**A PROPOSED OFFERING OF ONLINE DANCE CLASS BOOKING**

**MANAGEMENT SYSTEM FOR ZERO STUDIO PH**

A Requirement Specification Document 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

Presented by:

Huerto, Allen Christian Earl

Sang-an, Charllote

Tipay, Ronilo

Yana, Jovan

**INTRODUCTION**

This technical documentation aims to present the complete specifications about the booking system of Zero Studio, a dance class provider. It serves as a guide to gather all understanding regarding designers, system administrators, and even users of how the system was designed, developed, and implemented. This document shall also serve as a guide for the future maintenance and updating and optimizing to make the system perform better, be reliable, and be user-friendly.

Zero Studio's Booking System is engineered as a web-based platform that is meant to facilitate the contemporary processes of scheduling and organizing dance classes. It automates the essential features such as class reservations, scheduling management, and tracking of clients' records. Users can conveniently book their dance classes from any compatible device through online reservation, thus eliminating manual reservations. Administrators are presented with efficient tools to monitor bookings, adjust class schedules, and manage client records so that the booking workflow will be organized and seamless.

This technical documentation is all-inclusive of the technical components and processes regarding the booking system. It will include in-depth discussions on system architecture, design of database, structure of user interfaces, and their testing procedures and deployment guidelines. It will also include the technologies and methodologies as well as the tools used during development. The subject matter includes functional and non-functional aspects of the system and formulates the basis for readers to gain an overall understanding of capabilities, constraints, and operational environments.

**SYSTEM OVERVIEW**

The booking system is designed for Zero Studio, a dance class provider intending to have their online class reservations organized and facilitated. This system will allow dance enthusiasts to book their classes anywhere and everywhere using any device, avoiding the hassle of manual scheduling. The booking process will be automated to administrators managing class schedules, available slots, and customers' records at one stop. Efficiency will increase, errors will decrease, and customer experiences will be optimized while keeping as much information accurate and current as possible.

**SYSTEM ARCHITECTURE**

The system directs its communication to the Firebase database, which will handle the respective data concerning booking specific parts of the user interface for potential tuning into a client-server architecture. Client itself is made with HTML, CSS, and JavaScript to create maximum ability to adapt and serve students and administrators alike on the use of the system. On the other hand, all server functions include the validation of the bookings, viewing of classes, and data synchronization.

This architecture makes it very easy for the Zero Studio administrators to create schedules, view bookings, and keep instructor availability under real-time management. Data such as student registrations, class details, instructor profiles, and so on is preserved into the Firebase cloud database securely, thus not compromising on reliability, consistency, and availability at the same time. Automatic confirmations and notifications regarding bookings are sent through email integration, thus beautifying the convenience offered to the users.

The Dance Classes Booking System is designed to ensure a proper interaction between users and the administrators, eliminating errors and maintaining accurate up-to-date records. Flexible for future extensions to services such as online payments and attendance tracking as the system continues to grow and the technologies evolve.

**HIGH-LEVEL COMPONENTS AND INTERACTIONS**

The system consists of several key components:

* User Authentication Module: Handles the secure login and access control for the admin user.
* Class Booking Module: Enables students to browse through available dance classes, look at instructor profiles, and book a slot with the availability of real-time booking flow.
* Schedule and Instructor Management Module: Allows Admin to create, edit, and update class schedules, assign instructors, and manage their availability.
* Database and Notification Module: It uses Firebase to record data about booking, class information, and instructor content while automated booking confirmations are handled with EmailJS.
* Dashboard and Reporting Module: Provides the overview of the activities of the system including total bookings, class availability, instructor schedules and gives summaries for administrative purposes.

These components interact with each other incessantly in order to guarantee smooth operation-allowing students to easily reserve while administrators can manage schedules, instructors, and class information in a centralized organized system.

**DEPLOYMENT ARCHITECTURE**

The system is designed for deployment on web hosting services such as vercel, GitHub Pages, or Netlify, all of which, with good reason, cater to small- to medium-sized dance studios like Zero Studio PH. It provides smooth performance and easy accessibility via any modern web browser using relatively simple static web technologies such as HTML, CSS, and JavaScript. Whereas all data upload and download from class bookings to instructor info and student records is managed online with a Firebase backend database, thus ensuring security. Hence, there are no installation requirements on local machines, and the administrator can access and update any information anytime from anywhere.

The Dance Class Booking System for the Zero Studio is developed towards a stable, secure, and efficient environment for deployment. Administrators can set class schedules and look after marketing through a central web interface that requires minimal effort to operate. The system guarantees consistent performance, real-time data synchronization, and orderly information management while reaffirming reliability and scalability as the studio grows. Backup and maintenance services provided by Firebase Cloud will ensure integrity in data and availability of the system during updates or unpredicted errors.

**INSTALLATION GUIDE**

This section contains essential details as well as steps in the installation and configuration of the Dance Class Booking System for Zero Studio. It comprises system requirements, setup procedures, and configuration instructions so that the system deploys correctly and runs well on the host platform. The system can be hosted on Firebase Hosting, GitHub Pages, or Netlify, enabling admins and users to access it through any modern web browser with no installation needed.

**SYSTEM REQUIREMENTS**

To ensure smooth operation and compatibility, the system requires the following hardware and software specifications:

**Hardware Requirements (Minimum):**

* **Processor:** AMD Ryzen 3 3250U or Intel Core i3 (10th Gen)
* **Memory (RAM):** 4 GB
* **Storage:** 256 GB SSD
* **Display:** 14-inch HD (1366 × 768)

**Software Requirements:**

* **Operating System:** Windows or macOS
* **Web Browser:** Google Chrome or Microsoft Edge (latest version)
* **Hosting Platform:** Firebase Hosting, GitHub Pages, Vercel
* **Database:** Firebase Realtime Database
* **Programming Languages:** HTML, CSS, and JavaScript

**STEP-BY-STEP INSTALLATION INSTRUCTIONS**

**Step 1: Install Required Software**

Download and install **Visual Studio Code** as the primary code editor for editing and managing the system files. Create a **Firebase account** through the official Firebase website. This will provide access to the tools needed for hosting and database management. Ensure you also have a modern web browser such as **Google Chrome** or **Microsoft Edge** for testing.

**Step 2: Set Up the Database**

* Open the Firebase Console and create a new project named zero\_studio\_booking.
* In the Build section, navigate to Realtime Database and click Create Database.
* Set the database location and select Start in test mode for initial setup.
* Copy the Firebase configuration code (API key, project ID, etc.) from the project settings that will be needed later in your JavaScript files.

**Step 3: Deploy the System Files**

* Open your project folder in Visual Studio Code.
* Paste the Firebase configuration script inside your main JavaScript file
* Upload all system files including HTML, CSS, and JavaScript to your chosen hosting platform such as Firebase Hosting, GitHub Pages, or Netlify.
* Upload repository in vercel to deploy the System.

**Step 4: Configure Database Connection**

* In your JavaScript files, initialize Firebase using the configuration code you copied earlier.
* Set up database references for storing and retrieving class schedules, instructor information, and student bookings.
* Save the files and refresh the browser to verify that data syncs correctly with Firebase.

**Step 5: Test the System**

Log in as the administrator and perform test operations such as adding new class schedules, viewing instructor lists, and creating sample bookings. Check that all data is stored and displayed accurately in real time. Verify that booking confirmations and notifications function correctly through EmailJS integration.

**CONFIGURATION SETTINGS AND OPTIONS**

The system can be personalized depending on the preferences of the admin and requirements of the studio. The following adjustments can be made:

* Admin Account Settings: Set the username, password, and contact information for the administrator in the system for security and management purposes
* Class Management and Scheduling: Add, modify or delete dancing classes and change their schedule according to the current collection of sessions and availability of instructors.
* Instructor Information: Change instructor profiles that include their dance style, class schedule, and contact info for updated information.
* Email and Notification Settings: Configure the EmailJS integration to ensure that booking confirmations and notifications are successfully sent to students.

**CONFIGURATION GUIDE**

The Configuration Guide describes the step-by-step configuration and customization of the Dance Class Booking System for Zero Studio after installation. These steps ensure proper operation of the system, good synchronization of the data, and maintenance of security and stability according to the requirements of studio administrators and developers. With proper configuration, the platform conveniently manages class schedules, instructor details, and bookings.

**Initial System Setup**

Once the system successfully deployed and Firebase connection validated, the next step will be to set up basic configuration settings wherein the platform shall be customized to meet the preferences of the studio administrator and its operational needs.

**Admin Account Configuration:**

* Creating a standard account first.
* Using firebase, we can assign an account to be an admin.

**Studio Information Setup:**

* Studio information can be seen at the about us page.
* You can also see a synopsis of how the studio started and the founder itself.

**Class and Instructor Configuration:**

* In the admin dashboard, the admin has the ability to see the booking records, and the admin can also add and delete classes.

**DATABASE CONFIGURATION AND MANAGEMENT**

The system database houses all the important records like student info, class timetables, staff details, and booking history. And when it comes to getting the Firebase Realtime Database just right, you really need to get the configuration nailed to know that your data is sound, up to date and you can rely on it. Because any data that gets put in gets synced up in real time, admins can keep on top of day to day class operations while also keeping all important data secure and always consistent right across the system.

**4.1 Database Connection Settings:**

* Native SDK Preference: Use the official platform SDKs (Android, iOS, Web) over the REST API to leverage persistent connections and reduce database load/overhead.
* Connection Monitoring: Implement client-side listeners on the special path .info/connected to monitor real-time connection status and handle offline capabilities gracefully.

**4.2 Data Backup and Restore Configuration:**

* Automate Backups: Ensure the daily automated backup feature is enabled in the Firebase Console (requires the Blaze plan) to export data as a JSON file to a Google Cloud Storage bucket.
* Scheduled Retention: Configure the Cloud Storage bucket to maintain a specific number of backups and delete older files automatically to manage storage costs.

**4.3 Data Security Configuration:**

* Restrict Default Access: Start with the most restrictive rule: ".read": false, ".write": false at the database root.
* Authentication-Based Access: Use Firebase Security Rules to grant read/write access based on the authenticated user's ID (auth.uid) to ensure users can only access their own data.

**4.4 Customization and Best Practices**

* Query-Specific Indexing: Use the indexOn rule in your Security Rules (or configure indices in the console) for every child field you intend to sort or filter data by.
* Periodic Data Cleanup: Use Scheduled Cloud Functions to periodically scan the database and programmatically delete or archive old, unused, or duplicated records, replacing the need for manual "cleaning."

**User Interface Customization:**

The developers can change the system’s visual appearance by changing the CSS files in the appropriate assets or styles folder. Changing colors, fonts, backgrounds, and layout elements will help to align Zero Studio’s branding and aesthetic preferences. This will ensure that the interface maintains a uniform and professional appearance, relevant to the studio.

**System Maintenance Practices:**

* Perform database backups regularly.
* Check system performance at least once a month.
* Remove unused data or inactive tenant records to improve speed.

The Configuration Guide is all about making sure the Dance Class Booking System for Zero Studio is running like clockwork within the online space. By following the steps laid out in this guide and some general best practices, admins should be able to keep their system stable, accurate, and easy to use even on a daily basis with hardly any technical problems happening.

**API DOCUMENTATION**

**EMAIL JS**

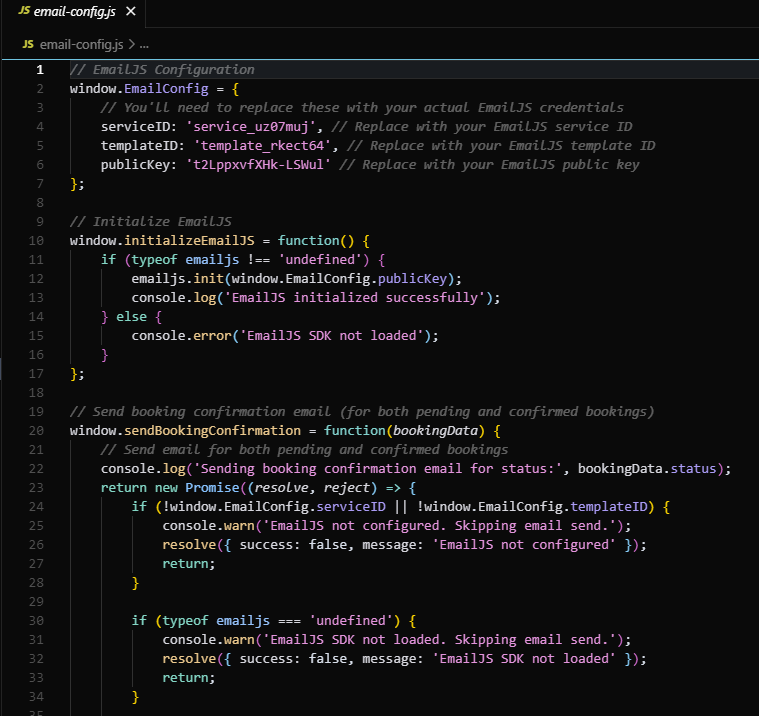
This section outlines the integration and parameters used for the EmailJS service, which handles client-side email transmission. The system utilizes EmailJS to securely send $\langle\text{briefly state what the emails are, e.g., 'user contact requests' or 'order confirmations'}.

**Integration Details**

| **Element** | **Value/Purpose** |
| --- | --- |
| Service ID | Unique identifier for the EmailJS service used. |
| Template ID | Identifier for the pre-defined email template used for this function. |
| User ID (Public Key) | The public key used to authenticate client-side requests. |
| SDK Method | emailjs.send(serviceID, templateID, templateParams) |

**Template Parameters{templateParams}**

| **Field Name** | **Type** | **Description** | **Required?** |
| --- | --- | --- | --- |
| from\_name | string | The sender's name as displayed in the email. | Yes |
| user\_email | string | The email address of the user who initiated the request. | Yes |
| message | string | The body content of the message or request. | Yes |

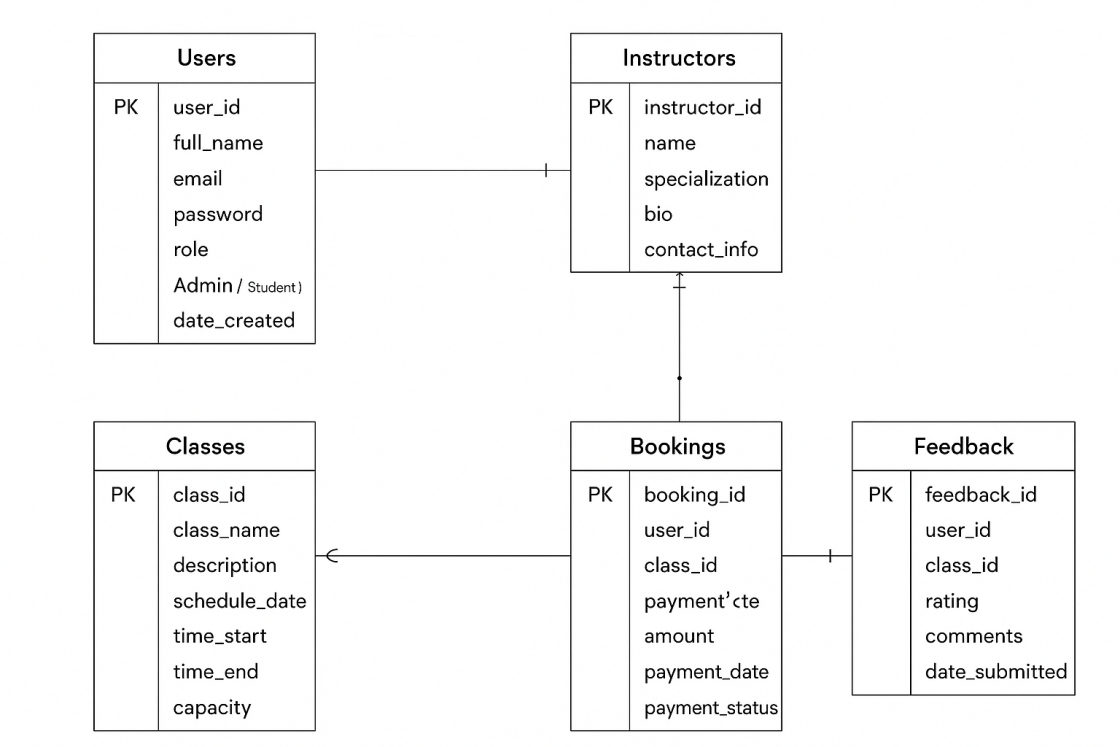
**Client-Side Usage**

**DATABASE DOCUMENTATION**

The database serves as the core of the Dance Class Booking System for Zero Studio. It manages the vital parts concerning student records, class timings, instructor information, and booking transactions. This database was built using the Firebase Realtime Database, which means the database is secure, data is structured properly, and data is synchronized in real-time so both administrators and users can access the most accurate and up-to-date information, propelling smooth and efficient booking and management operations of the system.

**Entity Relationship Diagram (ERD)**

The Entity Relationship Diagram (ERD) represents visually the logical arrangement of the database, including the primary entities, their properties, and interrelations. the principal entities are students, instructors, classes, bookings, and admins. The interconnection of these entities facilitates data structuring, fast booking transactions, real-time schedule control, and precise record keeping.



*Figure 1 Entity Relationship Diagram (ERD)*

**Description of Database Tables**

The database is composed of multiple interconnected tables, each designed to store specific data categories that support the system’s functionality.

**Users Table**

* Purpose: Stores account information for all authorized users, including administrators, instructors, and students.
* Fields: user\_id, username, password, email, role, date\_created.
* Notes: Used for login authentication, role-based access control, and user account management.

**Instructors Table**

* Purpose: Holds data about the dance instructors, including their profiles and areas of expertise.
* Fields: instructor\_id, first\_name, last\_name, specialization, contact, email, bio.
* Notes: Connected to the Classes Table to identify which instructor handles each class.

**Classes Table**

* Purpose: Contains information about the available dance classes offered in the studio.
* Fields: class\_id, class\_name, instructor\_id, schedule\_id, description, price, status.
* Notes: Linked to both Instructors and Schedules tables to specify class details, timing, and assigned instructor.

**Schedules** **Table**

* Purpose: Manages the time slots and session details for each dance class.
* Fields: schedule\_id, day, start\_time, end\_time, capacity.
* Notes:Used to organize and prevent overlapping schedules for classes and instructors.

**Bookings Table**

* Purpose: Tracks reservations made by students for specific dance classes.
* Fields: booking\_id, user\_id, class\_id, booking\_date, status
* Notes: Connects students to their booked classes and helps administrators monitor attendance and slot availability.

**Payments Table**

* Purpose: Records all payment transactions made for booked dance classes.
* Fields: payment\_id, booking\_id, amount\_paid, payment\_date, payment\_method, remarks
* Notes: Linked to the Bookings Table to ensure that all reservations have corresponding payment records.

**Feedback Table**

* Purpose: Stores reviews or comments submitted by students after attending a class.
* Fields: Feedback\_id, user\_id, class\_id, rating, comments, date\_submitted
* Notes: Useful for evaluating class quality and instructor performance.

**Data Migration and Backup Procedures**

Maintaining data integrity is essential for the continuous operation of the system. The following guidelines outline the proper methods for data migration and backup:

**Data Migration:**

* Map Data & Rules: Redesign your data from Realtime Database's (RTDB) JSON tree to Cloud Firestore's collections and documents. Update security rules to match the new structure and syntax.
* Mirror New Data: Use a Cloud Function to automatically copy new writes from the old RTDB to Cloud Firestore in real time, keeping both databases in sync during the transition.

**Data Backup:**

* Enable the Managed Export and Import Service to automatically back up our data to a secure Google Cloud Storage bucket every day. This is the main recovery tool.
* For mission-critical applications (Firestore only), enable PITR for continuous protection and recovery to any specific second within the last 7 days.

**USER MANUAL**

The Online Dance Class Booking System for ZERO Studio PH has been equipped with a user friendly guide that comes in the form of this User Manual. Navigation through the system, managing classes, bookings and payments, and performing other activities will be made easier for students, instructors and the administrator alike.

### **System Login**

**Access the System:**

* Open your web browser and go to the system https://zero-studio-web.vercel.app/mainpage.html
* The login page will appear, prompting you to enter your credentials

**Enter Login Credentials:**

* Input your Username and Password provided during installation or by the system administrator.
* Click Login to enter the system dashboard.

**Forgot Password:**

* If you cannot login, contact the studio’s system administrator to reset your password.
* Unauthorized users are not permitted access to the system admin dashboard.

**Dashboard Overview**

After successful login, you’ll be taken to the Dashboard. The dashboard acts as the control centre, showing an overview of operations including:

* The dashboard also provides quick navigation through a sidebar menu so you can easily move to other modules.

**BOOKINGS & CLASS MANAGEMENT**

**Access the Bookings Module:**

* Click on Bookings in the sidebar menu to view all reservations by students.  
  Add a New Booking:
* Click Add Booking.
* Select the student, choose the class and schedule (time slot).
* Click Save to register the booking in the database.

**Update or Cancel a Booking:**

* In the bookings list, select a booking and click **Edit** to modify details.
* To cancel, click **Delete** , then confirm to avoid accidental deletion.

**Class Module:**

* Navigate to Classes from the sidebar to view all classes offered by ZERO Studio.
* You can add a new class: click Add Class, enter class name, instructor, schedule, price, level (Beginner/Intermediate/Advanced) and click Save
* The class list shows styles like K-Pop, Hip-Hop, Dance hall, instructor, schedule and price.

**Reports & Analytics**

* Reports can be seen in the admin dashboard, admin can see the booking details.
* Plus, you can also add or delete classes.

**Logging Out**

* Basically after logging out on your account, you do not have the ability book but you can still check the schedules and other modules.

**TROUBLESHOOTING GUIDE**

The Troubleshooting Guide is for administrators on how to solve common problems that possibly arise in installing or using the Dance Class Booking Management System for ZERO Studio PH. It provides quick-fixes that would help smoother the operation of the system, prevent data corruption, and maintain an optimal performance.

**Common Issues and Error Messages**

Below are the most frequent problems encountered while using the system, along with their corresponding causes and recommended solutions:

| **Issue** | **Possible Cause** | **Resolution** |
| --- | --- | --- |
| Login failed | Wrong username or password entered. | Verify your credentials; if forgotten, reset password via the admin console. |
| Student/Booking not appearing in list | Data entry incomplete or class scheduling sync issue. | Verify all mandatory fields are filled; check schedule assignment. |
| Email not sending | Invalid User Email or not yet logged in. | Must fill the email section so in that way the system can send the email in the email you input. |
| Report not generating | No data available, wrong filters, or system error. | Verify database entries; ensure filters are correct; reload the report module. |

*Table 1 Common Issues and Error Messages (CIEM)*

**Troubleshooting Procedures**

**Checking Server Status:**

* Firebase Status Dashboard: https://status.firebase.google.com/
* This page lists the real-time status (Available, Service disruption, Service outage) for all products, including Authentication, Realtime Database, Firestore, Cloud Functions, and Hosting.

**Verifying Database Connection:**

* Go to Firebase and confirm that the database dancebooking db exists and contains all required tables.

**Fixing Login Problems:**

* If the admin cannot log in, open the users table in Firebase to check for authentication.
* Check that the username and password fields are correct.

**Resolving Page Loading Errors:**

* Clear the browser cache and reload the system URL (http://localhost/dance\_booking\_system).
* Ensure that no duplicate or missing files exist in the project folder.

**Repairing Database Corruption:**

* In Firebase , Restore from Scheduled Backup; Restore the entire database from a daily or weekly managed backup to a new database instance.
* Always back up the database before performing repair operations.

**Preventive Maintenance Tips**

* Rules are your only server-side validation. Never trust the client. Use the request.auth.uid variable to ensure users can only read/write their own data.
* Use the validate or request.resource.data object in your Rules to ensure all incoming data is the correct type, format, and size before it gets written.
* Protect your backend services (Firestore, RTDB, Storage) from abuse, unauthorized access, and billing fraud by verifying that requests come from your legitimate app.
* By default, deny all reads and writes (allow read, write: if false;). Only open up specific paths and methods (like read, write, create, update, delete) with strict conditions.

**Contact Information for Support:**

* Development Team:
* Email:
* Contact Number:

**CODE DOCUMENTATION**

**TESTING DOCUMENTATION**

The Testing Documentation defines the tests applied to validate the Dance Class Booking Management System for ZERO Studio PH in terms of smooth, accurate, and secure operation. It contains test plans, test cases, and results which prove that every single module functions properly, and the whole system works reliably.

**Test Plan**

The testing plan defines the objectives, scope, and types of testing conducted throughout the system development.

**Objectives of Testing:**

* Make sure all system components are functioning like originally designed.
* Verify that all features and functionalities of the system operate as intended.
* When entering data, make sure you check storage, retrieval, and display accuracy.
* Validate the performance of the system with different user operations.
* Check for security with reference to user authentication and data handling activities.
* Assess the responsiveness and performance of the system during simultaneous operations.
* Identify and resolve potential defects before deployment to ensure long-term stability.

**Scope of Testing:**

The scope includes all major modules of the system User Authentication, Educator Administration, Class Timetable, Booking Management, Payments, and Reports. Each of the modules was subjected to functional testing, integration testing, and user acceptance testing (UAT) phases.

**Testing Environment:**

* Hardware: AMD Ryzen 5 5600g
* Operating System: Windows
* Web Server: Vercel & Firebase
* Browser Used: Microsoft Edge, Google Chrome
* Testing Tools: VSCode

**Test Cases**

The table below presents the primary test cases performed, along with expected and actual outcomes:

| **Test Case ID** | **Module Tested** | **Test Description** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-01 | Login Module | Enter valid username and password | Redirect to Dashboard | Redirected successfully | Passed |
| TC-02 | Login Module | Enter invalid credentials | Display “Invalid login credentials” message | Error message displayed | Passed |
| TC-03 | Instructor Module | Add a new instructor record | Instructor saved and displayed in list | Record added successfully | Passed |
| TC-04 | Class Module | Add new dance class schedule | Class added and visible in class list | Record added successfully | Passed |
| TC-05 | Booking Module | Create a new booking | Booking saved and displayed under user account | Record added successfully | Passed |
| TC-06 | Booking Module | Cancel a booking | Booking status updated to “Cancelled” | Status updated successfully | Passed |
| TC-07 | Payments Module | Record class payment | Payment stored and linked to booking | Transaction recorded | Passed |
| TC-08 | Reports Module | Generate booking summary report | Report generated with accurate totals | Correct report displayed | Passed |
| TC-09 | Database | Verify data integrity | Data remains consistent across modules | No inconsistencies found | Passed |
| TC-10 | System Performance | Simulate multiple booking transactions | System responds without delay | Stable performance maintained | Passed |
| TC-11 | Reports Module | Generate daily booking Reports | Report displays all relevant data with correct totals | Report displayed accurately | Passed |
| TC-12 | Reports Module | Generate instructor performance report | Report shows accurate class counts | Data displayed correctly | Passed |
| TC-13 | Database | Verify data consistency across modules | All related data synchronized correctly | No inconsistencies found | Passed |
| TC-14 | System Performance | Simulate multiple user transactions | System responds smoothly with no lag | Stable performance maintained | Passed |
| TC-15 | UI/Navigation | Access all major pages from navbar | All pages load correctly without errors | Navigation works properly | Passed |

*Table 2 Test Cases*

**Test Results and Analysis**

The testing confirmed that all modules of the Dance Class Booking Management System at ZERO Studio PH were tested and confirmed to be working perfectly. The login feature uniquely authenticated users, and the modules for instructors, classes, and bookings were corresponding to the information they presented. Payments were recorded correctly and tied to their relevant bookings, with entirely accurate reports being generated. Performance tests revealed that the system was stable with many running processes. Minor interface issues were corrected before the final release of the project. After the User Acceptance Test (UAT), both administrators and users approved the system as compliant with operational requirements, managing the accommodation of classes more easily, while simplifying the bookings process for the studio.

**Defect Reports and Resolution**

During initial testing, a few minor defects were identified and resolved:

| **Defect ID** | **Description** | **Module Affected** | **Resolution** |
| --- | --- | --- | --- |
| D-01 | Payment totals not updating correctly after multiple transactions | Payments Module | Revised the calculation logic in JavaScript and ensured accurate database synchronization for totals. |
| D-02 | Missing validation for blank fields in booking and instructor forms | Booking & Instructor Modules | Added input validation for all required fields and implemented front-end alerts for incomplete forms. |
| D-03 | Inconsistent date format in reports and receipts | Reports Module | Standardized all date formats to “MM/DD/YYYY” across tables and report exports. |
| D-04 | Navigation issue after login, redirecting to incorrect page | Dashboard | Fixed the redirect path in script and verified proper routing based on user role (Admin, Instructor, Student). |
| D-05 | Booking duplication when refresh occurs after submission | Booking Module | Implemented duplicate submission prevention and added booking ID verification before insertion. |
| D-06 | Instructor image upload not displaying | Instructor Module | Adjusted file path references and updated file permissions for upload directory. |

*Table 3 Detach Reports*

All reported issues were fixed and verified through retesting. After corrections, no additional defects were found, confirming system stability.

**MAINTENANCE GUIDE**

The Maintenance Guide contains the procedures for the maintenance and updating of the Dance Class Booking Management System for ZERO Studio PH to ensure uninterrupted stability and security and optimal performance. It lays down detailed procedures for the resolution of bug fixes, version control, and enhancement of systems in an ongoing manner.

**Regular System Backups:**

* Perform weekly backups of the database using phpMyAdmin or any MySQL management tool.
* Save backup copies on secure external drives or cloud storage.
* Test backup files periodically to ensure they can be restored successfully

**. Data Management:**

* Review and delete outdated or duplicate records.
* Keep payment and invoice records updated for accurate reporting.
* Regularly optimize database tables in phpMyAdmin to maintain fast query performance.

**System Performance Monitoring:**

* Check that the system responds quickly when navigating between modules.
* Restart XAMPP services (Apache and MySQL) if performance issues occur.
* Ensure sufficient computer storage and memory resources are available.

**Security Updates:**

* Change the admin password regularly.
* Disable or remove unused user accounts to prevent unauthorized access.

**Version Control and Release Management**

Versioning system helps track changes, updates, and enhancements made to the Dance Class Booking Management System for ZERO Studio PH. It ensures that any code changes made by developers for feature enhancements or bug fixes have a proper history and can be rolled back if needed. Version control also helps to keep the development teams in sync and allows for stability of the system after every update.

**Version Tracking:**

* Maintain a changelog file documenting every update, bug fix, and added feature.

**Code Backup and Archiving:**

* Keep previous stable versions as backups before applying major updates.

**Release Management:**

* Test all new features on a development server before deploying them to the live environment.
* Prepare release notes summarizing what has been added, fixed, or changed
* Inform administrators of any changes in system functionality.

**Handling Bug Fixes and Enhancements**

As part of maintenance, the system errors or performance problems should be identified and solved promptly as part of regular services aimed at keeping systems running smoothly. Regular monitoring and structured debugging ensure a smooth operation and long-term reliability and security of the system.

**Bug Identification:**

* Monitor system logs and user feedback to detect unusual behavior or system errors.
* Record details of the bug, including date, affected module, and error message.

**Bug Resolution Process:**

* Reproduce the issue in a test environment.
* Debug the affected code using tools such as Visual Studio Code.
* Apply fixes and re-test to confirm successful resolution.

**System Enhancement:**

* Periodically review system features for improvement opportunities.
* Ensure backward compatibility when integrating new modules.

**REVISION HISTORY**

The following records all updates and changes to the Dance Class Booking Management System for ZERO Studio PH documentation and software. It provides a complete record of the version number, date, and reason for every modification along the project development life cycle. And Keeping a detailed set of changes ensures transparency, accuracy in documentation, and tracking of the progress from the very start of its proposal to final deployment and maintenance.

| **Version** | **Date** | **Description of Changes** |
| --- | --- | --- |
| 1.0 | September 20, 2025 | Initial draft creation including introduction and project background. |
| 1.1 | September 25, 2025 | Added system overview and overall architecture description. |
| 1.2 | September 28, 2025 | Included installation guide and configuration setup instructions. |
| 1.3 | October 1, 2025 | Added database documentation and entity relationship diagram (ERD). |
| 1.4 | October 4, 2025 | Expanded data handling design and UI layout sections |
| 1.5 | October 6, 2025 | Created test plan and initial test case documentation. |
| 1.6 | October 8, 2025 | Added troubleshooting guide and data migration procedures. |
| 1.7 | October 10, 2025 | Revised user interface design and navigation flow. |
| 1.8 | October 12, 2025 | Added glossary and appendix sections to documentation. |
| 1.9 | October 14, 2025 | Conducted functional and integration testing for main modules. |
| 2.0 | October 15, 2025 | Optimized performance and improved database queries. |
| 2.1 | October 16, 2025 | Added defect report and bug resolution logs. |
| 2.2 | October 17, 2025 | Included maintenance guide and version control procedures. |
| 2.3 | October 18, 2025 | Updated user acceptance testing results and summary |
| 2.4 | October 19, 2025 | Finalized technical documentation for project submission. |
| 2.5 | October 20, 2025 | Implemented security enhancements and file optimization. |
| 2.6 | October 22, 2025 | Completed final quality assurance and system verification. |
| 2.7 | October 25, 2025 | Added final appendices and project credits. |
| 2.8 | October 27, 2025 | Released final stable version for deployment and presentation. |

**APPROVAL**

This certifies that the approval section endorses the Dance Class Booking Management System of ZERO Studio PH, with its technical documentation. It shows completion of thorough review and approval by the stakeholders of the project. By such a certification, it affirms their compliance with technical, functional, and performance requirements established during system development.

| **Name** | **Designation** | **Signature** | **Date** |
| --- | --- | --- | --- |
| Mr. | Project Adviser |  |  |
| Ms. | Panelist |  |  |
| Mr. Jovan Yana | Software Engineer |  |  |
| Mr. Ace Allen Huerto | Project Leader |  |  |
| Mr. Ronilo Tipay | Data Gatherer |  |  |
| Ms. Charllote Sang-an | Data Gatherer |  |  |