

Web-Based Management System: Luna Yoga

Ng Jing Jing¹, Mubarak Saif Mohsen Al-Absi^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: mubaraksm@uthm.edu.my
DOI: <https://doi.org/10.30880/aitcs.2025.06.01.007>

Article Info

Received: 13 June 2024

Accepted: 8 May 2025

Available online: 30 June 2025

Keywords

Web-based system, Yoga, Sound bath,
Yoga class management, Course
management, Class scheduling
management, Booking system,
Attendance tracking, Reward
redemption

Abstract

Luna Yoga Management System is a web-based management system for lessening administrative workload with the class management function and financial and reporting function; it empowers instructors to facilitate the yoga class while also improving learners' booking, payment, and communication processes. The system is proposed to solve the overlap booking problem, which is error-prone when doing accounting analysis, and enhance booking processing time. The methodology of this system uses the Waterfall model. The implementation part of the system uses PHP and HTML programming language with MySQL as the database. The development of this management system makes yoga class management simpler for the administrator and instructor, allowing them to schedule, book, update, and facilitate yoga and sound bath classes. At the same time, learners can easily book their yoga classes through this online system.

1. Introduction

Luna Yoga is a yoga studio that started in 2022 and was founded by Tan Sze Ying, a yoga instructor. Yoga is a philosophical practice that brings mind, body and spirit together. Luna Yoga has a yoga class named Hatha Yoga, an antique practice that focuses on physical postures, breathing techniques, and relaxation. It represents the balance of strength and flexibility, energy and relaxation. Other than that, Luna Yoga also provides a Singing Bowl Sound Therapy class. A Tibetan singing bowl is a type of bowl that vibrates and produces a rich, deep tone when played. The singing bowls are played to create a symphony of healing vibrations that release stress and provide relaxation.

Currently, Luna Yoga depends only on a manual booking system via WhatsApp, which has restrictions that may affect the overall efficiency of the booking process. Without a customized booking system, there are missed messages, schedule conflicts, reservation confirmation delays, overcrowded classes and increased administrative workload.

The objectives are to design a web-based management system for Luna Yoga to manage their yoga classes. Secondly, to develop a web-based yoga class management system for yoga learners to book for a class effectively. To test the Web-Based Management System: Luna Yoga through user acceptance test.

The significance of developing the Luna Yoga online management system is that it provides easier accessibility no matter where a user is. Other than that, the Luna Yoga management system increases the efficiency of administrative management by facilitating student engagement and class administration.

The paper is organized as follows: Section 1 introduces the background of Luna Yoga. Section 2 reviews the related works. Section 3 outlines the methodology. Section 4 highlights the testing results. Section 5 summarises the project.

2. Related Work

A management system consists of individuals, computers, processes, and data models as components of the system [1]. Currently, Luna Yoga's management system is paper-based to record all the data while the booking process goes through the WhatsApp platform. With the development of the Luna Yoga management system, Luna Yoga can facilitate class schedules, learner registration and profiles, financial reporting, announcement updates, and feedback forms through a web-based system. The scheme collects information from an organization's internal and external sources [1]. For example, class management includes scheduling, managing student information, and tracking the booking details. The system approach correlates with both the dynamics of problem-solving and the value of stakeholders over time, as well as the levels of the system connection and detailed management [2]. The system is essential in solving issues like overbooking classes, which can result in crowded classrooms and lower-than-expected instruction quality. Booking is defined as the customer's pre-order that needs to be queued and sorted [3]. An electronic booking system can prevent the order from missing and manage the order accurately. Learners can easily browse for available classes, view class requirements, and schedule sessions using the web-based system. Computerized systems are fast, reduce data anomalies, reduce data redundancy, low paper usage, and generate information needed automatically [4]. Admin will be able to download financial reports through the developed system.

Initially, the administrator will manage next month's class schedule in Fig. 1. The administrator will update the posts on their Facebook and Instagram accounts. To place reservations, learners can join yoga classes by contacting the administrator through WhatsApp. The administrator will check for the learner quantity of a class to ensure it is not overcrowded. Learners will pay by cash when the class ends for attending classes at the counter. The administrator will need to do finance work daily and record the data manually.

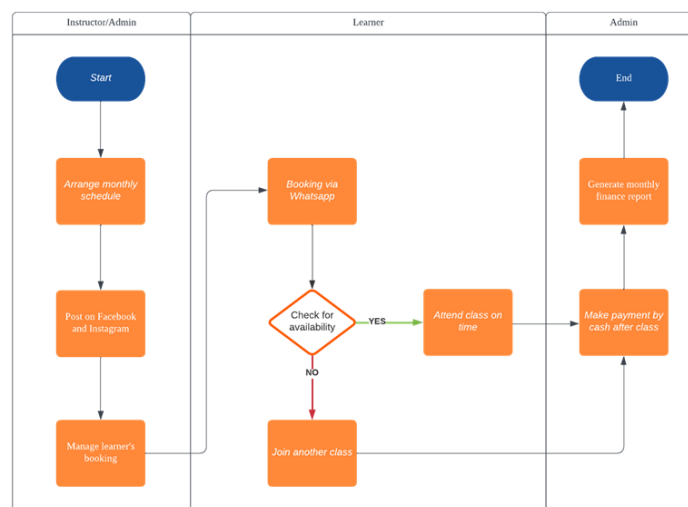


Fig. 1 Operation Process of Luna Yoga

2.1 Study of Related Existing System

In this section, three existing systems are being discussed and compared which are Life Hot Yoga, Karma Yoga and Jal Yoga.

The Life Hot Yoga booking website has several significant features that make it an informative tool for studio owners and students[5]. It is simple for users to register and provide personal data, which facilitates personalized class selection. The system offers class types and benefits, and real-time class scheduling, allowing users to browse available classes and book classes that fit their schedules. Additionally, the system includes communication features like live chat with messages that have been set with questions and answers. Moreover, an upcoming event notice will be shown on the website.

The Karma yoga booking website system has several features that make it a useful medium for both studio owners and learners[6]. When browsing the website, a pop-up message will be shown for users to register and provide personal data before signing up for a yoga course. This yoga studio allows users to join courses to become yoga instructors or to join courses as learners. The system shows some articles about their yoga community- community updates and information about the yoga studio. There are several types of courses available for new instructors and existing teachers; details such as the schedule of training, trainer name, and application process are displayed on that page. Additionally, the system includes communication features using Messenger that require users to log in for chat sessions.

The Jal Yoga booking website system provides several features that make it a valuable platform for studio owners and learners[7]. However, it does not allow users to log in, register, send feedback and make payments.

Table 1 illustrates the comparison of the proposed system, Luna Yoga, with the existing system, which are Life Hot Yoga, Karma Yoga MY and Jal Yoga.

Table 1 Comparison with Existing System

Functions/System	Life Hot Yoga	Karma Yoga MY	Jal Yoga	Luna Yoga
Login Module	Yes	Yes	No	Yes
Registration Module	Yes	Yes	No	Yes
Administrator View	Yes	Yes	Yes	Yes
Instructor View	Yes	Yes	No	Yes
User View	Yes	Yes	Yes	Yes
Analytic and Profit Module	Yes	Yes	Yes	Yes
Class Management Module	Yes	Yes	Yes	Yes
Announcement Module (news feed, event)	Yes	Yes	Yes	Yes
Communication Module	Yes	Yes	Yes	Yes
Booking Module	Yes	Yes	Yes	Yes
Manage Waiting List	No	No	Yes	Yes
Review and Feedback Module	Yes	Yes	No	Yes
Payment Module	Yes	Yes	No	Yes
Reward Module	No	No	No	Yes

3. Methodology

The Waterfall model is chosen as the methodology for developing the Luna Yoga Management System, which is one method in the System Development Life Cycle (SDLC) that has the characteristics of working on each phase in the waterfall that must be completed before proceeding to the next phase [8]. This means that the focus on each phase can be maximized because there is no parallel work. In Luna Yoga Management System, the SDLC starts from the planning phase, which defines the scopes, objectives and expected outcomes that meet the needs of Luna Yoga. Next, the analysis phase of Luna Yoga focuses on the system's requirements analysis. System developers require discussion that aims to identify project constraints and user expectations [9]. The design phase presents the user interface of the proposed system, including the dashboard, class schedules, user profile, communication page, and others. In the implementation phase, all the coding will be applied to deploy the system interface and the backend of the system will be deployed. Finally, the systems support and security provide vital protection and maintenance services for system hardware and software, networks, and transaction processing systems. Fig 2 illustrates the SDLC for the Waterfall model, while Table 2 shows the system development workflow.

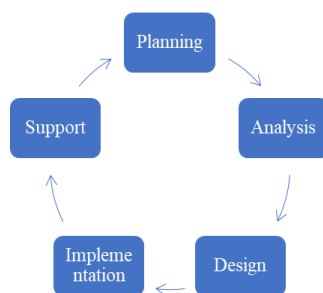


Fig. 2 Waterfall Model [8]

Table 2 System Development Activities and Their Tasks

Phase	Task	Output
Planning	<ul style="list-style-type: none"> - Conduct preliminary analysis on the current system - Propose system title, objectives, problem statement and project significance - Determine project schedule, deliverables 	<ul style="list-style-type: none"> - Preliminary version of the project proposal - A Gantt chart - Conduct an interview session
Analysis	<ul style="list-style-type: none"> - Analysis of the current system with some system-related terms - Gather system requirements - Analysis of system structure such as data store, data flow and entity relationship. - Select methodology model 	<ul style="list-style-type: none"> - Literature review of related work - Functional and non-functional requirements - Describe the system's functionalities modules - Explain the Waterfall model
Design	<ul style="list-style-type: none"> - Plot system diagrams - Design database - Design system prototype 	<ul style="list-style-type: none"> - Create flowchart, context diagram, data flow diagram, ERD. - Construct a data dictionary to define the database - Develop system interface design
Implementation	<ul style="list-style-type: none"> - Construct code for the proposed system using tools and technologies - Applied unit testing for each interface and integration testing between interfaces - Debug the code - Deploy the system 	<ul style="list-style-type: none"> - Write the code using HTML and PHP programming language - Connect SQL database - Test the source code - Fix the errors and bugs in the code and re-test - Publish the system
System security and support	<ul style="list-style-type: none"> - Secure sensitive user information - Maintain the system's authenticity - Implement multi-factor authentication - Implement security checks and vulnerability assessments - Set up user support 	<ul style="list-style-type: none"> - Set password following the rule (at least eight characters, at least one digit(s), at least one letter(s), and at least one non-alphanumeric character(s)) - Apply security reset token other than password for forget password process - Set session timeout

3.1 Planning Phase

A preliminary analysis of the project to determine the aim of the project and the issues of the business's existing system. A project timeline, and Gantt chart as shown in Appendix A, Fig. A1 and Fig. A2 were presented to schedule the duration of each phase and set the milestones for progress.

3.2 Analysis Phase

Information can be gathered through interviews, discussions, and in-person surveys[9]. An interview session is conducted to gather and document the specific needs and expectations of stakeholders for a project or system. Requirement analysis is the systematic process of identifying, gathering, and documenting the specific needs and expectations of stakeholders for a particular system or project. Table 3 illustrates the functional requirements. The non-functional requirements analyzed include the performance of the system which allows quick response times, availability for 24 hours, scalability of the system in managing demands for class scheduling, security to allow only authorized users to log in to the system and use the function while guest cannot access privileged functions and compatibility so that the system can be used on different web browsers, operating systems, and mobile devices.

Table 3 Functional Requirement

System Module	Function	User
Login module	Allow valid users to access the system.	Administrator,
	Enable forget the password to reset a new password.	Instructor, Learner
	Allow for registration to become a user of the system.	Learner
Class management module	Create, edit and delete a class type by assigning an instructor.	Administrator
	Create, edit and delete a class schedule.	
	View the class schedule.	Instructor, Learner
Analytic and profit module	Analyze the number of learners monthly and monthly income from yoga and sound bath.	Administrator
	Update learners' payment status.	
Announcements module	Create, edit and delete a post or important announcement.	Administrator
Booking module	Allow to book a class and join the waiting list.	Learner
	Manage the waiting list.	Administrator
	Update the attendance status for each class.	Instructor
Review and feedback module	Give feedback.	Learner
	Check and update feedback status.	Administrator, Instructor
Profile management module	Create, edit and delete learners' and instructors' profiles.	Administrator
	Edit profile.	Administrator, Instructor, Learner
Payment module	Allow for package purchase.	Learner
Communication module	WhatsApp communication platform.	Administrator, Learner
Reward module	Add a new redeemable item.	Administrator
	Redeem the reward using points collected.	Learner
Communication module	WhatsApp communication platform.	Administrator, Learner
Reward module	Add a new redeemable item.	Administrator

3.2.1 Hardware and Software Requirements

The hardware requirement is related to the system's hardware like computer specifications while the software requirement indicates the software chosen to develop the proposed system. Table 4 depicts the hardware specification to develop the proposed system while Table 5 shows the software specification used to develop the proposed system.

Table 4 Specification of Hardware Requirement

Hardware	Specification
Central Processing Unit	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
Memory/RAM	8.00 GB
Graphic Processing Unit	Intel(R) Iris(R) Xe Graphics NVIDIA GeForce MX450
Screen Resolution	1920 x 1080 full HD

Table 5 Specification of Software Requirement

Software	Description
Microsoft Windows 11	Operating system to support other software
Project Libre	Build a Gantt chart for the project timeline
Lucidchart	Draw the diagrams and charts (flowchart, context diagram, DFD, ERD)
Visual Studio Code	Source code editor for HTML, PHP, JavaScript and CSS programming
MySQL	Design and manage the database of the system
XAMPP Control Panel	Web server used to test program on localhost

3.3 Design Phase

The flowcharts, data flow diagrams and entity relationship diagrams are depicted to fulfil the data requirements using Lucidchart. The flowchart shows the system's process, which involves different users. Next, the context diagram presents the relationship between the system and external entities with the data flow between them. The data flow diagram describes the interaction between the system's processes and external entities with the stored database. The entity relationship diagram shows the relationship between entities and attributes, while the data types and descriptions are shown in a data dictionary. The interface design of the system is illustrated to present the intended look and feel by developing the website pages

3.3.1 Flowchart

There are three roles in the Luna Yoga Management System which are administrator, instructor and learner. The flowchart for learners is shown in Fig. 3. The flowchart for the administrator is shown in Fig. 4. The flowchart for the instructor is shown in Fig.5.

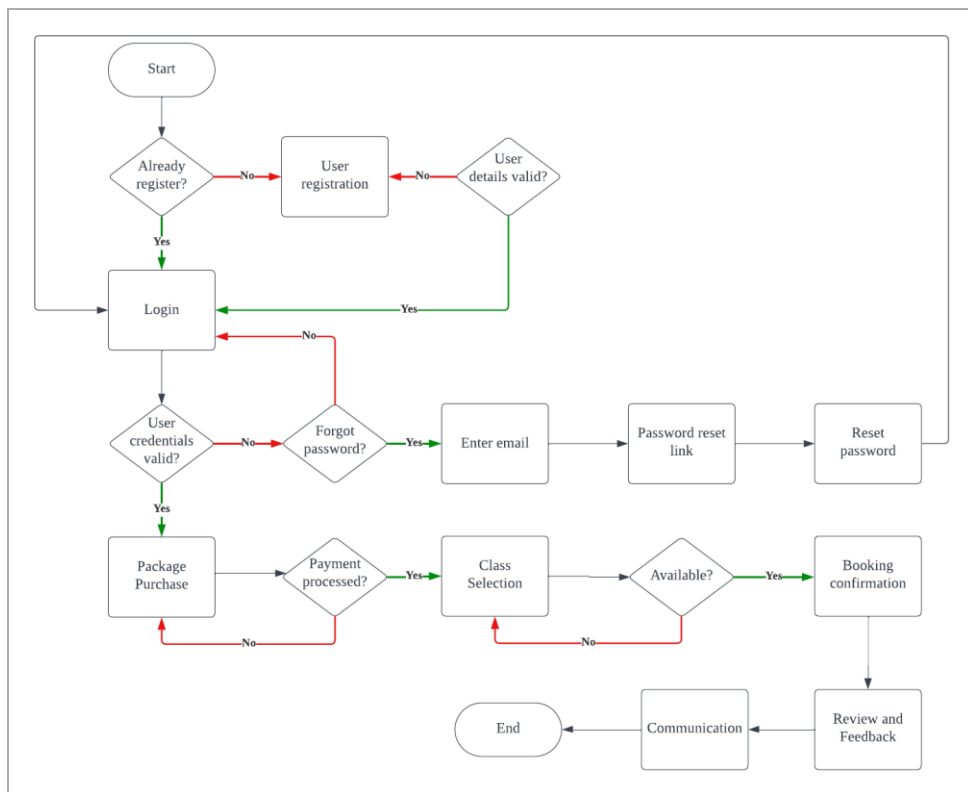


Fig.3 Flowchart for User

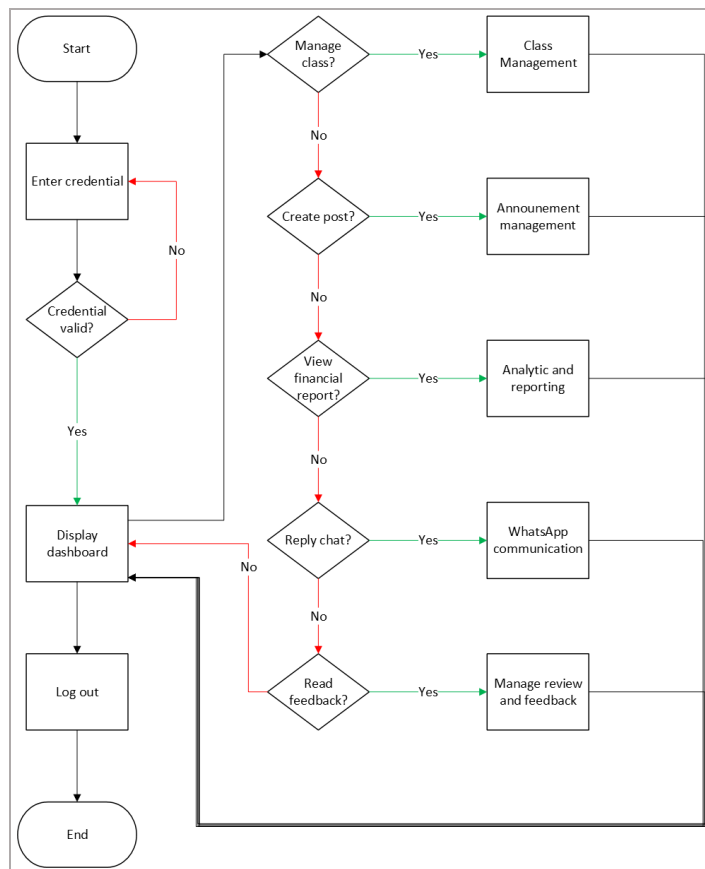


Fig. 4 Flowchart for Administrator

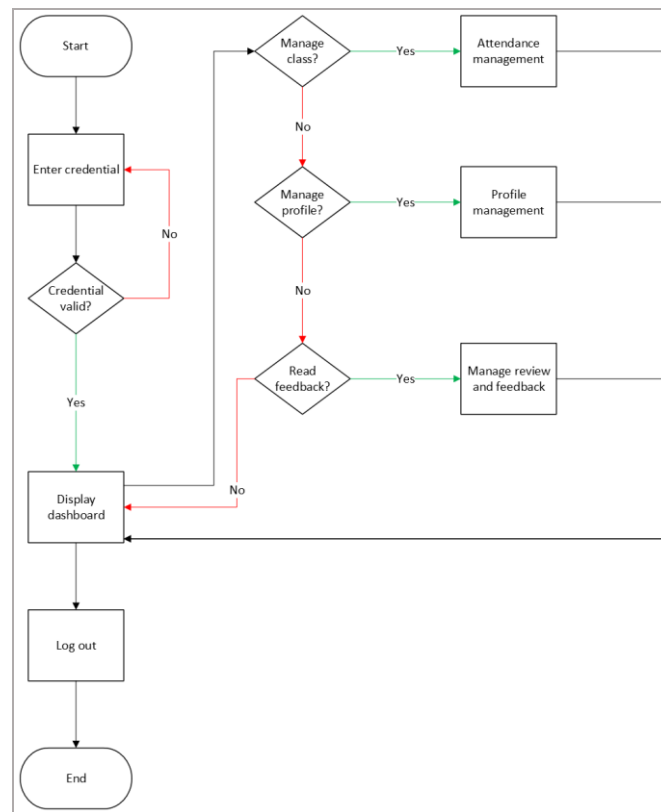


Fig. 5 Flowchart for Instructor

3.3.2 Context Diagram

The user engages in user registration and booking details, provides feedback, initiates live chat, and makes payment. Next, the instructor is involved in user registration, updating class schedules, receiving booking details, replying to the chat, and viewing the feedback. Furthermore, the administrator manages user registrations, verifies booking confirmations, receives financial reports, and updates schedules. In booking details, the flow shows where users provide the details to the system, and the system will send it to instructors. The system will receive the user's payment details and send them to the administrator. The financial report generated will also be sent to the administrator. Instructors can update class schedules for process updating, and this information may flow to users. The context diagram is shown in Appendix B, Fig. B1.

3.3.3 Data Flow Diagram(DFD)

In the context of the Luna Yoga Management System, a level 0 DFD would outline the primary processes and entities without delving into detailed sub-processes. In process 1.0 Login & register, can be used by all entities. This process will check authentication and authorise an entity to access the system. Process 2.0 is booking class represents the bookings made by users for specific yoga classes. Instructors can retrieve more information about the learners that join the class meanwhile the administrator can check the class status to make sure there is no overbooking. Process 3.0 indicates creating a class which involves instructors setting up a class and the administrator will be notified of the class requirements. In process 4.0 update the schedule and announce it, the administrator will create important events that will shown in the user's and instructor's dashboard. Process 5.0 is a redeem reward where the administrator creates a reward while users use points to redeem a reward. Process 6.0 process payment allows users to make online payments after selecting a package while the payment details will be stored and financial reports to the administrator. Process 7.0 is reviewing feedback where users can submit feedback so the administrator and instructors can view the feedback and make improvements. The DFD level 0 is shown in Appendix B, Fig B2(a) and (b).

3.3.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram is a visual representation of the data model that defines the structure and relationships within a database. There are several entities included in the system which are the users, classes, schedule, bookings, packages, user_package, payment, attendance, user_points, redemptionHistory and redeemableitems. The ERD is shown in Fig. 6.

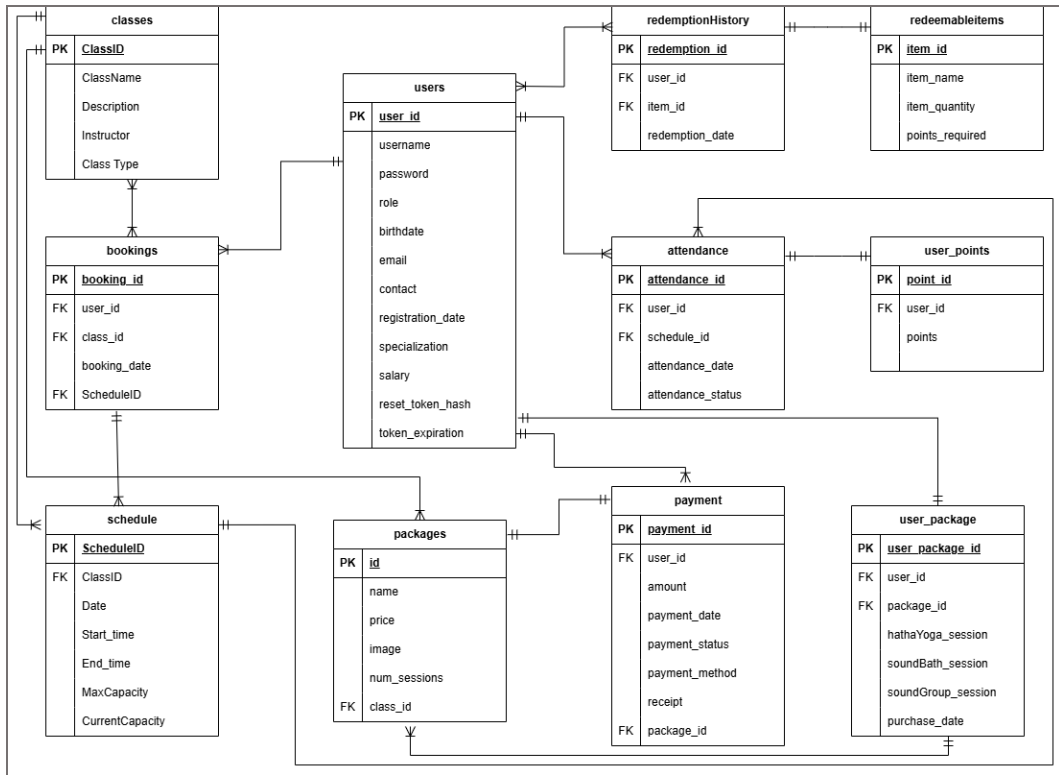
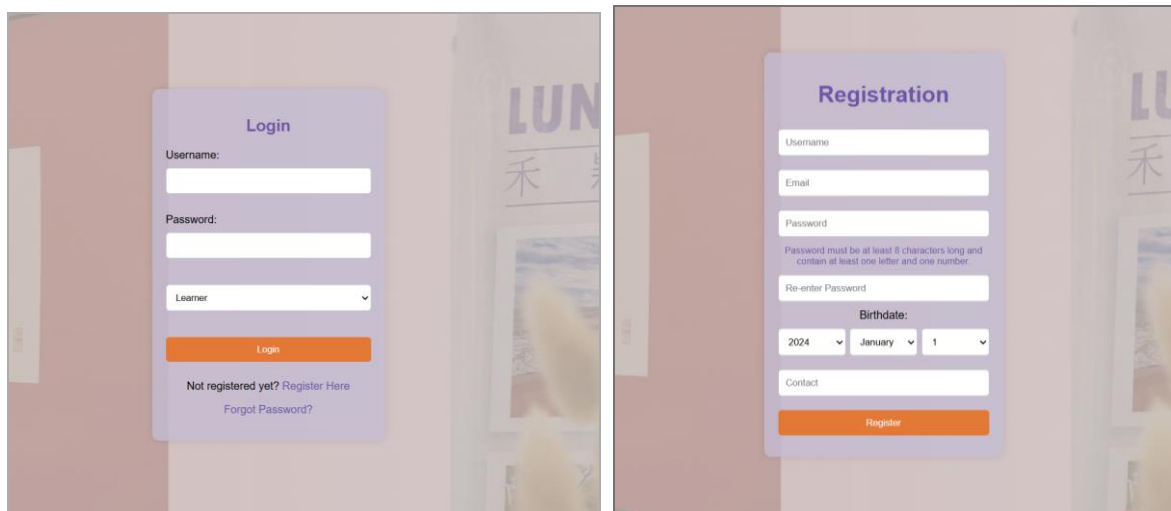


Fig.6 Entity Relationship Diagram

3.4 Interface Design of System

Interface design means creating a visual structure to illustrate the layout design of the Luna Yoga management system. It is a model for visually representing the layout and design of a system’s important components before the final product is created. The Luna Yoga Management System user interface design is displayed below. Fig. 7(a) illustrates the registration page while Fig. 7(b) shows the login page. Figure 8(a) presents the dashboard page for learners. Figure 8(b) indicates the pricing page. Figure 9(a) depicts the schedule page. Figure 9(b) below shows the profile page. Figure 10(a) below depicts the user interface for the learner feedback page. Figure 10(b) below shows the reward page for learners.



(a)

(b)

Fig. 7 User Interface (a) Login Page; (b) Register Page

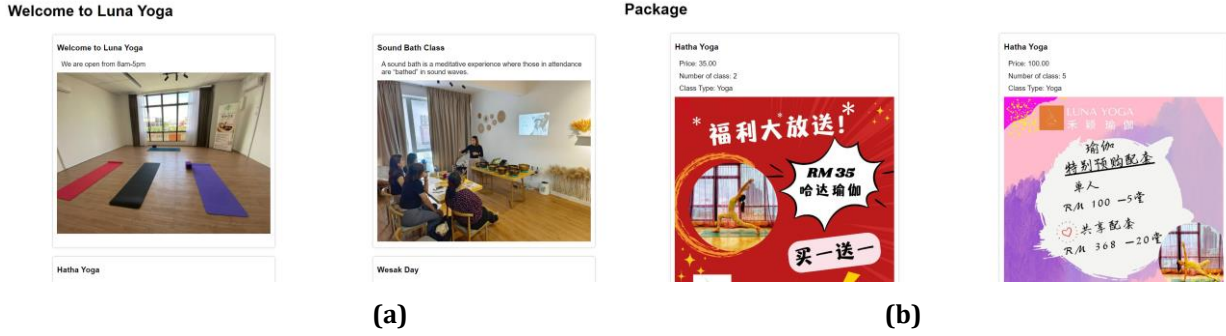


Fig. 8 User Interface (a) Dashboard Page; (b) Pricing Page

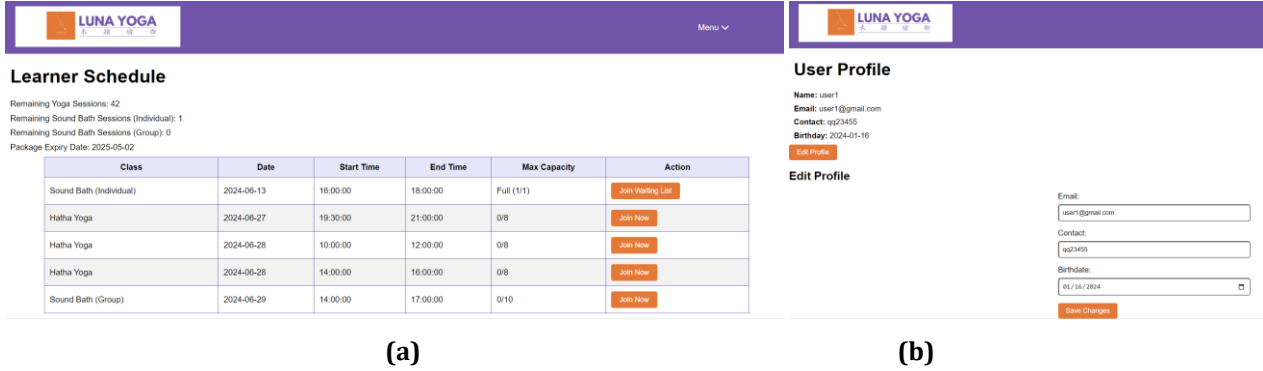


Fig. 9 User Interface (a) Schedule Page; (b) Profile Page

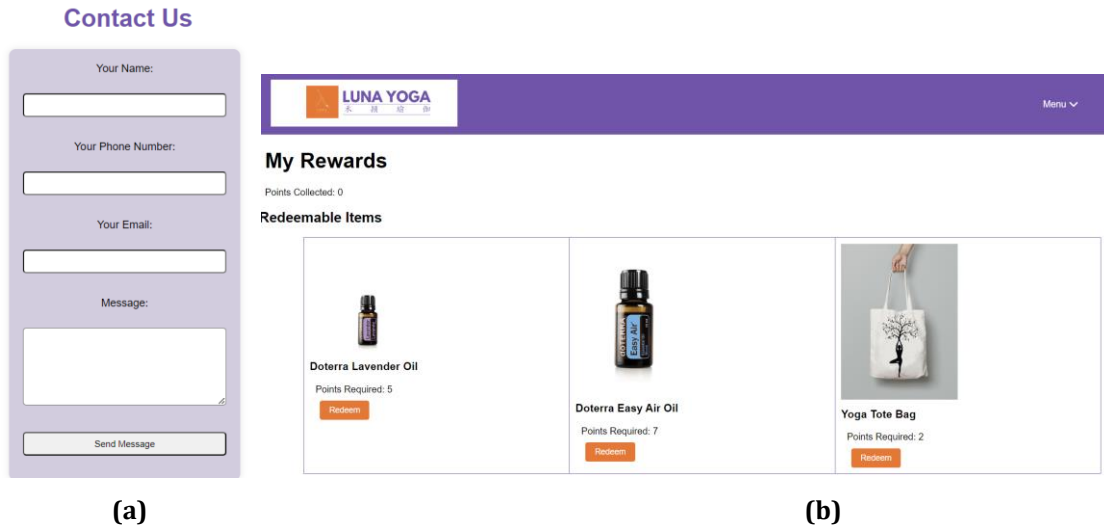


Fig. 10 User Interface (a)Feedback Page; (b) Reward Page

4. Implementation and Testing

The system modules and functionalities are developed using the software, hardware, and tools that are determined. This section describes the system implementation process and the system testing for the developed system.

4.1 Implementation of System

The system's coding is presented in Appendix C. The login module allows users, including learners, administrators, and instructors, to access the system by entering their username, password, and role. Fig. C1(a) represents the PHP script that handles user authentication for login functionality. Fig. C1(b) indicates the code segment for user registration validation. The script validates the submitted data, ensuring that the username, email, and password meet the required criteria. Appropriate error messages alert if any validation errors occur.

The code segment for the booking module which fetches user package details from a database table based on a user ID is presented in Fig. C2(a). If conditions are met the script proceeds on Fig. C2(b), it checks if the user

already has a booking for the same time slot; if not, it offers a button to join the class. If the maximum capacity has been reached, it provides an option to join the waiting list. For adding to the waiting list, the code shown in Fig.C3(a) is structured to check if learners have remaining sessions and ensure they are not already on the list while displaying appropriate alerts. The code in Fig. C3(b) handles moving users from the waiting list to the booked list for a class if there is an available capacity.

When a learner attends a class, their attendance status is updated if marked 'attend', checks and 1 point is added to the user points for reward redemption. The code segment is highlighted in Fig. C4. The script in Fig.C5 checks if the user has enough points to redeem the item, deducts the points, updates the item quantity, records the redemption, and notifies the user of the successful redemption and their updated points balance while displaying appropriate error messages if there are issues retrieving item details.

4.2 Functional Testing

During functional testing, test cases are designed based on the system's functional requirements, outlining the inputs, expected outputs, and steps to validate the functionality. Unit testing is conducted after the development of each module as shown in Appendix D. Each table conducts the testing as follows: Table D1 Testing Result for Login Module, Table D2 Testing Result for Analytic and Profit Module, Table D3 Testing Result for Announcement Module, Table D4 Testing Result for Class Management Module, Table D5 Testing Result for Booking Module, Table D6 Testing Result for Reward Module, D7 Testing Result for Payment Module, Table D8 Testing Result for Review and Feedback Module, Table D9 Testing Result for Profile Management Module

4.3 User Acceptance Testing

The user acceptance test was conducted through a survey on Google Forms to collect feedback from a selected group of users. There are a total of 10 Luna Yoga learners took the user acceptance test. The Likert scale responses indicate acceptance among respondents, with the majority expressing either "strongly agree" or "agree" with various aspects of the UI design. The result of UI testing is shown in Fig. 11. Users were able to rate their satisfaction and identify any areas that required improvement by rating each functionality with a yes or no. The result is shown in Fig. 12, which indicates that all functionalities can be used.

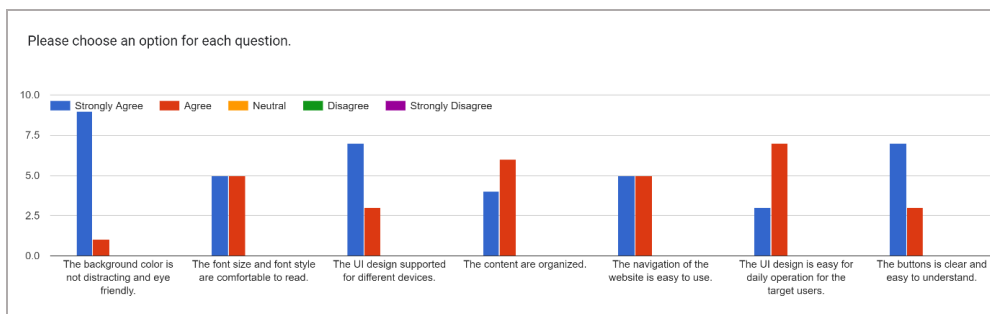


Fig. 11 Result of User Interface Testing

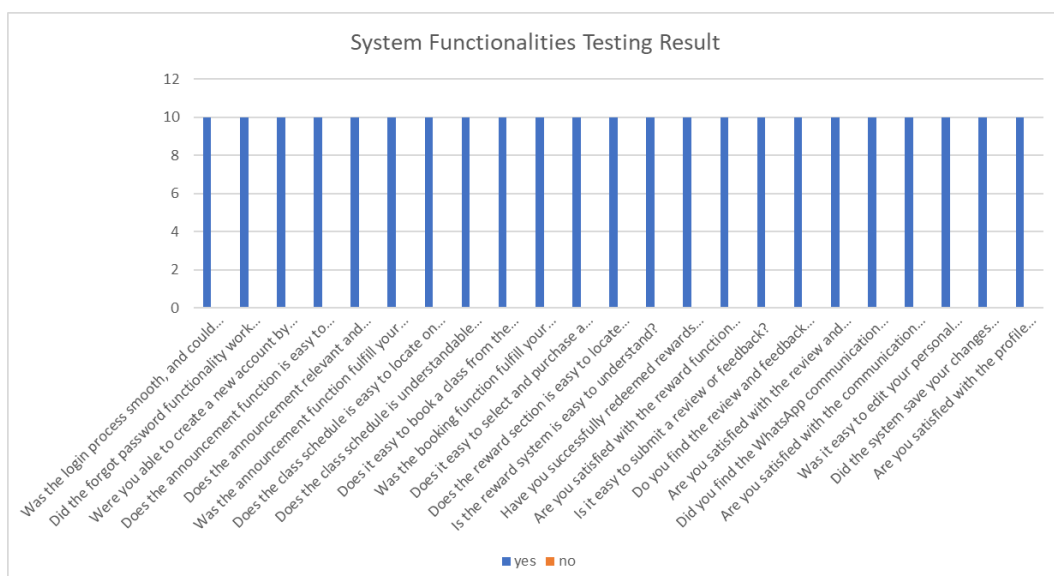


Fig. 12 Result of System Functionalities Testing

5. Conclusion

The deployment of Luna Yoga's web-based administration system represents a big achievement in simplifying class management, improving user experience, and reducing administrative work. The system has transformed Luna Yoga's operating landscape by providing a user-friendly platform for learners to book yoga courses while simplifying administrative work for instructors and administrators. Moving forward, the system's continuous improvement and transformation to keep Luna Yoga up-to-date, fostering a world where individuals thrive in balance and harmony in practising yoga.

There are a few limitations and possibilities for future enhancement. Firstly, the enhancement of the project's integration of an online transaction or wallet API could greatly improve the payment process. Furthermore, adding a notification API to notify learners via WhatsApp when they successfully book a class could improve user interaction and engagement. Lastly, Luna Yoga's web-based management system can enhance security by implementing extra authentication levels like multifactor authentication (MFA), which can greatly improve user account security.

In conclusion, Luna Yoga's web-based management system is an online management system that improves the management of Luna Yoga by transforming the traditional booking and administrative processes into a digital format. This online management system simplifies various features of Luna Yoga's operations, including class scheduling, booking, communication, and administrative tasks.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interest regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

*The authors confirm contribution to the paper as follows: **study conception and design:** Ng Jing Jing, Dr Mubarak Saif Mohsen Al-Absi; **data collection:** Ng Jing Jing; **analysis and interpretation of results:** Ng Jing Jing, Dr Mubarak Saif Mohsen Al-Absi; **draft manuscript preparation:** Ng Jing Jing, Dr Mubarak Saif Mohsen Al-Absi. All authors reviewed the results and approved the final version of the manuscript.*

Appendix A: Gantt Chart

Name	Duration	Start	Finish
FYP 1	29 days	10/1/23 12:00 AM	11/9/23 5:00 PM
FYP briefing	0 days	10/1/23 12:00 AM	10/2/23 5:00 PM
Title proposing	7 days	10/1/23 8:00 AM	10/10/23 5:00 PM
Form A and Form B submiss	7 days	10/11/23 8:00 AM	10/19/23 5:00 PM
Title Defense	1 day	10/22/23 8:00 AM	10/23/23 5:00 PM
Form C submission	5 days	10/24/23 8:00 AM	10/30/23 5:00 PM
Planning	14 days	10/23/23 8:00 AM	11/9/23 5:00 PM
Chapter 1 writing	10 days	10/23/23 8:00 AM	11/3/23 5:00 PM
Chapter 1 submission	5 days	11/3/23 8:00 AM	11/9/23 5:00 PM
Analysis	30 days	11/10/23 8:00 AM	12/21/23 5:00 PM
Chapter 2 writing	6 days	11/10/23 8:00 AM	11/17/23 5:00 PM
Chapter 2 submission	5 days	11/13/23 8:00 AM	11/17/23 5:00 PM
Chapter 3 writing	10 days	11/20/23 8:00 AM	12/1/23 5:00 PM
Chapter 3 submission	5 days	11/27/23 8:00 AM	12/1/23 5:00 PM
Chapter 4 writing	14 days	12/4/23 8:00 AM	12/21/23 5:00 PM
Chapter 4 submission	5 days	12/15/23 8:00 AM	12/21/23 5:00 PM
Design	10 days	12/5/23 8:00 AM	12/18/23 5:00 PM
Create charts and diagrams	3 days	12/5/23 8:00 AM	12/7/23 5:00 PM
Design database	2 days	12/8/23 8:00 AM	12/11/23 5:00 PM
Design system prototype	5 days	12/12/23 8:00 AM	12/18/23 5:00 PM
FYP 1 submission	2 days	12/27/23 8:00 AM	12/28/23 5:00 PM
FYP 1 evaluation	10 days	12/29/23 8:00 AM	1/11/24 5:00 PM
Implementation	88 days	3/17/24 8:00 AM	7/17/24 5:00 PM
Implementation Coding	44 days	3/17/24 8:00 AM	5/16/24 5:00 PM
Implementation MySql with	44 days	5/17/24 8:00 AM	7/17/24 5:00 PM
Unit Testing	3 days	5/18/24 8:00 AM	5/22/24 5:00 PM
User Acceptance Testing	5 days	5/23/24 8:00 AM	5/29/24 5:00 PM
FYP 2	26 days	6/12/24 8:00 AM	7/17/24 5:00 PM
Draft Report Submission	2 days	6/12/24 8:00 AM	6/13/24 5:00 PM
Evaluation	2 days	6/23/24 8:00 AM	6/25/24 5:00 PM
Report Submission	8 days	7/7/24 8:00 AM	7/17/24 5:00 PM

Fig. A1 Project Timeline

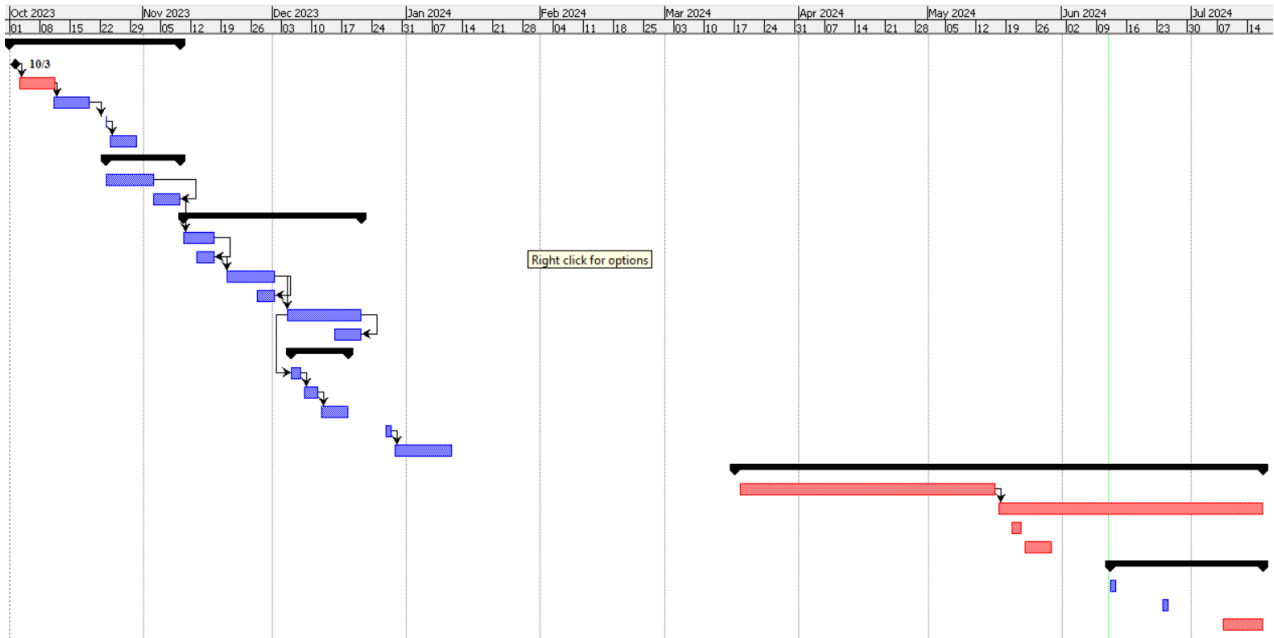


Fig. A2 Gantt Chart

APPENDIX B: Deliverables of System Design

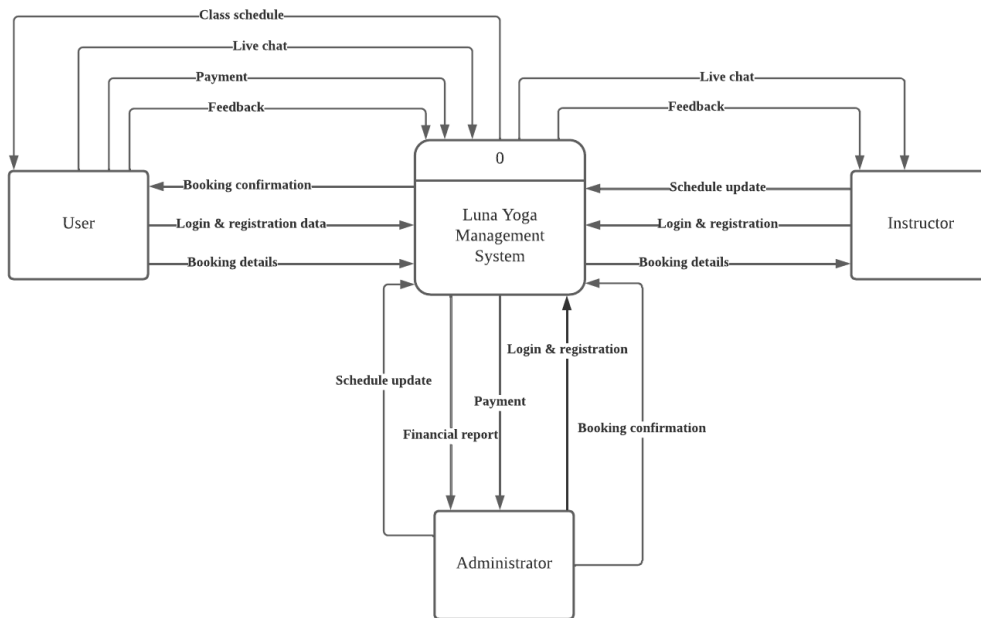
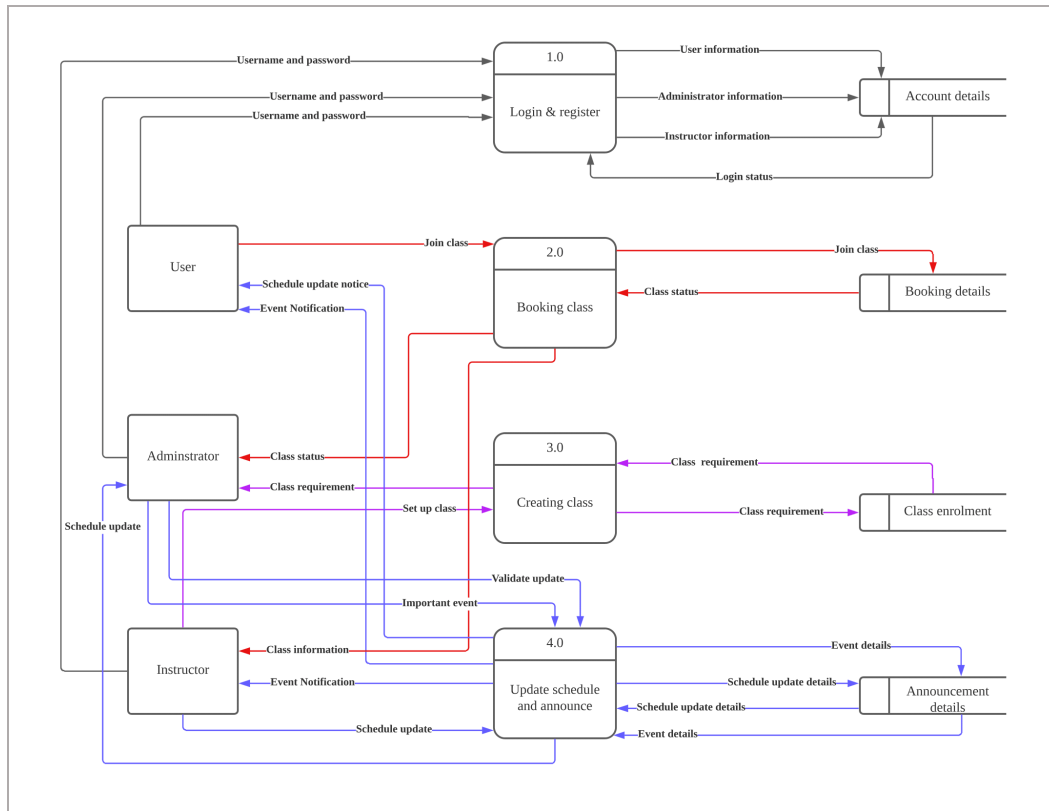
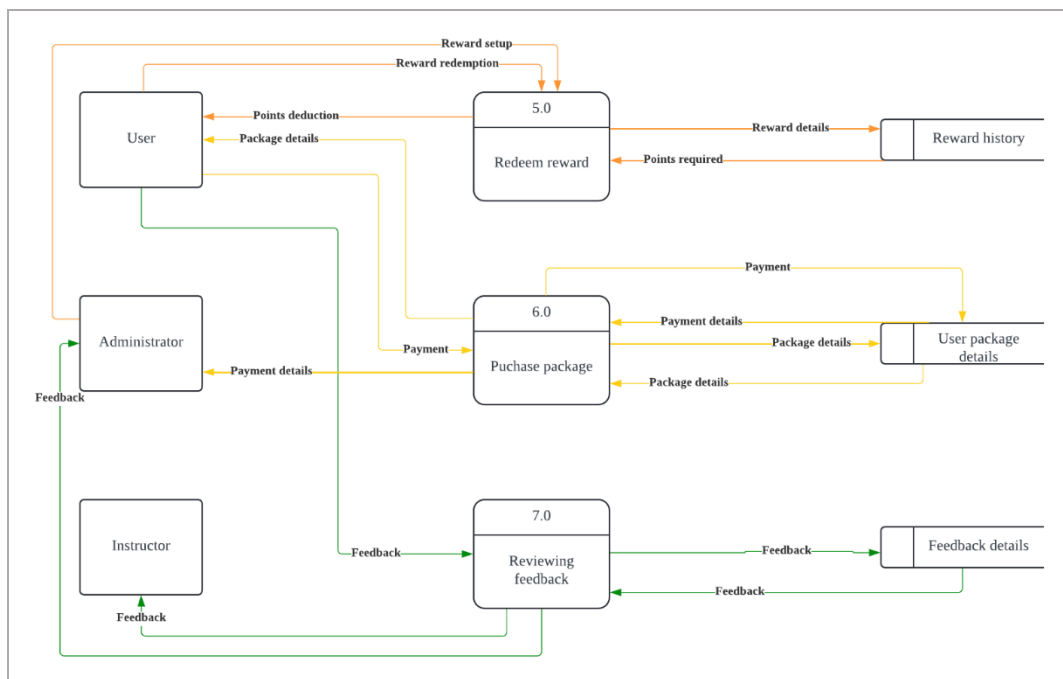


Fig. B1 Context Diagram



(a)



(b)

Fig. B2 DFD Level 0 (a) Process 1.0 until Process 4.0; (b) Process 5.0 until Process 7.0


```

if ($remaining_sessions > 0) {
    $check_waiting_list_query = "SELECT * FROM waiting_list WHERE user_id = $user_id AND schedule_id = $schedule_id";
    $check_waiting_list_result = mysqli_query($conn, $check_waiting_list_query);

    if (mysqli_num_rows($check_waiting_list_result) == 0) {
        $insert_waiting_list_query = "INSERT INTO waiting_list (user_id, schedule_id) VALUES ($user_id, $schedule_id)";
        if (mysqli_query($conn, $insert_waiting_list_query)) {
            echo "<script>alert('You have been added to the waiting list.');";
            echo "window.location='learner_schedule.php';</script>";
        } else {
            echo "Error: " . mysqli_error($conn);
        }
    } else {
        echo "<script>alert('You are already on the waiting list for this class.');";
        echo "window.location='learner_schedule.php';</script>";
    }
} else {
    echo "<script>alert('You do not have enough sessions remaining to join the waiting list for this class.');";
    echo "window.location='learner_schedule.php';</script>";
}
exit();

```

(a)

```

if ($schedule_row['CurrentCapacity'] < $schedule_row['MaxCapacity']) {
    $move_query = "INSERT INTO bookings (user_id, ScheduleID) VALUES ($user_id, $schedule_id)";
    if (mysqli_query($conn, $move_query)) {
        $update_capacity_query = "UPDATE schedule SET CurrentCapacity = CurrentCapacity + 1 WHERE ScheduleID = $schedule_id";
        mysqli_query($conn, $update_capacity_query);

        $remove_waiting_list_query = "DELETE FROM waiting_list WHERE user_id = $user_id AND schedule_id = $schedule_id";
        mysqli_query($conn, $remove_waiting_list_query);

        if ($class_id == 1) {
            $session_type = 'hathaYoga_session';
        } elseif ($class_id == 2) {
            $session_type = 'soundBath_session';
        } elseif ($class_id == 3) {
            $session_type = 'soundGroup_session';
        } elseif ($class_id == 4) {
            $session_type = 'soundBath_session';
        } elseif ($class_id == 5) {
            $session_type = 'soundGroup_session';
        } else {
            header("Location: error.php");
            exit;
        }

        $update_query = "UPDATE user_package SET $session_type = $session_type - 1 WHERE user_id = $user_id";
        if (mysqli_query($conn, $update_query)) {
            echo "<script>alert('User moved to booked list successfully.');";
            echo "window.location='admin_schedule.php';</script>";
        } else {
            echo "Error deducting session: " . mysqli_error($conn);
        }
    } else {
        echo "Error moving user to booked list: " . mysqli_error($conn);
    }
} else {
    echo "<script>alert('No available capacity.');";
    echo "window.location='admin_schedule.php';</script>";
}

```

(b)

Fig. C3 Code Segment (a) Add Waiting List; (b) Manage Waiting List

```

foreach ($_POST['attendance'] as $user_id => $attendance_status) {
    $attendance_status = $attendance_status === 'attend' ? 'attend' : 'absent';

    $update_query = "UPDATE attendance
                    SET attendance_status = '$attendance_status'
                    WHERE user_id = $user_id AND schedule_id = $schedule_id";
    mysqli_query($conn, $update_query);

    if ($attendance_status === 'attend') {
        $check_points_query = "SELECT points FROM user_points WHERE user_id = $user_id";
        $points_result = mysqli_query($conn, $check_points_query);

        if (mysqli_num_rows($points_result) > 0) {
            $increment_points_query = "UPDATE user_points SET points = points + 1 WHERE user_id = $user_id";
            mysqli_query($conn, $increment_points_query);
        } else {
            $insert_point_query = "INSERT INTO user_points (user_id, points) VALUES ($user_id, 1)";
            mysqli_query($conn, $insert_point_query);
        }
    }
}

```

Fig. C4 Code Segment Attendance

```

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['item_id'])) {
    $item_id = $_POST['item_id'];

    $item_query = "SELECT * FROM redeemableitems WHERE item_id = $item_id";
    $item_result = mysqli_query($conn, $item_query);
    if ($item_result && mysqli_num_rows($item_result) > 0) {
        $item = mysqli_fetch_assoc($item_result);
        $points_required = $item['points_required'];

        $points_query = "SELECT points FROM user_points WHERE user_id = $user_id";
        $points_result = mysqli_query($conn, $points_query);

        if ($points_result && mysqli_num_rows($points_result) > 0) {
            $row = mysqli_fetch_assoc($points_result);
            $current_points = $row['points'];

            if ($current_points >= $points_required) {
                $updated_points = $current_points - $points_required;
                $update_points_query = "UPDATE user_points SET points = $updated_points WHERE user_id = $user_id";
                mysqli_query($conn, $update_points_query);

                $update_item_quantity_query = "UPDATE redeemableitems SET item_quantity = item_quantity - 1 WHERE item_id = $item_id";
                mysqli_query($conn, $update_item_quantity_query);

                $redemption_date = date('Y-m-d H:i:s');
                $insert_redemption_query = "INSERT INTO redemptionhistory (user_id, item_id, redemption_date)
                VALUES ($user_id, $item_id, '$redemption_date')";
                mysqli_query($conn, $insert_redemption_query);

                echo "Item redeemed successfully. Your remaining points: $updated_points";
            } else {
                echo "You don't have enough points to redeem this item.";
            }
        } else {
            echo "Error: Unable to retrieve user's points.";
        }
    }
}
    
```

Fig. C5 Code Segment Redeem Reward

APPENDIX D: Functional Testing Result

Table D1 Testing Result for Login Module

No	Test Case Description	Test Result	
		Success	Fail
1	The user can login using valid username, password and role matched to the database	√	
2	The user cannot login using invalid username, password and role that is not matched to the database	√	
3	The system display an error message when user login using invalid username, password and role.	√	
4	The user can receive a reset password token through email if their email exists in the database.	√	
5	The user cannot receive a reset password token through email if their email does not exist in the database.	√	
6	The user can reset a new password using the reset token sent to their email.	√	

Table D2 Testing Result for Analytic and Profit Module

No	Test Case Description	Test Result	
		Success	Fail
1	The administrator can view the total income from yoga and sound baths.	√	
2	The administrator can view the number of users registered monthly	√	
3	The administrator can manage the payment of learners by updating the payment status.	√	
4	The administrator can search for specific learners' payment using their username.	√	
5	The administrator can search for monthly payments.	√	
6	The administrator can print the payment report.	√	
7	The administrator can print the income report.	√	

Table D3 Testing Result for Announcement Module

No	Test Case Description	Test Result	
		Success	Fail
1	The administrator can create, edit, delete a post.	√	
2	The system will prompt an appropriate message to alert the administrator for confirmation.	√	
3	The instructor and learner can view the post	√	

Table D4 Testing Result for Class Management Module

No	Test Case Description	Test Result	
		Success	Fail
1	The administrator can create, edit and delete a class.	√	
2	The system will prompt an appropriate message to alert the administrator for confirmation.	√	
3	The administrator can assign a class type to an instructor.	√	
4	The administrator can create, edit and delete a class schedule.	√	
5	The instructor can view the schedule that assign to them.	√	
6	The learner can view the upcoming classes schedule.	√	

Table D5 Testing Result for Booking Module

No	Test Case Description	Test Result	
		Success	Fail
1	The learner can purchase class sessions to book a class	√	
2	The learner can book a class when they have a remaining class session.	√	
3	The system will prompt a message as booking confirmation.	√	
4	The system will prompt an error message if the learner does not have a specific class remaining session.	√	
5	The learner can view the success booking in learner joined class page.	√	
6	The administrator can view the booking list.	√	
7	The administrator can delete a booking as needed.	√	
8	The learner can join the waiting list when the class is full if they have remaining class session.	√	
9	The learner cannot join the waiting list when the class is full if they do not have remaining class session.	√	
10	The administrator can view the waiting list.	√	
11	The administrator can manage the waiting list by moving the learner in waiting list to class if there is an empty place.	√	
12	The instructor can update learners attendance during the class.	√	

Table D6 Testing Result for Reward Module

No	Test Case Description	Test Result	
		Success	Fail
1	The administrator can insert rewards for redemption.	√	
2	The learner can redeem reward using points by attending a class.	√	
3	The system can deduct the learner’s points based on the reward item points needed.	√	
4	The system will prompt a message to alert the learner if they do not have enough points for redemption.	√	

Table D7 Testing Result for Payment Module

No	Test Case Description	Test Result	
		Success	Fail
1	The learner can purchase the class package successfully.	√	
2	The system will prompt a message for confirmation to purchase the package.	√	
3	The system will display of the payment summary to inform users about the purchased package	√	
4	The system provides cash and online payment for payment methods	√	
5	The system displays an appropriate message for the learner upon confirming payment.	√	

Table D8 Testing Result for Review and Feedback Module

No	Test Case Description	Test Result	
		Success	Fail
1	The learner can send feedback by filling up a feedback form.	√	
2	The administrator and instructor can view feedback from learners.	√	
3	The administrator and instructor can mark feedback as read.	√	

Table D9 Testing Result for Profile Management Module

No	Test Case Description	Test Result	
		Success	Fail
1	All users can edit their profiles to update their personal information.	√	
2	The administrators can edit learners' profiles and instructors' profiles.	√	
3	The administrators can delete the instructors' profiles when the instructors leave the yoga studio.	√	
4	The administrator can add a new learner and a new instructor.	√	
5	The system will prompt a suitable message for create, edit and delete profile confirmation.	√	

References

- [1] Pearson Deutschland, "Management Information Systems | Pearson eLibrary," 2021. <https://elibrary.pearson.de/book/99.150005/9781292403571>
- [2] A. B.- Shaqiri, "Management Information system and Decision-Making," *Academic Journal of Interdisciplinary Studies*, Jun. 2014, doi: 10.5901/ajis.2014.v3n2p19.
- [3] Omesuh, Barry. (2019). Electronic Booking Systems: Exploring Black Barbershops in East London. 10.13140/RG.2.2.27100.00646.
- [4] S.-J. Ernst, A. Janson, C. Peters, and J. M. Leimeister, "Understanding IT-Culture conflicts to Drive successful technochange Projects A case study," *Social Science Research Network*, Jan. 2017, doi: 10.2139/ssrn.3159129.
- [5] "LIFE Hot Yoga – Kuala Lumpur, Malaysia – Far infrared hot yoga studio offering variety of hot yoga classes taught by certified instructors. Full facilities; Ample sheltered carpark in a mall. Read more." <https://lifehotyoga.com.my/>
- [6] "Jalyoga – Live More with Jal." <https://www.jalyoga.com.my/>
- [7] karma yoga school - do good. everyday., "karma yoga school • karma yoga," *Karma Yoga*, May 12, 2024. <https://karmayoga.my/>
- [8] F. Heriyanti and A. Ishak, "Design of logistics information system in the finished product warehouse with the waterfall method: review literature," *IOP Conference Series. Materials Science and Engineering*, vol. 801, no. 1, p. 012100, May 2020, doi: 10.1088/1757-899x/801/1/012100.
- [9] N. Raptama and N. Riki, "Analysis and design of information systems for Make-Up using the Waterfall Methodology," *bit-Tech*, vol. 5, no. 2, pp. 54–66, Dec. 2022, doi: 10.32877/bt.v5i2.553.