Author Name: Mayur Salve.

1. Introduction

Objective:
To ensure the system (Login Interface, Patient Dashboard, and Add Patient Form) meets functional, security, usability, and compatibility requirements through structured testing.

Document Purpose:
Define the scope, approach, resources, and schedule for testing activities. Identify risks, tools, and deliverables to ensure thorough validation.

Stakeholders:

- Product Manager
- Development Team
- QA Team
- Security Team
- End Users

---

2. Test Objectives

1. Validate login functionality (authentication, session management).
2. Ensure accurate patient data entry, storage, and display.
3. Verify security against vulnerabilities (SQLi, XSS).
4. Confirm cross-browser/mobile compatibility.
5. Assess usability and accessibility.

---

3. Scope

In-Scope:

- **Functional Testing**: Login, Patient CRUD operations, data validation.
- **Security Testing**: Input sanitization, session security.

- **Compatibility Testing**: Chrome, Firefox, Safari, iOS/Android.
- **Usability Testing**: UI responsiveness, error messages.

Out-of-Scope:

- Performance/Load Testing.
- Third-party integrations (EHR systems).
- Database infrastructure testing (unless directly impacted by frontend).

---

4. Assumptions

1. Backend APIs are fully functional and documented.
2. Test environment mirrors production (hardware, OS, software versions).
3. Test data (e.g., patient records) will be provided or generated.
4. All stakeholders will participate in defect triage.

---

5. Tools & Frameworks

Category	Tools	Purpose
Test Automation	Selenium, Cypress	UI testing for forms and workflows.
API Testing	Postman, REST Assured	Validate backend endpoints.
Test Management	Jira, TestRail	Test case tracking, defect management.
Security	OWASP ZAP, Burp Suite	Vulnerability scanning (SQLi, XSS).
Accessibility	Axe, WAVE	WCAG 2.1 compliance.
Reporting	Confluence, Excel	Test summaries and traceability matrices.

Frameworks:

- **BDD**: Cucumber with Gherkin for scenario-based testing.
- **Data-Driven Testing**: Parameterized inputs (e.g., MRN, dates).

---

6. Test Methodology

1. Requirement-Based Testing:

- Map test cases to functional requirements (e.g., "User must log in to access dashboard").
- Example: Validate MRN uniqueness (Requirement ID: REQ-007).

2. Risk-Based Testing:

- Prioritize high-risk areas:
    - **Login Security**: Password masking, brute-force protection.
    - **Data Integrity**: MRN uniqueness, date/time validation.

3. Boundary Value Analysis (BVA):

- Test input limits:
    - MRN: 1–10 digits.
    - Date of Birth: Valid ranges (e.g., 1900-01-01 to current date).

4. Exploratory Testing:

- Ad-hoc testing for usability (e.g., button alignment, error message clarity).

5. Regression Testing:

- Automate critical test cases post-bug fixes.

ṇṇṇṇṇ7. Risk Assessment & Mitigation

Risk ID	Risk Description	Impact	Likelihood	Mitigation	Priority
RISK-01	SQL Injection in Login	Critical	Medium	Input sanitization, parameterized	High

Risk ID	Risk Description	Impact	Likelihood	Mitigation	Priority
				queries.	
RISK-02	Duplicate MRN Entries	High	High	Database constraints, UI validation.	High
RISK-03	Date Format Inconsistencies	Medium	High	Standardize formats (ISO 8601).	Medium
RISK-04	Timezone Ambiguity	Low	Medium	Display UTC offsets (e.g., EST: UTC-5).	Low
RISK-05	Mobile Responsiveness	Medium	Medium	Use Bootstrap/CSS media queries.	Medium

---

8. Entry & Exit Criteria

Entry Criteria:

- Test environment ready.
- Test cases reviewed and approved.
- Test data prepared.

Exit Criteria:

- All critical defects resolved.
- 100% test coverage for high-priority cases.
- Stakeholder sign-off on test summary.

---

9. Test Deliverables

1. **Test Cases**: Manual and automated scripts.

2. **Defect Reports**: Severity, priority, status (Jira).
3. **Traceability Matrix**: Requirements vs. test cases.
4. **Test Summary Report**: Metrics (pass/fail rates), lessons learned.

---

10. Timeline & Resource Allocation

Phase	Duration	Resources
Test Planning	3 Days	Test Lead, QA Manager
Test Design	5 Days	QA Engineers, Business Analyst
Test Execution	10 Days	QA Engineers, Automation Team
Defect Triage	2 Days	Dev Team, Product Owner
Reporting	2 Days	Test Lead

---

11. Defect Management

- **Logging**: Jira tickets with steps to reproduce, screenshots.
- **Priority**:
    - **Critical**: Blocks testing (e.g., login failure).
    - **High**: Major functionality broken (e.g., MRN duplication).
    - **Medium**: UI/Usability issues.
    - **Low**: Cosmetic flaws.

---

12. Training Requirements

- QA Team: Selenium/Cypress workshops.

- Security Team: OWASP ZAP training.
- Stakeholders: Demo sessions for UAT.

---

13. Glossary

- **MRN**: Medical Record Number.
- **DoB**: Date of Birth.
- **UTC**: Coordinated Universal Time.

---

14. Appendices

- **Appendix A**: Requirement Specifications.
- **Appendix B**: Risk Assessment Matrix.
- **Appendix C**: Test Environment Setup Guide.