**A PROPOSED OFFERING OF WEB-BASED RESERVATION SYSTEM FOR THE ADDLIB DANCE STUDIO AT DATAMEX COLLEGE OF SAINT ADELINE VALENZUELA BRANCH**

A Research Project Presented to the

Faculty of Datamex College of Saint Adeline, Inc.

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science in Information Technology

Bacani, Princess Mahallalel  
Delos Angeles, Arjay O.  
Mejila, Steven Carl A.  
Norte, Sakura Ann G.

**TESTING DOCUMENTATION**

# 

# **INTRODUCTION**

The web-based Addlib Studio Booking System is made to simplify booking and managing reservations at dance studios. Whether you are an Admin or a Client, the system is designed for a smooth and competent way of working with you. As Admin, you are responsible for managing class offers, booking requests, and system data. The Client can select a studio session that suits them, book classes and see which bookings are still incomplete. This documentation offers a full description of the testing process, including: evironment, methods, findings and conclusions. Introduction The aim of the testing phase is to see what works with the AddLib Studio Booking System in terms of stability and usabiliy. It also tries to make sure that all basic functions are as expected and that Admin and Client roles have correct roles operate with their proper permissions.

The objective of the testing to validate the functionality of all system modules,including user authentification,offer management and booking process and identify and document any issue or inconsistencies.also to confirm that admin and client user experiences,and to ensure the system’s performance and usability meet established quality standards

Testing AddLib Studio Booking System Documentation.A web-based tool called the AddLib Studio Booking System was developed to make booking and managing dance studio reservations easier. It makes it possible for both admin and client users to engage with the system effectively. While the client can explore available studio sessions, book lessons, and monitor booking statuses, the administrator is in charge of handling class offerings, booking requests, and system data. A through description of the testing procedure, including the setting, procedures, results, and suggestions, is provided in this publication.

**TESTING ENVIRONMENT**

Our system has been tested to ensure that the system works and is functionally working. This is what we used to perform testing on our system.

.

Hardware Specifications:

Laptop: Intel Core i7, 16 GB RAM, Windows 11, Chrome v127

Desktop: AMD Ryzen 5, 8 GB RAM, Ubuntu 24.04, Firefox v125

Software Requirements:

HTML, CSS, and JavaScript (Vanilla JS) for user interface and client-side logic.

Responsive design tested across different devices and browsers.

**Backend and Database:**

**Supabase (PostgreSQL) for user authentication, data storage, and real-time updates.**

**Testing Tools and Resources:**

**Chrome Manual brower testing**

**Test Data:**

**Admin account: HeadAdmin@gmail.com/12345**

**Client account: client@addlib.com / Client@123**

This environment ensured that both the frontend and backend were tested together for performance, functionality, and data integrity.

**TESTING METHODOLOGY**

# The testing process combined Black-box Testing and User Acceptance Testing (UAT) methods. Black-box testing was used to examine the system’s functional behavior without looking at internal code. It focused on verifying that each component performed correctly based on inputs and outputs. User Acceptance Testing was carried out by actual Admin and Client users, who performed everyday tasks such as logging in, booking classes, and managing offers.

The main criteria for testing success were:

1.Accurate login validation and error handling.

2..Proper enforcement of role-based access restrictions.

3.Smooth and accurate booking flow, including confirmation messages.

4..Quick system responses (ideally under three seconds for most actions).

5.Registered as a client and see the flow and interface of the system

6.No critical or high-severity bugs remaining before deployment.

# While testing, I found a problem where a client could add a schedule for a past date, which shouldn’t be possible. This could mess up the booking records and cause confusion. To fix this, I added a validation rule in the booking form using JavaScript. The rule checks the date before saving it to the database and only allows dates that are the same as or later than the current date. After adding this validation, I tested it again, and it worked perfectly clients could no longer choose past dates when booking.

# Overall, using both Black-box and User Acceptance Testing helped me see not only if the system worked as expected, but also how it felt to actually use it. It gave me a better understanding of what needed improvement, especially in terms of performance, usability, and the overall user experience.

**TEST CASES**

During the testing of the AddLib Studio Booking System, I created several test cases to check if the main features were working properly. I tested the system using both Admin and Client accounts to see how it behaved for different users. The testing helped me make sure that every feature worked as intended and also allowed me to find some issues that needed to be fixed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Expected Output** | **Actual Output** | **Status** | **Remarks** |
| **TC001** | Login using valid Admin credentials | Admin successfully logs in and is redirected to dashboard | Worked as expected | Pass | N/A |
| **TC002** | Login using invalid Client credentials | Error message “Invalid email or password” appears | No error message appeared | Fail | Fixed Later |
| **TC003** | View all available offers as Client | All available offers are displayed | Worked as expected | Pass | N/A |
| **TC004** | Book a “Dance Class” as Client | Booking is successfully saved and shown under “My Bookings” | Worked as expected | Pass | N/A |
| **TC005** | Approve a Client booking as Admin | Booking status updates to “Approved” | Worked as expected | Pass | N/A |
| **TC006** | Access Admin page as Client | Access is denied and redirected to Client dashboard | Worked as expected | Pass | N/A |
| **TC007** | Delete an offer as Admin | Offer is removed and no longer visible to clients | Worked as expected | Pass | N/A |
| **TC008** | Booking performance test | Booking confirmation loads within 3 seconds | Took around 1 seconds | Pass | N/A |
| **TC009** | Add a booking with a past date | System should not allow booking on past dates | System still allowed past booking | Failed | Issues found and fix |
| **TC010** | Decline button | Admin dashboard have declined button | No declined button but have delete button | pending | For improvement |

**BUG TRACKING AND ISSUES LOG**

During the testing phase of the Addlib Studio Booking System, all issues and errors found were carefully recorded and tracked. Each bug was described based on what caused it and how it affected the system. The bugs were then classified by severity to know which ones needed to be fixed first. This process helped me monitor the progress of each issue and make sure that the important ones were resolved before deployment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bug ID** | **Description** | **Severity** | **Status** | **Resolution** |
| **B001** | Client login fails to show error message on invalid credentials | High | Open | Resolved |
| **B002** | Booking confirmation takes more than 5 seconds | Medium | In Progress | Resolved |
| **B003** | Admin deletion does not immediately remove offers from client view | Medium | In Progress | Progress |
| **B004** | Minor layout issue on mobile view | Low | Open | Resolved |