Rajalakshmi Engineering College

An AUTONOMOUS institution
Affiliated to ANNA UNIVERSITY, Chennai

MediTrack

LAB RECORD
CS19442-SOFTWARE ENGINEERING CONCEPTS

Submitted by

Naveen Kumar R	**220701183**
Neela A	**220701184**
Nidarshana K S	**220701185**
Nikilashree M	**220701186**
Nishal I P	**220701187**
Nithish Rao P	**220701188**

TABLE OF CONTENT

OVERVIEW OF THE PROJECT

Project Title: MediTrack

Problem statement:

Manual patient tracking processes in healthcare facilities are time consuming, error prone, and lack realtime updates. Healthcare providers face challenges in efficiently managing patient information, appointments, and medication tracking, leading to potential errors, delays, and communication gaps. There is a pressing need for a streamlined system that automates these tasks, enhances data accuracy, and facilitates realtime communication to improve overall healthcare efficiency and patient care.

Objective:

MediTrack aims to streamline patient information management, appointments, and medication tracking in healthcare facilities. Its objectives include automating administrative tasks to boost efficiency, ensuring the accuracy of patient records through digitization and error checking mechanisms, providing realtime updates for timely decisionmaking, engaging patients through secure access to their health records, and ultimately enhancing healthcare delivery by improving communication and quality of care.

Target Audience:

1. Healthcare Providers: Including doctors, nurses, and administrative staff involved in patient care, appointment scheduling, and medication management.
2. Patients: Individuals seeking healthcare services who can benefit from accessing their health records, scheduling appointments, and managing medications conveniently.
3. Pharmacists: Professionals involved in medication dispensing and management, who can benefit from real time updates and synchronization with patient records.
4. Healthcare IT Professionals: Specialists involved in system development, maintenance, and integration, ensuring the smooth operation and security of the patient tracking application.

Key Features:

1. Patient Registration and Management: Streamlined patient onboarding and comprehensive management of patient records.
2. Realtime Updates and Notifications: Instant updates and alerts on patient status, appointments, and medication schedules.
3. Medication Management: Electronic prescriptions, dosage tracking, and pharmacy updates for efficient medication handling.
4. Online Appointment Scheduling and Reminders: Simplified appointment booking with automated reminders to reduce no shows and scheduling conflicts.
5. Pharmacy Updation: Real Time synchronization with pharmacies for uptodate medication information and prescription management.

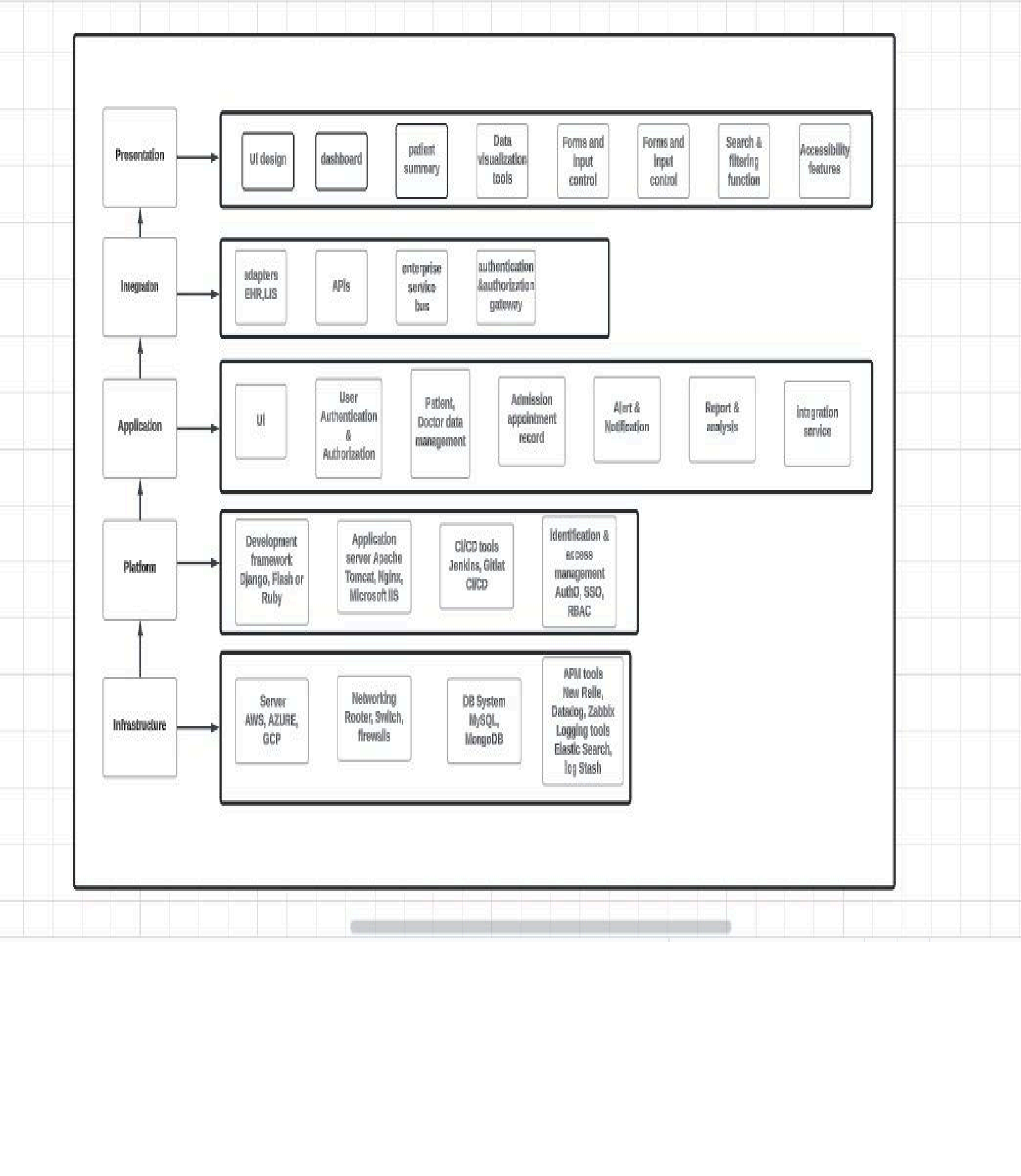
Goals:

1. Increase Efficiency: Automate administrative tasks to save time and reduce the workload for healthcare providers.
2. Enhance Accuracy: Ensure precise and reliable patient information through automated data entry and error checking mechanisms.
3. Improve Real Time Data Access: Provide real time updates and notifications to enable timely decision making and better coordination among healthcare teams.

Conclusion:

MediTrack, the patient tracker application is a transformative solution designed to address the inefficiencies and inaccuracies of manual patient tracking systems. By automating administrative tasks, ensuring data accuracy, and providing realtime updates, the application significantly enhances healthcare efficiency and quality. Key features such as medication management, online appointment scheduling, and secure patient portals empower both healthcare providers and patients, leading to improved communication, engagement, and care outcomes. With its comprehensive capabilities, MediTrack is poised to revolutionize patient management, optimize workflows, and support informed decisionmaking, ultimately contributing to a more effective and reliable healthcare system.

BUSINESS ARCHITECTURE DIAGRAM


Presentation
UI design
dashboard
patient summary
Data visualization tools
Forms and Input control
Forms and Input control
Search & filtering function
Accessibility features
Inegraion
adapters EHR,LIS
APIs
enterprise service bus
authentication &authorization gateway
Application
UI
User Authentication & Authorization
Patient, Doctor data management
Admission appointment record
Alert & Notification
Report & analysis
integration service
Platform
Development framework Django, Flash or Ruby
Application server Apache Tomcat, Nginx, Microsoft IIS
CI/CD tools Jenkins, Gitlat CI/CD
Identification & access management Auth0, SSO, RBAC
Infrastructure
Server AWS, AZURE, GCP
Networking Rooter, Switch, firewalls
DB System MySQL, MongoDB
APM tools New Relle, Datadog, Zabbix Logging tools Elastic Search, log Stash

REQUIREMENTS AS USER STORIES

Functional Requirements:

1. Patient Registration:
    - User Story: As a healthcare provider, I want to register new patients easily, so that their information is accurately recorded in the system.
    - Acceptance Criteria: Given a new patient, when I enter their personal details and submit the form, then the system should create a new patient record.

2. Appointment Scheduling:
    - User Story: As a patient, I want to schedule an appointment online, so that I can choose a convenient time without calling the clinic.
    - Acceptance Criteria: The system must prevent doublebooking by checking slot availability in realtime.

3. Medication Management:
    - User Story: As a healthcare provider, I want to manage patient medications electronically, so that I can ensure accurate and uptodate records.
    - Acceptance Criteria: The system must track dosage and frequency for each medication entry.

4. Realtime Notifications:
    - User Story: As a healthcare provider, I want to receive notifications for critical updates, so that I can respond promptly to patient needs.
    - Acceptance Criteria: Given a critical update (e.g., emergency admission), when it occurs, then the system should send an immediate notification to all relevant staff.

5. Analytics and Reporting:
    - User Story: As a healthcare administrator, I want to generate reports on patient data and operational efficiency, so that I can make informed decisions.
    - Acceptance Criteria: Given the analytics dashboard, when I select a reporting period and parameters, then the system should generate and display the report within 30 seconds.

NonFunctional Requirements:

1. Data Security:
    - User Story: As a patient, I want my health records to be securely stored, so that my personal information is protected.
    - Acceptance Criteria: The system must use encryption to protect data at rest and in transit.

2. System Performance:
    - User Story: As a healthcare provider, I want the system to respond quickly, so that I can access patient information without delay.
    - Acceptance Criteria: Given a user action (e.g., data entry, search), when it is performed, then the system should respond within 2 seconds.

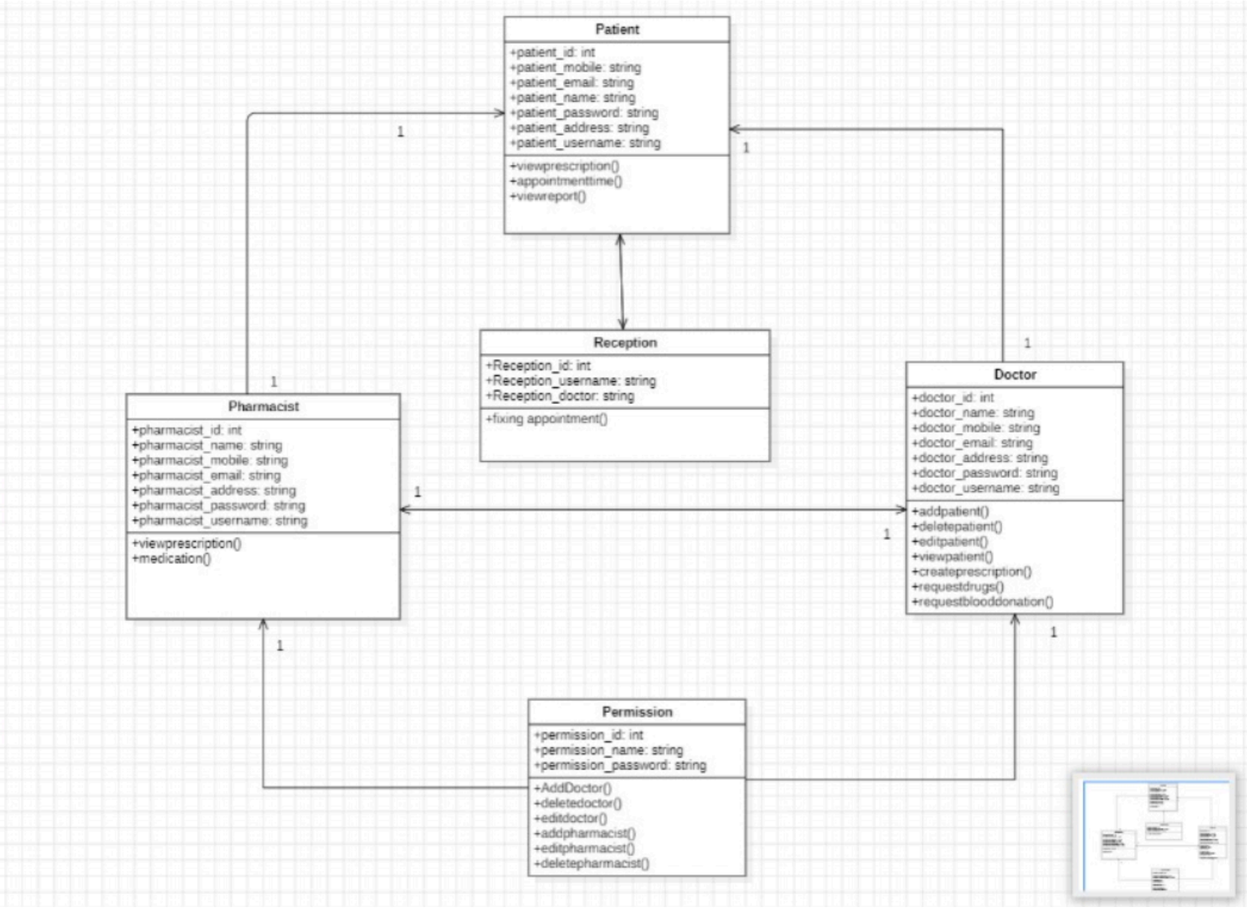
3. Usability:
    - User Story: As a healthcare provider, I want the system to be userfriendly, so that I can navigate and use it efficiently.
    - Acceptance Criteria: Given a new user, when they start using the system, then they should be able to complete basic tasks (e.g., patient registration, appointment scheduling) without requiring extensive training.

4. Reliability:
    - User Story: As a healthcare administrator, I want the system to be reliable, so that it is available whenever needed.
    - Acceptance Criteria: The system must automatically back up data every hour and support full restoration within 30 minutes.

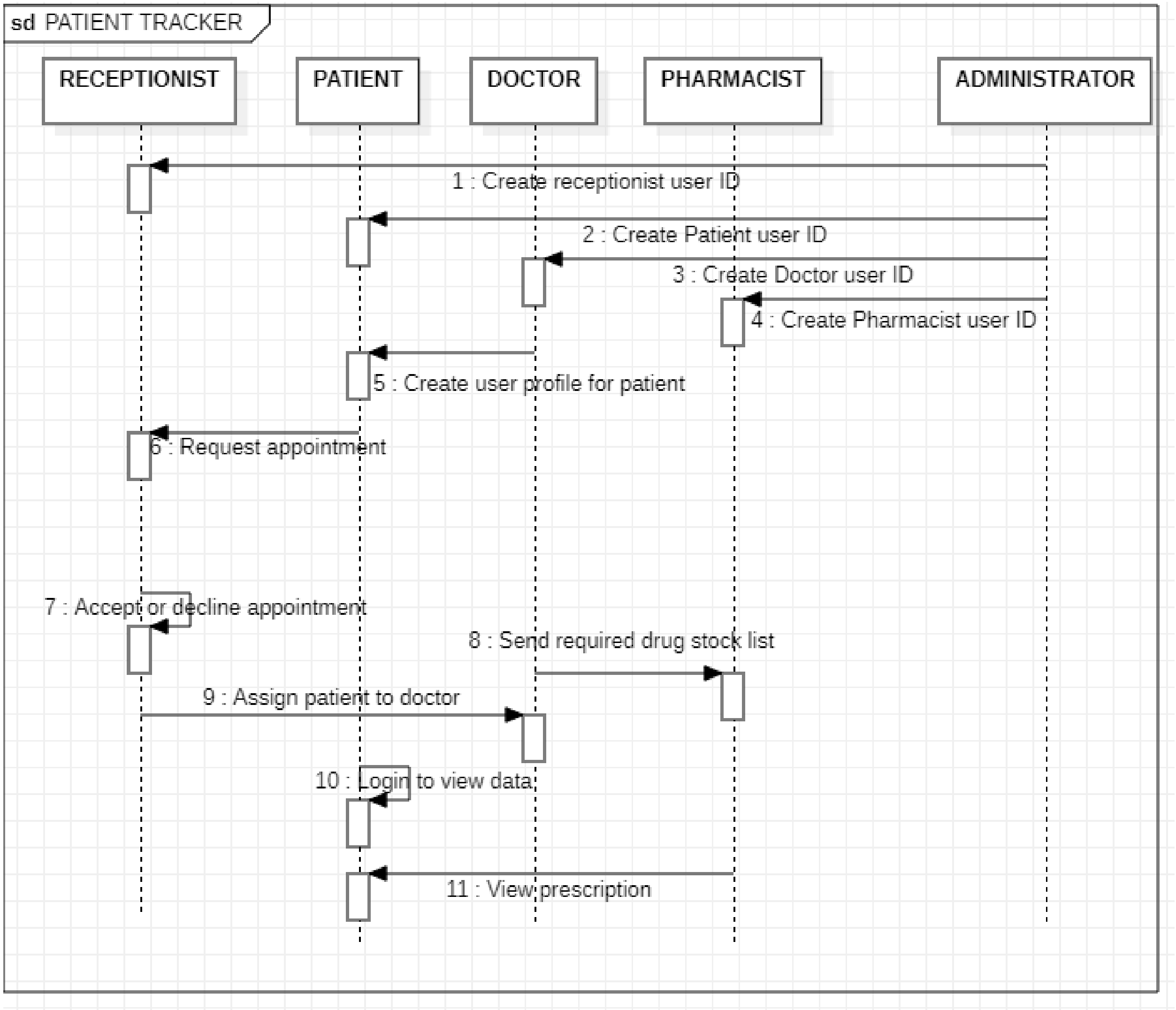
5. Scalability:
    - User Story: As a healthcare provider, I want the system to scale with the growth of the healthcare facility, so that it can handle increasing amounts of data and users.
    - Acceptance Criteria: The system must support adding new users and handling increased data loads without requiring significant downtime or performance loss.

UML DIAGRAM

Class diagram:



Sequence diagram:


sd PATIENT TRACKER
RECEPTIONIST
PATIENT
DOCTOR
PHARMACIST
ADMINISTRATOR
1 : Create receptionist user ID
2 : Create Patient user ID
3 : Create Doctor user ID
4 : Create Pharmacist user ID
5 : Create user profile for patient
6 : Request appointment
7 : Accept or decline appointment
8 : Send required drug stock list
9 : Assign patient to doctor
10 : Login to view data
11 : View prescription

TEST STRATEGY

This test plan outlines the approach, resources, and schedule for testing the "MediTrack" application. The objective is to verify the reliability, usability, security, performance, and realtime capabilities of the system.

Objective:

- Ensure System Reliability
- Enhance Usability
- Verify Security
- Assess Performance
- Validate Realtime Capabilities

Scope:

The testing scope includes functional and nonfunctional aspects of the "MediTrack" application, covering patient registration, appointment scheduling, medication management, data security, usability, performance, and realtime updates.

Test Strategy:

- Functional Testing: Validate core functionalities.
- NonFunctional Testing: Assess usability, security, performance, and realtime capabilities.
- Automated Testing: Implement automated scripts for regression and performance testing.
- Manual Testing: Conduct exploratory and usability testing manually.

Test Environment:

- Hardware: Servers, test devices (desktops, tablets, smartphones).
- Software: Browsers (Chrome, Firefox, Safari, Edge), operating systems.
- Tools: Selenium, JMeter, OWASP ZAP, JIRA.

Test Cases:

1. User Authentication:

   Happy Path

   - Valid Login

     Description: Verify that a user can log in with valid credentials.

     Steps: 1. Open the login page.
     2. Enter a valid username and password.
     3. Click on the login button.

     Expected Result: User is redirected to the dashboard.

   Error Scenario

   - Invalid Login

     Description: Verify that the system prevents login with invalid credentials.

     Steps: 1. Open the login page.
     2. Enter an invalid username and password.
     3. Click on the login button.

     Expected Result: User receives an error message indicating invalid credentials.

2. Patient Registration

   Happy Path

   Add new patient

   Description: Verify that a new patient can be successfully added.

   Steps: 1. Navigate to the patient registration page.
   2. Enter all required patient details (name, DOB, address, contact info, etc.).
   3. Click the 'Submit' button.

   Expected Result: New patient record is created and visible in the patient list.

   Error Scenario

   - Missing Required Fields

     Description: Verify that the system prompts for missing required fields.

     Steps: 1. Navigate to the patient registration page.
     2. Leave one or more required fields empty.
     3. Click the 'Submit' button.

     Expected Result: The system displays an error message indicating which fields are required.

3. Patient Search:

   Happy Path

   - Search by name

     Description: Verify that a patient can be found by searching for their name.

     Steps: 1. Navigate to the patient search page.
     2. Enter the patient’s name in the search bar.
     3. Click the 'Search' button.

     Expected Result: The patient with the entered name appears in the search results.

   Error Scenario

   No Results Found

   Description: Verify that the system handles searches with no matching results.

   Steps: 1. Navigate to the patient search page.
   2. Enter a nonexistent patient's name in the search bar.
   3. Click the 'Search' button.

   Expected Result: The system displays a message indicating that no results found.

4. Appointment Scheduling:

   Happy Path

   - Schedule New Appointment

     Description: Verify that a new appointment can be scheduled.

     Steps: 1. Navigate to the appointment scheduling page.
     2. Select a patient.
     3. Choose a date and time for the appointment.
     4. Click the 'Schedule' button.

     Expected Result: Appointment is successfully created and appears in the patient’s appointment list.

   Error Scenario

   Conflict Detection

   Description: Verify that the system detects scheduling conflicts.

   Steps: 1. Try to schedule an appointment at a time that is already booked.

   Expected Result: The system displays an error message indicating a scheduling conflict and prevents the appointment from being created.

5. Patient History and Records:
   Happy Path
   - View Patient History
     Description: Verify that a user can view a patient’s medical history.
     Steps: 1. Navigate to a patient’s profile page.
     2. Click on the 'History' or 'Records' tab.
     Expected Result: The system displays the patient's past medical history, including previous visits, diagnoses, and treatments.

   Error Scenario
   - Add Medical Record with Missing Information.
     Description: Verify that the system prompts for missing required fields when adding a medical record.
     Steps: 1. Navigate to a patient’s profile page.
     2. Click on the 'Add Record' button.
     3. Enter incomplete details of the new medical record.
     4. Click 'Submit'.
     Expected Result: The system displays an error message indicating which fields are required.

Test Schedule:

- Requirement Analysis: 1 week
- Test Design: 2 weeks
- Environment Setup: 1 week
- Test Execution: 4 weeks
- Defect Resolution and Retesting: Concurrent with test execution
- Test Closure and Reporting: 1 week

Risks and Mitigation:

Identify potential risks (e.g., data breaches, performance bottlenecks) and develop strategies to mitigate them (e.g., security audits, performance optimization).

Test Reporting:

Provide detailed test reports and summaries, including daily/weekly status reports, test metrics, and a final test report summarizing testing activities, outcomes, and recommendations.

DEPLOYMENT ARCHITECTURE

To design a robust deployment architecture for MediTrack, we need to consider the various components and layers that will enable it to meet its objectives efficiently. Below is a detailed deployment architecture comprising multiple layers: Presentation, Application, Business Logic, Data, and Security.

1. Client Layer (Presentation Layer)
Web Application: For healthcare providers and administrative staff to manage patient information, appointments, and medication tracking.
Mobile Application: For patients to access their health records and appointments.

2. API Gateway
Manages all client requests and directs them to appropriate microservices.

3. Microservices Layer (Application Layer)
User Management Service: Manages user authentication, roles, and permissions.
Patient Information Service: Manages patient records, including personal information and medical history.
Appointment Service: Handles scheduling, reminders, and calendar management.
Medication Tracking Service: Manages medication prescriptions, refills, and tracking.
Notification Service: Sends realtime updates and reminders via email/SMS/push notifications.
Reporting and Analytics Service: Provides data analysis and reporting for decision making.

4. Business Logic Layer
Rules Engine: Implements healthcare specific business rules and decision making logic.

5. Data Layer
Database: Centralized storage for all structured data.
NoSQL Database: For unstructured data such as logs, patient notes.
Data Warehouse: For storing historical data and performing complex queries.
Cache: For improving read performance of frequently accessed data.

6. Security Layer
Authentication and Authorization: Ensures only authorized access to the system.
Encryption: Protects data at rest and in transit.
Intrusion Detection and Prevention System (IDPS): Monitors network traffic for suspicious activities.
Security Information and Event Management (SIEM): Collects and analyzes security data.