**Purpose of the Document**

Use the Test Plan document to describe the testing approach and overall framework that will drive the testing of the project.

Table of Contents

[1 Introduction 3](#_Toc515524393)

[1.1 Purpose 3](#_Toc515524394)

[1.2 Project Overview 3](#_Toc515524395)

[2 Scope 3](#_Toc515524396)

[2.1 In-Scope 3](#_Toc515524397)

[2.2 Out-of-Scope 3](#_Toc515524398)

[3 Testing Strategy 3](#_Toc515524399)

[3.1 Test Objectives 3](#_Toc515524400)

[3.2 Test Assumptions 3](#_Toc515524401)

[3.3 Data Approach 3](#_Toc515524402)

[3.4 Level of Testing 3](#_Toc515524403)

[3.5 Unit Testing 4](#_Toc515524404)

[3.6 Functional Testing 4](#_Toc515524405)

[3.7 User Acceptance Testing 4](#_Toc515524406)

[3.8 Regression Testing 5](#_Toc515524407)

[4 Execution Strategy 5](#_Toc515524408)

[4.1 Entry Criteria 5](#_Toc515524409)

[4.2 Exit criteria 6](#_Toc515524410)

[4.3 Validation and Defect Management 6](#_Toc515524411)

[5 Environment Requirements 7](#_Toc515524412)

[5.1 Test Environments 7](#_Toc515524413)

[6 Significantly Impacted Division/College/Department 7](#_Toc515524414)

[7 Dependencies 7](#_Toc515524415)

# Introduction

## Purpose

The purpose of this test plan is to describe how I will test the rental system to ensure it works correctly. This includes checking features like login, registration, and role management. I will also make sure it meets the needs of the users and doesn’t break during normal use.

## Project Overview

This project involves a rental system that allows users to register, log in, and manage their accounts. The goal is to ensure that all main functions, like user authentication and role-based access, work smoothly and are secure.

# Scope

## In-Scope

I will test the login and registration features to ensure users can access their accounts. Check that passwords are stored securely, and roles (like admin or regular users) are assigned correctly.

Test how the system interacts with the database to ensure data is saved and retrieved properly.

**Interfaces:**

1. ICar
   1. AddCarAsync, UpdateCarAsync, RemoveCarAsync, etc.
   2. GetAllCarsAsync, GetAllClientCarsAsync.
2. IUser
   1. SigInAsync, RegisterAsync.
   2. GetUserWithRoleByEmailAsync, UpdateUserWithRoleAsync.
3. IRole
   1. AddRoleAsync, GetAllRolesAsync.
4. IRental
   1. RentCarAsync, GetRentalsAsync, CancelRentalAsync, CloseRentalAsync.
   2. GetRentalsCustomerAsync
5. IDealer
   1. AddDealerAsync, UpdateDealerAsync, RemoveDealerAsync.
   2. GetAllDealersAsync, GetDealerAsync.
6. IWishlist
   1. AddWishListAsync, RemoveWishListAsync, GetWishesListWithCarsAsync.

**Processes:**

1. User Authentication Workflow
   1. Test user registration, login, and role-based access control.
2. Car Management Workflow
   1. Add a car, update its details, and remove it from the system.
3. Rental Workflow
   1. Rent a car, view rental history, cancel or close rentals.

## Out-of-Scope

Performance testing under extreme load conditions.

Advanced reporting features or analytics not related to the listed interfaces.

# Testing Strategy

## Test Objectives

The primary objectives of testing are to validate functionality, ensure quality, and identify issues before deployment.

**Objectives:**

1. Verify that all workflows (e.g., user registration, car rental) function as expected.
2. Ensure data integrity for user, car, and rental data.
3. Confirm role-based access control works for admin and regular users.
4. Identify, log, and resolve critical defects.

## Test Assumptions

## Data Approach

*Describe the approach on the test data maintained in QA environments for functional and user acceptance testing.*

## Level of Testing

*List the types of testing to be performed*.

|  |  |  |
| --- | --- | --- |
| **Test Type** | **Description** | **Responsible Parties** |
| Unit testing | Testing individual methods | Developer |

## Unit Testing

* Interfaces
* Repositories
* Functionalities like logging in, registration, renting a car.

## Functional Testing

* Testing the login and registration features.
* Verifying workflows for admin and regular users.
* Checking CRUD operations for cars, users, dealers and rentals.
* Testing specific error scenarios.

## Regression Testing

*Specify what features are to be tested.*

# 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* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *Code has been merged successfully* | C:\Users\arxp\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\7F9Z3IW4\MC900441310[1].png |  |  |
| *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.*