**Appointment Booking System Testing Plan**

**1. Introduction**

The purpose of this document is to outline the testing strategy, scope, objectives, approach, and resources required for testing the Appointment Booking System. The system is designed to allow users to schedule appointments with doctors, view available slots, and manage bookings.

**2. Test Objectives**

* Verify that the system meets all functional requirements.
* Ensure that the system is user-friendly and performs efficiently under various conditions.
* Identify any defects or potential improvements.
* Validate the security of the system to prevent unauthorized access.

**3. Scope of Testing**

* **In-Scope:**
  + Functional testing of appointment booking, editing, and cancellation.
  + User authentication and role-based access (patients, doctors, admins).
  + Performance testing under different user loads.
  + Security testing for vulnerabilities.
  + Usability testing to assess the user experience.
* **Out-of-Scope:**
  + Integration with third-party services that are not part of the internal system (e.g., external payment gateways).

**4. Testing Approach**

The testing approach will include manual and automated testing methods to cover functional, non-functional, and security testing.

* **Manual Testing:** Conducted to ensure basic functionality works as expected across different user roles.
* **Automated Testing:** Use Selenium for UI automation and JMeter for performance testing.
* **Regression Testing:** Performed after each release to ensure that new changes do not affect existing functionalities.
* **Exploratory Testing:** Used to identify potential weaknesses by simulating real-world scenarios.

**5. Test Plan**

* **Environment:**
  + Testing will be conducted on development, staging, and production environments.
  + Supported browsers include Chrome, Firefox, and Safari. Mobile testing will be conducted on iOS and Android devices.
* **Tools:**
  + Selenium for UI Automation
  + JMeter for Performance Testing
  + Postman for API Testing
  + OWASP ZAP for Security Testing
* **Roles and Responsibilities:**
  + **Test Lead:** Oversees the entire testing process.
  + **QA Engineers:** Execute test cases and report defects.
  + **Developers:** Fix identified defects.

**6. Test Cases**

Here are sample test cases covering various scenarios.

| **Test Case ID** | **Description** | **Steps** | **Expected Result** | **Priority** |
| --- | --- | --- | --- | --- |
| TC001 | Verify user login functionality | 1. Navigate to login page  2. Enter valid credentials  3. Click 'Login' | User should successfully log in and be redirected to the dashboard | High |
| TC002 | Verify appointment creation | 1. Log in as a patient  2. Click 'Create Appointment'  3. Select date and time, choose doctor  4. Confirm booking | Appointment should be created, and a confirmation message displayed | High |
| TC003 | Verify cancellation of an appointment | 1. Log in  2. Go to 'My Appointments'  3. Cancel an existing appointment | Appointment should be successfully canceled, and the status updated | Medium |
| TC004 | Test appointment overlap prevention | 1. Schedule an appointment  2. Try booking another appointment for the same time | System should prevent booking and display an error message | High |
| TC005 | Assess system behavior under high load | 1. Simulate 1000 users booking appointments concurrently | System should handle the load without performance degradation | High |
| TC006 | Test SQL injection vulnerability on login form | 1. Enter malicious SQL code in username/password fields | System should prevent injection attempts and display an error | Critical |

**7. Defect Management**

* **Defect Tracking:** Defects will be logged and tracked using JIRA. Each defect will include:
  + **Title:** Short description of the defect.
  + **Severity:** Critical, High, Medium, Low.
  + **Steps to Reproduce:** How to replicate the defect.
  + **Expected vs. Actual Result:** Comparison of the intended and actual outcomes.
* **Defect Lifecycle:**
  + **New:** Defect is reported.
  + **Assigned:** Assigned to a developer.
  + **In Progress:** Being investigated or fixed.
  + **Resolved:** Fix is complete.
  + **Closed:** Verified and confirmed as fixed.

**8. Regression Testing Strategy**

* Regression tests will be conducted after bug fixes or feature updates.
* Test cases related to the modified areas will be prioritized.
* Automated regression suites will be executed to minimize manual effort.

**9. Performance Testing**

* **Objectives:** Ensure that the system remains responsive under various loads.
* **Tools:** JMeter will be used to simulate concurrent user activity.
* **Scenarios:**
  + Test with normal load (100 users).
  + Test with peak load (1000+ users).
  + Measure response times and server resource usage.

**10. Security Testing**

* **Objectives:** Identify vulnerabilities and ensure data protection.
* **Tools:** OWASP ZAP will be used for penetration testing.
* **Scenarios:**
  + Test for SQL injection, XSS, and CSRF vulnerabilities.
  + Ensure role-based access control functions as expected.

**11. Usability Testing**

* **Goals:** Ensure the system is user-friendly.
* **Participants:** Select a group of real users (patients and doctors) to perform common tasks.
* **Feedback Collection:** Record user feedback and identify potential improvements.

**12. Test Schedule**

| **Activity** | **Duration** | **Responsible** |
| --- | --- | --- |
| Test Planning | 1 week | Test Lead |
| Test Case Development | 2 weeks | QA Engineers |
| Test Execution | 3 weeks | QA Engineers |
| Defect Fixing & Retesting | 2 weeks | Developers, QA Engineers |
| Regression Testing | Ongoing | QA Engineers |

**13. Risks and Mitigation**

* **Risk:** Lack of testing resources may delay testing timelines.
  + **Mitigation:** Prioritize critical functionalities and automate repetitive tests.
* **Risk:** Unavailability of test environments.
  + **Mitigation:** Coordinate with DevOps to ensure environment availability in advance.

**14. Exit Criteria**

* All high-severity defects have been resolved.
* Regression tests have passed.
* The system meets performance benchmarks.

**15. Sign-Off**

* Testing will be signed off once exit criteria are met, and stakeholders agree on the quality level.