Toothbook.ca Test Design Document

Authors:

* Arifali Baghirli - 210201848
* Zoher Matar - 210201919
* Yahye Said Abdi - 210201922
* Bashir Tayani - 220201812
* Abdulbari Abdulghani Ahmed Leftesi - 220201877
* Ramiz Shafiyev - 220201988
* Bashar Ali Mohamed Osman - 220201863
* Ahmed Mohamed Ahmed – 200201874

Table of Contents:

**1. Introduction**

* Purpose of testing
* Target audience

**2. Test Plan**

* Testing types: Unit, Integration, System, UAT
* Tools: xUnit, Postman, Swagger,Selenium
* Test environment setup

**3. Test Scenarios and Cases**

* Table of test cases
  + ID
  + Description
  + Input
  + Expected Output
  + Actual Output
  + Status

**4. Test Data**

* Example values for patients, appointments, logins, etc.

**5. Bug Tracking and Reporting**

* Example bug report template
* Severity levels

**6. Summary and Acceptance Criteria**

* When is the system considered tested and complete?

**1. Introduction**

**1.1 Purpose of Testing**

The purpose of this testing phase is to ensure that the Toothbook Online Clinic Appointment System functions correctly, reliably, and securely across all user modules (Patient, Clinic, Admin). Testing will validate both functional and non-functional requirements and help detect bugs early in development.

**1.2 Target Audience**

* Developers
* QA Engineers
* Project Stakeholders
* System Administrators
* End-users (during UAT)

**2. Test Plan**

**2.1 Testing Types**

|  |  |
| --- | --- |
| **Type** | **Description** |
| Unit Testing | Verifies the logic of individual functions and components (e.g., patient registration logic) using **xUnit**. |
| Integration Testing | Ensures modules (e.g., booking and notification) interact correctly. Tools: **Postman**, **Swagger** |
| System Testing | Validates the whole application flow (end-to-end). |
| UAT (User Acceptance Testing) | Final stage with selected users to validate business requirements. |

**2.2 Tools**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| **xUnit** | Backend unit tests (.NET) |
| **Postman** | API testing |
| **Swagger** | API documentation + testing |
| **Selenium** | Frontend automated UI testing |
| **TestRail / Excel** | Test case documentation |

**2.3 Test Environment Setup**

* **Backend:** .NET Core Web API (running on Docker)
* **Frontend:** React TypeScript
* **Database:** MSSQL
* **Test users:** Seeded test data in DB
* **Email/Telegram mock services** for notification testing

**3. Test Scenarios and Cases**

**3.1 Sample Table of Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Description** | **Input** | **Expected Output** | **Actual Output** |
| TC01 | Register new patient | Name, Email, Password | User created, JWT returned | As expected |
| TC02 | Login with wrong password | Valid email, wrong password | Unauthorized (401) | As expected |
| TC03 | Book appointment | Valid clinicId, datetime | Appointment ID returned, confirmation sent | As expected |
| TC04 | Double booking check | Same patient, same datetime | Error: "Already booked for this time" | As expected |
| TC05 | View clinic reviews | Clinic ID | List of reviews | As expected |
| TC06 | Admin deletes clinic | Clinic ID | Success, clinic marked inactive | Not implemented |

**4. Test Data**

|  |  |
| --- | --- |
| **Entity** | **Sample Data** |
| **Patient** | ID: p001, Name: Arif Ali, Email: arif@test.com, Password: 12345678 |
| **Clinic** | ID: c001, Name: HighMedical, Address:Jasper, Specialties: Braces, Whitening |
| **Appointment** | ID: a001, PatientID: p001, ClinicID: c001, Time: 2025-06-01T10:00 |
| **Admin** | ID: a001, Email: admin@test.com, Role: SuperAdmin |

**5. Bug Tracking and Reporting**

**5.1 Bug Report Template**

|  |  |
| --- | --- |
| **Field** | **Example** |
| Bug ID | BUG-0001 |
| Title | "Appointment booking fails with 500" |
| Description | Internal server error on booking flow |
| Steps to Reproduce | 1. Login → 2. Click book → 3. Submit |
| Severity | High |
| Status | Open/Fixed/Closed |
| Assigned To | Backend Dev |
| Reported By | QA |
| Timestamp | 2025-05-17 14:00 |

**5.2 Severity Levels**

|  |  |
| --- | --- |
| **Severity** | **Description** |
| Critical | App crashes, security vulnerability |
| High | Major functionality broken |
| Medium | Minor logic bug or incorrect UI data |
| Low | Cosmetic UI issue, spelling mistake |

**6. Summary and Acceptance Criteria**

**6.1 Completion Criteria**

The system is considered fully tested and ready for production when:

* All high/critical bugs are fixed
* Minimum 95% of test cases pass
* All modules (Patient, Clinic, Admin) are verified
* Security tests (auth, roles) pass
* UAT is signed off by stakeholders
* Test summary report is delivered