Batch: A1	**Roll No.: 16010123012**
Experiment No.: 3	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Project 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 of the project using tools and prepare a document for the same by using the concept of software engineering

Expected OUTCOME of Experiment:

Analyse the software requirements and Model the defined problem with the help of UML diagram

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

Software Project Management Plan

for

MediSlot

Aaryan Sharma (16010123012)

Aditey Kshirsagar(16010123017)

Aditya Baheti(16010123023)

6th August, 2025.

Version	Release Date	Responsible Party	Major Changes
1.0	06/08/25	KJSSE	Initial Document Release for Comment

Table of Contents

1. Introduction

1.1 Project Overview

Medislot is built to make scheduling medical appointments simple and hassle-free for patients, doctors, and clinic staff. The project involves understanding what users need, designing and building the system, testing it thoroughly, and then launching it. Along the way, we'll hit key milestones like creating a working prototype and completing the final version. We'll need tools, hosting, and a dedicated team to get it done. While Medislot will work independently at first, there's room to connect it with existing hospital systems down the line.

1.2 Project Deliverables

For Medislot, we will deliver a fully working web-based appointment scheduling system, ready to use online. Along with the software, we'll provide easy-to-follow user manuals and help guides to support patients, doctors, and admins. Technical documentation will also be shared to help the client's IT team maintain the system. All deliverables will be provided digitally and according to the project schedule.

1.3 Evolution of the SPMP

This SPMP will serve as a dynamic guide throughout the development of the "MediSlot" project. As the project progresses, the plan may evolve to accommodate any changes in requirements, timelines, or resources. Regular updates will ensure that the SPMP remains aligned with the project's objectives and facilitates effective project management.

1.4 Reference Materials

This project referenced the online scheduling platform Calendly (https://calendly.com/) for its appointment booking features and user experience. Calendly, developed by Calendly, Inc., served as a key source of inspiration in designing Medislot's scheduling system.

1.5 Definitions and Acronyms

- SPMP: Software Project Management Plan
- PWA: Progressive Web Application
- MediSlot: Medislot refers to the smart medical appointment scheduling system being developed.

2. Project Organization

2.1 Process Model

1. Project Initiation

Activities:
1. Define project scope and goals.
2. Gather and document requirements.
3. Prepare a detailed project plan.

Entry Criteria:
Approved requirements and project charter.

Exit Criteria:
Signed project charter, defined roles, and final requirements.

2. Product Development

Activities:
1. Design system architecture, database, and user interfaces.
2. Develop core features and perform unit testing.

Entry Criteria:
Finalized requirements and environment setup.

Exit Criteria:
Core modules developed, unit tests passed.

3. Testing & QA

Activities:
1. Perform functional, integration, and UAT testing.
2. Identify and fix bugs, perform regression testing.

Entry Criteria:
Code completed, unit tests passed.

Exit Criteria:
Successful completion of all tests, critical issues resolved.

4. Product Release

Activities:
1. Deploy to production environment.
2. Monitor post-deployment for issues.

Entry Criteria:
Testing completed and stakeholder approval.

Exit Criteria:
Successful deployment with no major issues detected.

5. Project Termination & Maintenance

Activities:
1. Provide ongoing support and maintenance.
2. Gather feedback, address bugs, and release updates.

Entry Criteria:
Product deployed and stakeholder feedback collected.

Exit Criteria:
Stable system, project closure, and final feedback review.

2.2 Organizational Structure

The internal management structure of the "MediSlot" project consists of the following key roles:

- Project Manager: Leads the project team, oversees project planning and execution, and is responsible for project deliverables.
- Technical Team Leader(s): Leads the technical aspects of the project, including code development, architecture, and quality assurance.

The project operates independently within the Department of Computer Engineering at K. J. Somaiya College of Engineering, Mumbai-77, and is part of the educational and research activities within the department.

2.3 Organizational Interfaces

Describe the administrative and managerial interfaces between the project and the primary entities with which it interacts. A table may be a useful way to represent this.

ORGANIZATION	LIASON	CONTACT INFORMATION
Customer	Project Manager	officesupplies@gmail.com
Software Quality Assurance	Assurance Architecture	testify@gmail.com
Software Configuration	Project Manager	testify@gmail.com

Project Interfaces

The project management function has the following responsibilities

1. Schedule and prepare meetings with clients
2. Assign presentations (in-class project meetings, client review, client acceptance test) to project members
3. Listening to gripes from the team members
4. Resolve conflicts if they cannot be resolved otherwise

Architecture Liaison: The liaison interacts with the liaisons of the other teams and with the project management. Each team has a liaison to the Architecture Team. The responsibilities of the liaison are:

1. Responsible for intergroup communication
2. Make available public definitions of each subsystem service ("API") to the other teams (ensure consistency, etc.)
3. Coordinate tasks that overlap subsystems with the teams
4. Responsible for team negotiations, that is, resolve technical issues spanning more than one subsystem

2.4 Project Responsibilities

ROLE	DESCRIPTION	PERSON
Project Manager	Leads project team; responsible for project deliverables	Aditya Baheti
Technical Team Leader(s)	Responsible for the business logic, creating backend and internal functionality	Aaryan Sharma, Aditey Kshirsagar

3. Managerial Process

3.1 Management Objectives and Priorities

Philosophy

The project management philosophy for Medislot is to deliver a high-quality, secure, and reliable medical appointment scheduling platform that caters to the diverse needs of patients and healthcare providers. The focus is on ensuring a seamless user experience through features such as an intuitive booking process, secure data handling, timely notifications, and accessibility for all user groups.

Goals and Priorities

- **Cost: Fixed** – The project will be delivered within the allocated budget without compromising on essential features and security standards.
- **Schedule: Fixed** – The platform must launch within the established timeline to meet stakeholder and market expectations.
- **Scope (Functionality): Flexible** – Core functionalities like appointment booking, doctor dashboards, patient profiles, and notifications are defined, but additional enhancements may be introduced based on user feedback, emerging requirements, and technological feasibility.

Project Dimension	Fixed	Constrained	Flexible
Cost	✓		
Schedule	✓		
Scope (functionality)			✓

3.2 Assumptions, Dependencies, and Constraints

Assumptions:

The project assumes that the team possesses the necessary skills and expertise to build the Medislot appointment booking platform effectively. It also assumes access to essential hardware, software, and secure hosting resources required for development and deployment.

Dependencies:
The project is dependent on timely access to healthcare provider schedules, reliable integration with third-party SMS/email notification services, and stable APIs for payment processing and location services.

Constraints:
The project is constrained by budget and time limitations. The priority is to meet the established schedule while delivering the defined core functionality, ensuring compliance with healthcare data privacy regulations.

3.3 Risk Management

For the Medislot project, we'll handle risks by keeping a close, ongoing watch, spotting potential issues early, understanding their impact, and having backup plans ready. We'll track everything in a simple risk log that's reviewed regularly, so nothing slips through the cracks. Possible challenges, like delays from partners, tech glitches, managing multiple user types, losing key team members, or low user adoption, will be tackled with clear agreements, solid testing, flexible design, team cross-training, and early user feedback. If something goes wrong, our contingency plans will kick in quickly to keep the project on track.

3.4 Monitoring and Controlling Mechanisms

To ensure effective monitoring and control of the project, the following mechanisms will be implemented:

- Reporting Mechanisms: Regular status reports will be generated by the project team and submitted to the Project Manager on a weekly basis. Monthly project reviews will be conducted.
- Review and Audit Mechanisms: Monthly project reviews will involve a comprehensive review of the project's progress, issues, and risks.
- Communication Plan: A communication and reporting plan will be established to ensure that the Technical team, Project Manager, and other stakeholders are informed at key points in the project.

Information Communicated	From	To	Time Period
Status Report	Technical Team	Project Manager	Weekly
Status Report	Project Manager	Technical Team	Weekly

Project Review	Technical Team	Project Manager	Monthly

3.5 Staffing Approach.

For the Medislot project, we need a well-rounded team with the right mix of skills from coding and API integration to database handling, design, and testing. We'll also need people who are good at managing tasks, communicating clearly, and solving problems quickly. The team will be chosen based on their past work, proven expertise, and how well they can adapt to the project's needs. If anyone needs to brush up on certain skills, we'll arrange focused training and collaborative sessions so everyone stays on the same page and can deliver their best work together.

4. Technical Process

4.1 Methods, Tools, and Techniques

For the "Medislot" project, the following methods, tools, and techniques will be utilised:

VS Code	Code Editor
Chrome	Internet Browser
Github	Version Control
Figma	Prototype Creation
Canva	Design Creation
PowerPoint	Slide Presentation

- Development Method - Agile Software Development with iterative and incremental approach to enable flexibility, rapid updates, and continuous integration for Medislot's scheduling and booking features.
- Programming Languages - JavaScript (React.js),Node.js, Express.js, NoSQL and other relevant languages for server-side development.
- Development Tools - Visual Studio Code, Figma for UI/UX design, and Docker for containerization

- Standards, Policies, and Procedures - Adherence to industry-standard coding conventions, Secure handling of medical and personal data following data privacy policies and use of project management practices aligned with Agile methodology.
- Team Structure - A cross-functional team composed of software developers, designers, and quality assurance testers.

Other Notations, Tools, Techniques, and Methods

- Version Control: Git and GitHub for collaborative coding and branch management.
- Testing: Unit testing, integration testing, and user acceptance testing (UAT) to ensure platform reliability.
- Documentation: Standardized formats with clear style guides for user manuals, developer notes, and API documentation.
- Integration: Continuous Integration/Continuous Deployment (CI/CD) pipelines for efficient updates.
- Maintenance: Regular updates and bug fixes managed via issue tracking tools.

4.2 Software Documentation

For this project, we will create several key documents:

- **Software Requirements Specification (SRS):**
  This document will clearly describe what the software needs to do, its main functions, performance goals, design limits, and how it interacts with other systems. It will be reviewed by team members to catch any issues early.
- **Software Design Description (SDD):**
  The design document will explain how the software is structured, including key components, databases, and how different parts connect and communicate.
- **Software Test Plan:**
  This plan will outline how we're going to test the software at every stage from checking requirements, to validating the design, and finally testing the actual code. It will detail test procedures, specific test cases, and how results will be tracked. The team will review this to ensure thorough testing.

4.3 User Documentation

User documentation will be developed to assist users in effectively using the MediSlot platform. This documentation will include online help, guides, and tutorials. The user documentation plan will be created to outline the process of developing and maintaining this documentation.

4.4 Project Support Functions

To make sure our project runs smoothly and delivers high-quality software, we've planned for some important supporting activities. These include managing our project files and versions, ensuring quality throughout development, and verifying that the software works as expected. Each of these areas has clear responsibilities, resources, schedules, and budgets aligned with the overall project.

- **Configuration Management:**
  We'll keep track of all project materials like code, documents, and deliverables and manage changes carefully. This ensures everyone is working on the latest versions and helps avoid mix-ups. We'll assign team members to handle this and include it in our project timeline and budget.
- **Software Quality Assurance:**
  Quality is a priority. Our SQA plan will outline how we'll monitor the project's quality through regular checks, code reviews, and audits. We'll define who's responsible and when these activities will happen to keep everything on track.
- **Verification and Validation:**
  To make sure the software truly meets all requirements, we'll carry out thorough testing and reviews throughout the development process. The plan will specify what tests will be done, who will do them, and what success looks like.

5. Work Packages, Schedule, and Budget

5.1 Work Packages

For the MediSlot project, we've broken down the work into clear, manageable parts, each with its own focus and goals. This helps us stay organized and makes sure everyone knows what to work on:

1. **Getting Started (WP-1):** First, we'll set our project goals, figure out who's doing what, and nail down the project's scope. We'll also put together an overall plan to guide us.

2. **Gathering Requirements (WP-2):** Next, we'll talk with patients, doctors, and admins to understand what they need. We'll list out all the features the system should have and write detailed specifications.

3. **Design and Prototyping (WP-3):** Then, we'll design the look and feel of the system for all users and build the overall structure behind the scenes. We'll create prototypes

so users can test things early and give feedback.

4. **Development (WP-4):** This is where we build the actual system—coding the front-end and back-end, creating key features like booking appointments and sending reminders, and connecting with email/SMS services.

5. **Testing and Quality Checks (WP-5):** We'll rigorously test every part of the system to make sure it works well and is easy to use. We'll also listen to user feedback and fix any issues.

6. **Documentation and User Guides (WP-6):** We'll create easy-to-understand manuals and help guides for patients, doctors, and admins. Plus, we'll prepare tutorials to help users get started.

7. **Launch and Deployment (WP-7):** Once everything's ready, we'll deploy MediSlot online, set up the servers securely, and open it up for users.

8. **Ongoing Support and Monitoring (WP-8):** After launch, we'll keep an eye on how the system is performing, provide support to users, and regularly update the platform based on feedback.

5.2 Dependencies

In our MediSlot project, some work packages (WPs) need to be completed before others can begin. These dependencies help keep the project organized and on track.

- We need to complete Requirements Gathering (WP-2) before starting Design and Prototyping (WP-3) because the design depends on the requirements.
- Development (WP-4) can only start once the design (WP-3) is finalized.
- Testing and Quality Assurance (WP-5) follows after development is mostly done to ensure everything works smoothly.
- We proceed to Deployment and Release (WP-7) only after thorough testing confirms the system is ready.
- Finally, Monitoring and Support (WP-8) begins after deployment and continues as long as the system is live.

5.3 Resource Requirements

The estimated resource requirements for the "MediSlot" project include:

- Personnel: A cross-functional team of software developers, designers, quality assurance specialists, and project managers.
- Computer hardware and software: Development and testing environments, servers, and software tools.
- Office and laboratory facilities: Workspace for the project team.
- Maintenance requirements: Ongoing support and maintenance resources.

5.4 Budget and Resource Allocation

Budget and resource allocation will be determined based on the specific resource requirements for each work package. A detailed budget and resource allocation plan will be established to ensure that each work package receives the necessary resources.

5.5 Schedule

For MediSlot, we've planned out a 12-week timeline that keeps everything moving smoothly while respecting the order in which tasks need to happen. Here's how it breaks down:

Weeks 1-2: Project Kickoff and Requirements Gathering
We'll start by setting clear goals, defining who does what, and gathering all the needs from patients, doctors, and admins to make sure we build exactly what's required.

Weeks 3-4: Design and Prototyping
Next, we'll work on designing the user interface and the system architecture. We'll also create early prototypes to get feedback and make sure we're on the right track.

Weeks 5-8: Development
This is the core building phase where we write the code for both the frontend and backend, including features like booking appointments, notifications, and dashboards.

Weeks 9-10: Testing and Quality Assurance
Once development is mostly done, we'll thoroughly test the system to catch bugs, fix issues, and make sure everything works smoothly.

Weeks 10-11: Documentation and User Guides
While testing wraps up, we'll prepare easy-to-understand user manuals, help guides, and tutorials so users can get started quickly.

Week 12: Deployment and Launch
Finally, we'll deploy MediSlot to the live environment and make it available for patients,

doctors, and admins to use. From here on, we'll keep monitoring the system and offer support as needed.

6. Additional Components.

6.1 Index.

- **SRS - Software Requirements Specification**
- **SDD - Software Design Description**
- **QA - Quality Assurance**
- **PWA - Progressive Web Application**
- **API - Application Programming Interface**
- **UI/UX - User Interface/User Experience**

6.2 Appendices

Appendices may be included, either directly or by reference, to provide supporting details that could detract from the SPMP if included in the body of the SPMP. Suggested appendices include:

A. Current Top 10 Risk Chart

The main risks include: delays in getting healthcare provider schedules, third-party notification service failures, critical bugs in booking workflow, low early adoption, unavailability of key team members, server downtime, payment gateway integration issues, security breaches, regulatory changes, and scope creep. Each risk has assigned likelihood and impact ratings, with mitigation strategies such as early agreements, fallback providers, robust testing, cross-training, secure hosting, encryption, and a strict change-control process.

B. Current Project Work Breakdown Structure

The project is divided into eight phases: initiation, requirements gathering, design/prototyping, development, testing/QA, documentation, deployment, and support/maintenance. Each phase contains clearly defined tasks, deliverables, and dependencies to ensure orderly progress.

C. Current Detailed Project Schedule

The 12-week plan is as follows:
Weeks 1–2 – kickoff and requirements;

Weeks 3–4 – design and prototyping;
Weeks 5–8 – development;
Weeks 9–10 – testing and QA;
Weeks 10–11 – documentation;
Week 12 – deployment and launch. Post-launch monitoring and support continue thereafter.

Conclusion:

In this project, we've laid out a clear roadmap to build MediSlot a simple and effective tool for scheduling medical appointments. We've broken the work into manageable parts, set realistic timelines, and made sure everyone knows what they need to do. By keeping a close eye on quality and staying flexible to feedback, we're confident we'll create a platform that's easy to use and meets the needs of patients, doctors, and clinics. With good teamwork and solid planning, MediSlot will be ready on time and ready to make healthcare scheduling hassle-free.

Post Lab Descriptive Questions

1. State various Scheduling principles and explain them in detail.

1. **Prioritization**
   This principle is about executing the most critical or time-sensitive tasks first. It ensures that high-impact jobs are completed on time, rather than getting delayed by trivial ones. Without prioritization, important processes may suffer while less significant ones consume resources.
2. **Fairness**
   Fairness ensures that every process or user gets an appropriate share of resources. It's about preventing any one task from dominating the system while others are left waiting. This is essential to avoid starvation, where some jobs never get executed.
3. **Efficiency**
   Efficiency means making the best use of available system resources. A good scheduler keeps the CPU and other hardware busy with useful work, minimizing idle time and maximizing performance
4. **Responsiveness**
   Responsiveness is critical in interactive systems. It ensures that the system reacts quickly to user input. If a user clicks something or types, the system should respond without noticeable delay. Poor responsiveness results in frustration and a bad user experience.
5. **Throughput**
   This refers to the number of processes completed in a given amount of time. High throughput means the system is productive and handling more tasks, which is ideal in environments with heavy workloads.

6. **Turnaround Time**
   Turnaround time is the total time taken from when a process is submitted until it's completed. Lower turnaround times are better, especially in batch systems where fast job completion is essential.
7. **Waiting Time**
   Waiting time is how long a process sits in the ready queue before being assigned CPU time. The goal is to minimize this, as long waits can lead to bottlenecks and lower overall performance.
8. **Deadlock Avoidance**
   This principle ensures that the scheduler doesn't allow processes to get into a state where they are waiting for each other indefinitely. It's about making sure resource allocation doesn't trap the system.
9. **Predictability**
   A predictable scheduler behaves consistently. If a job usually takes five seconds, it shouldn't suddenly take twenty without a good reason. Predictability helps in planning and system reliability.
10. **Scalability**
    Scalability means the scheduling system should perform well even as the workload or number of processes increases. A scheduler that works fine with ten tasks but crashes with a hundred is practically useless in real-world systems.