|  |
| --- |
| **A logo with a green and blue letter  AI-generated content may be incorrect.** |

**Children Care System**

**Test Plan**

**Project Code: CCS**

**Document Code: CCS\_Test Plan\_v1.0**

**Hanoi, Jul 2025**

**RECORD OF CHANGE**

\*A - Added M - Modified D - Deleted

|  |  |  |  |
| --- | --- | --- | --- |
| **Effective Date** | **A\* M, D** | **Change Description** | **New Version** |
| 07/06/25 | A | First version created | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1](#_heading=h.hbg1w2s9bh2k) Introduction 4

[1.1](#_heading=h.1vjnqkqobfpq) Purpose 4

[1.2](#_heading=h.jq7fki5fjlvr) Project Overview 4

[2](#_heading=h.lted1pn8w7d9) Scope 4

[2.1](#_heading=h.lya6wu3d1ph0) In-Scope 4

[2.2](#_heading=h.lqgdj9gbf70e) Out-of-Scope 5

[3](#_heading=h.1twavso1srpm) Testing Strategy 5

[3.1](#_heading=h.tzlc1srx85e6) Test Objectives 5

[3.2](#_heading=h.59ruherigvtf) Test Assumptions 5

[3.3](#_heading=h.ikjwwzaxa038) Data Approach 6

[3.4](#_heading=h.ruijy44j841a) Level of Testing 7

[3.5](#_heading=h.hsoiwfdjrrfg) Unit Testing 7

[3.6](#_heading=h.jzxftjsuyawx) Functional Testing 8

[3.7](#_heading=h.vjhfntkaxzad) User Acceptance Testing 9

[3.8](#_heading=h.qa5o8whmudjj) Regression Testing 9

[4](#_heading=h.j13zffitxk0u) Execution Strategy 10

[4.1](#_heading=h.llxoxnhadylp) Entry Criteria 10

[4.2](#_heading=h.f9za3omlanbc) Exit criteria 10

[4.3](#_heading=h.yuqyelircd1q) Validation and Defect Management 10

[5](#_heading=h.ktzre5q4h7dk) Environment Requirements 10

[5.1](#_heading=h.u6gwanhbq8i5) Test Environments 10

[6](#_heading=h.n9971y7cyfch) Significantly Impacted Division/College/Department 10

[7](#_heading=h.akvorid2hlrt) Dependencies 10

# Introduction

## Purpose

This Test Plan describes the testing approach, scope, objectives, and criteria for the Children Care System (CCS). Its purpose is to ensure that all functional and quality requirements of CCS are validated, defects are identified and fixed, and risks are mitigated before release.

## Project Overview

Children Care System (CCS) is a web-based application for managing pediatric healthcare services. Major features include:

* **Customer Portal:** user registration, profile management, service selection, appointment booking, feedback submission
* **Staff Interface:** view and update patient appointments, record examination results
* **Manager Dashboard:** approve or reject bookings, assign staff, generate reports
* **Admin Console:** manage user accounts, configure system settings

# Scope

## In-Scope

The testing scope of this Test Plan covers all core functions and business processes of the **Children Care System (CCS)**. The system supports multiple roles including Guest, Customer, Staff, Manager, and Admin. The following features, workflows, and interfaces will be tested:

* **User account management**: including user registration, login, logout, password reset, and profile update.
* **Service browsing and selection**: display of services and detailed service information for customers.
* **Appointment booking process**: full workflow from selecting a service and schedule, to submitting and confirming bookings.
* **Feedback workflow**: allowing customers to submit feedback for completed appointments.
* **Medical record update**: staff members update examination results and finalize appointments.
* **Reservation management**: managers approve or reject appointments and manage customer bookings.
* **Content and configuration management**: managers manage services, blog posts, sliders, customer accounts, and feedback.
* **Administrative functionalities**: system administrators manage user accounts and system settings.
* **Role-based access control (RBAC)**: ensuring that users only access permitted functionalities based on their roles.
* **Basic UI validation**: input validation, navigation, and form completeness across screens.
* **Internal system workflows and interfaces** between booking, user, and feedback modules.

All internal modules, pages, and data flows related to the features above will be included in testing. The CCS is a standalone web application and does not involve vendor integrations or third-party APIs beyond email notifications.

## Out-of-Scope

The following features will not be tested in this phase:

* Integration with external email services (e.g. account verification via email)
* Advanced security testing such as penetration tests
* Stress and load testing with high user concurrency
* Mobile responsiveness and cross-browser compatibility

These aspects are either assumed to be handled separately by developers or will be tested in later phases if necessary.

# Testing Strategy

## Test Objectives

The main objective of this testing effort is to ensure that all critical features of the Children Care System (CCS) function correctly and meet the specified requirements. The testing will focus on verifying functional correctness, usability, and access control across different user roles (Guest, Customer, Staff, Manager, Admin).

Specific objectives include:

* Ensure users can successfully register, log in, and book appointments.
* Verify that staff and managers can access and manage appointment workflows.
* Confirm that feedback and content management features operate as expected.
* Validate role-based access restrictions across all modules.

## Test Assumptions

The following assumptions are made for this test plan:

* The functional and non-functional requirements of the CCS are complete, clear, and approved by the stakeholders.
* The development team has completed and deployed all features to the test environment before testing begins.
* All required test data (e.g. test users, services, appointments) will be prepared and available prior to test execution.
* The testing will be performed on a stable version of the system with no critical unresolved bugs from previous phases.
* No major changes will be introduced to the system during the testing period.
* The test environment will reflect the intended production environment closely enough to identify significant issues.
* Team members have access to all necessary documentation and tools required for test design and execution.

## Data Approach

The testing team will prepare and maintain test data in the QA environment to support both functional and user acceptance testing activities. The data will reflect realistic usage scenarios for all user roles in the Children Care System (CCS), including:

* **Customer test accounts** with varying profiles (e.g., with/without bookings, new/returning users)
* **Sample services** across different categories to test browsing and booking functionality
* **Predefined appointment slots** to test slot availability, booking confirmation, and staff assignment
* **Feedback entries** to verify moderation and display behavior
* **Staff and Manager accounts** with appropriate permissions for testing workflows
* **Admin account** for testing access to system configuration features

Test data will be created manually or via pre-seeded scripts and stored in a controlled QA database. Any changes to test data will be documented and communicated across the team to avoid conflicts or inconsistencies during execution.

## Level of Testing

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| Unit Testing | Testing individual modules and components of the system, such as registration, login, and feedback. | Vũ Đức Đại – HE171816 |
| Functional Testing | Verifying system functionalities against the requirements using black-box testing techniques. | Đỗ Thị Thúy Anh – HE190984 |
| User Acceptance Testing | Validating that CCS meets the business needs through real-world use scenarios. | Trịnh Tuấn Anh – HE176749 |
| Regression Testing | Re-testing system features after changes to ensure no new bugs are introduced. |  |

## Unit Testing

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Vũ Đức Đại – HE171816 | Registration module: test username/password validation and email format check | Unit Tester |
| Vũ Đức Đại – HE171816 | Login module: test credential matching and session handling | Unit Tester |
| Vũ Đức Đại – HE171816 | Feedback module: test rating input, content limit, and form submission logic | Unit Tester |

## Integration Testing

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Trịnh Tuấn Anh – HE176749 | Register → Login → Book Service: test data flow from account creation to booking | Integration Tester |
| Trịnh Tuấn Anh – HE176749 | Booking → Staff Confirmation: test link between reservation and staff view modules | Integration Tester |
| Trịnh Tuấn Anh – HE176749 | Feedback Submission → Manager Approval: test feedback visibility and status updates | Integration Tester |

## User Acceptance Testing

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Đỗ Thị Thúy Anh – HE190984 | End-to-end flow: Customer registers → logs in → books service → gives feedback | System Tester |
| Đỗ Thị Thúy Anh – HE190984 | Manager approves reservation → staff updates result → customer checks status | System Tester |
| Đỗ Thị Thúy Anh – HE190984 | Admin manages accounts and system settings | System Tester |

## Acceptance Testing

Participants:

|  |  |  |
| --- | --- | --- |
| **Tester’s Name** | **Department/ Area** | **Role** |
| Trịnh Tuấn Anh – HE176749 | Simulate customer usage: register, login, book service, submit feedback | Acceptance Tester |
| Đỗ Thị Thúy Anh – HE190984 | Simulate admin/manager workflows: approve bookings, manage content, update settings | Acceptance Tester |
| Vũ Đức Đại – HE171816 | Review staff functions: view reservations, update examination results | Acceptance Tester |

# Execution Strategy

## Entry Criteria

* *The entry criteria refer to the desirable conditions in order to start test execution*
* *Entry criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Entry Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *Test environment(s) is available* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *Test data is available* |  |  |  |
| *Code has been merged successfully* |  |  |  |
| *Development has completed unit testing* |  |  |  |
| *Test scripts are completed, reviewed and approved by the Project Team* |  |  |  |

## Exit criteria

* *The exit criteria are the desirable conditions that need to be met in order proceed with the implementation.*
* *Exit criteria are flexible benchmarks. If they are not met, the test team will assess the risk, identify mitigation actions and provide a recommendation.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Exit Criteria** | **Test Team** | **Technical Team** | **Notes** |
| *100% Test Scripts executed* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *90% pass rate of Test Scripts* |  |  |  |
| *No open Critical and High severity defects* |  |  |  |
| *All remaining defects are either cancelled or documented as Change Requests for a future release* |  |  |  |
| *All expected and actual results are captured and documented with the test script* |  |  |  |
| *All test metrics collected based on reports from daily and Weekly Status reports* |  |  |  |
| *All defects logged in Defect Tracker/Spreadsheet* |  |  |  |
| *Test environment cleanup completed and a new back up of the environment* |  |  |  |

## Validation and Defect Management

* *Specify how test cases/test scenarios should be validated*
* *Specify how defect should be managed*
  + *It is expected that the testers execute all the scripts in each of the cycles described above.*
  + *The defects will be tracked through Defect Tracker or Spreadsheet.*
  + *It is the responsibility of the tester to open the defects, retest and close the defect.*

Defects found during the Testing should be categorized as below:

|  |  |
| --- | --- |
| **Severity** | **Impact** |
| *1 (Critical)* | * *Functionality is blocked and no testing can proceed* * *Application/program/feature is unusable in the current state* |
| *2 (High)* | * *Functionality is not usable and there is no workaround but testing can proceed* |
| *3 (Medium)* | * *Functionality issues but there is workaround for achieving the desired functionality* |
| *4 (Low)* | * *Unclear error message or cosmetic error which has minimum impact on product use.* |

# Environment Requirements

## Test Environments

* *Specify the test environment(s) requirements*
* *Specify the security requirements.*

# Significantly Impacted Division/College/Department

|  |  |  |
| --- | --- | --- |
| **Business Area** | **Business Manager** | **Tester(s)** |
|  |  |  |
|  |  |  |
|  |  |  |

# Dependencies

*Identify any dependencies on testing, such as test-item availability, testing-resource availability, and deadlines.*