**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 2](#_Toc196220735)

[1.1 Purpose 2](#_Toc196220736)

[1.2 Project Overview 3](#_Toc196220737)

[2 Scope 3](#_Toc196220738)

[2.1 In-Scope 3](#_Toc196220739)

[2.2 Test Assumptions 3](#_Toc196220740)

[2.3 Functional Testing 3](#_Toc196220741)

[2.4 Regression Testing 3](#_Toc196220742)

[3 Dependencies 4](#_Toc196220743)

# 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

This testing scope focuses on verifying the core user flows and backend booking logic to ensure correct behavior, proper data flow, and consistent system state.

### Functional Features to Test

* **Login and Registration**  
  Ensure users can register and log in with valid credentials. Invalid inputs (e.g., wrong passwords or missing fields) should return proper error messages.
* **Role Assignment and Access**  
  Confirm that role-based logic behaves correctly: regular users see only their own data, while admins have broader access (if implemented).
* **Booking Creation and Validation**
  + Ensure users can book available seats only
  + Booking fails gracefully if the seat is already booked or time overlaps
  + Validation errors are shown if inputs are missing (seat not selected, invalid time)
* **Seat Availability**
  + Check that seats marked as unavailable do not appear in available lists
  + Ensure expired bookings release seats for future use
* **Seat Status Update Logic**
  + Trigger stored procedure/API to reset seat availability for expired bookings
  + Confirm seats become available again
* **Data Persistence**
  + Verify that data is properly saved in the database (e.g., new bookings, cancellations)
  + Booking records reflect correct user, seat, time, and duration

## Test Assumptions

* All necessary components (controllers, services, database) have been implemented and are accessible during testing.
* The application will be hosted locally or in a test environment that reflects the production structure.
* User data used during testing will be test-specific and will not affect real accounts.
* All stored procedures and database scripts have been applied before testing.
* Time-dependent features (e.g. booking expiration) can be triggered manually for test scenarios.

## Functional Testing

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

# Dependencies

* The database must be initialized and populated with valid test data (seats, users, zones).
* Stored procedures (e.g. UpdateExpiredBookingsSeats) must be available and working.
* Test accounts (admin and user roles) must be created in advance.
* Internet or local network connectivity.