducing new errors.

Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
Automated scripts will be prepared to re-run critical test cases whenever updates are made. These tests will quickly verify that core functions remain stable. Any failed test cases will be logged, analyzed, and corrected before moving to the next release stage.

4.7 Beta Testing
Participants:
Aaryan Sharma
Aditey Kshirsagar
Aditya Baheti

Methodology:
Beta testing will be conducted by selected users to try MediSlot in a real-world environment. They will use all major features like registration, appointment booking, viewing medical history, and notifications. Feedback will be collected, issues will be reported, and necessary improvements will be made before the final release.

5.0 HARDWARE REQUIREMENTS

Hardware	Configuration	No. of units
Computer	Intel Core 5, 8GB RAM, 50GB Hard disk.	1

6.0 ENVIRONMENT REQUIREMENTS

A suitable browser with active internet connection.

6.1 Main Frame
Hardware: Computers with Intel i5 or higher, 8GB RAM, and stable internet.

Software:
OS: Windows / macOS / Linux
Browsers: Chrome, Edge, Firefox
Backend: Node.js with MongoDB
Tools: Postman, Excel/Google Sheets for tracking

Mode of Usage: Web-based testing in stand-alone or networked setup.

Security: Only authorized testers, dummy data used, secured access.

6.2 Workstation
Each tester will use a personal workstation to perform testing activities.

Specifications:
Hardware: Standard desktop or laptop with Intel i5 or higher processor, 8GB RAM, and internet access.
Software: Web browser (Chrome/Edge/Firefox), Postman for API checks, and access to the MediSlot web application.
Mode of Usage: Stand-alone systems connected to the main test server.
Security: Only authorized users can log in; all test data is sample-based.

7.0 TEST SCHEDULE

Duration	Milestone	Output
1st week	Unit testing	Documentation and feedback.
2nd week	System integration testing	Developer's comments, automated tool reports.
3rd week	Performance testing and User Acceptance testing	User's feedback, system performance report
4th week	Beta testing	User's feedback

8.0 CONTROL PROCEDURES

Problem Reporting
If any issue is found during testing, it will be noted in a defect log or tracking sheet. Each problem will include details like where it occurred and what went wrong. The tester will report it to the project lead or developer, who will fix it and send it back for re-testing.

Change Requests
If a change or update is needed, it will be discussed with the project lead before making it. Once approved, the change will be made and tested again to make sure everything still works properly.

9.0 FEATURES TO BE TESTED

The following features of MediSlot will be tested to ensure proper functionality and performance:

User Registration and Login – Verifying secure sign-up, login, and authentication for patients, doctors, and admins.
Appointment Booking and Cancellation – Ensuring users can book, reschedule, and cancel appointments smoothly.
Doctor Availability Management – Testing doctor schedule setup and time slot updates.
Patient Medical History – Checking that patients and doctors can view and update past health records accurately.
Notifications and Reminders – Validating that email/SMS notifications are sent correctly for appointments and updates.
Admin Functions – Testing admin access to manage users, doctors, and system data.
User Interface (UI) – Verifying that all pages are user-friendly, responsive, and work across devices.
Database Integration – Ensuring data is stored, retrieved, and updated correctly between modules.

10.0 FEATURES NOT TO BE TESTED

The following features will not be tested in this phase of MediSlot due to limited scope and resources:
Third-Party API Integration – External hospital or pharmacy system connections are outside the current testing scope.
Mobile Application Version – Only the web version of MediSlot will be tested at this stage.
Advanced Analytics or Reporting – Detailed data reports and analytics will be tested in future releases.
Voice or Chatbot Support – Planned for future updates, not part of current testing.
Reason: These features are either not developed yet or are planned for future versions of MediSlot.

11.0 RESOURCES/ROLES & RESPONSIBILITIES

Aaryan Sharma: Preparation of Slides, resolving test cases
Aditey Kshirsagar: Preparing, executing and resolving test cases.
Aditya Baheti: Managing, designing test cases.

12.0 SCHEDULES

Major Deliverables

- Test Plan
- Test Cases
- Test Incident Reports
- Test Summary Reports

13.0 SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs)

SID	Department-in charge
Development	Aaryan Sharma
Design(UI/UX)	Aditya Baheti
Quality Assurance	Aditey Kshirsagar

14.0 DEPENDENCIES

The testing of MediSlot depends on the completion of key modules like login, appointments, and database setup. It also relies on the availability of testers, developers, and required tools. A stable test environment and prepared sample data are essential. Any delay in development or updates may affect the testing schedule.

15.0 RISKS/ASSUMPTIONS

Development Delays: Testing may need rescheduling or extended hours.
Resource Shortage: Focus will shift to critical features first.
Server/Network Issues: Backup systems will be used if needed.
Data Loss: Regular backups will ensure quick recovery.
Unexpected Bugs: Extra time will be kept for fixes and re-testing.

16.0 TOOLS

Postman for automation and no specific tools for bug testing.

17.0 APPROVALS

Prof. Mansi Kambli

Signature Date

Conclusion:

This experiment helped us understand how to create a proper test plan for a software project. Using MediSlot, we learned how to plan, organize, and document different testing stages to ensure software quality and reliability. It also showed the importance of teamwork, planning, and clear testing procedures before deployment.

Post Lab Descriptive Questions:

1. **Distinguish between Black Box and White Box Testing**

Ans1)

Aspect	Black Box Testing	White Box Testing
Knowledge Required	No code knowledge	Requires code knowledge
Focus	Functional behavior	Internal logic and structure
Tester Type	End-user or QA tester	Developer or technical tester
Eg. in MediSlot	Login/Booking function	Database and code logic