End Project Report

Project name:	Fitness Club Management System
Issue date:	30/03/2023
Group 6:	Lydia Rong Ryan Chen Ryan Ko Yang Liu Zichen He

Purpose of the Project

The aim of the Fitness Club Management System project is to create a Proof of Concept (POC) to automate the current manual system used to manage 1000 members and 10 personal trainers at Lincoln Fitness. The project follows Agile and Scrum fundamentals to ensure that the project is delivered efficiently and collaboratively, with regular feedback from the Product Owner and stakeholders. This approach provides flexibility to adapt to changing requirements and priorities, ensuring that the system remains relevant and valuable to Lincoln Fitness Club and its members. The project is divided into three weekly sprints, with each sprint delivering a working software increment that adds value to the system.

- The first sprint is primarily aimed at developing the core functionalities that allow all users to log in, creating a skeleton user interface, and displaying database information. This sprint sets the foundation for the system and ensures that it is functional and user-friendly.
- The second sprint is focused on user stories that require changes to the database, such as making group class bookings, updating member profiles, and managing personal trainer sessions. This sprint aims to improve the system's functionality by adding new features and enhancing existing ones. By the end of this sprint, the system will be more comprehensive, and users will be able to perform a wider range of tasks.
- The final sprint is focused on delivering additional functionalities, such as sending messages, viewing trainer schedules, and displaying the training calendar. This sprint is aimed at enhancing the system's capabilities, providing users with more features to make their experience smoother and more efficient. By the end of the third sprint, the Fitness Club Management System will be a comprehensive, user-friendly, and effective system that meets the needs of Lincoln Fitness Club and its members.

The Proof of Concept is designed to provide an efficient and effective way for Lincoln Fitness to manage its members and trainers while automating many of the manual tasks currently performed by staff. The system can keep track of member subscriptions, attendance, personal trainer sessions, and group exercise classes. Members' information can be stored, allowing them to update their profiles and book sessions with personal trainers and exercise classes. By automating tasks such as processing payments, managing member details, and generating financial reports, the system will save time and reduce errors for the staff at Lincoln Fitness Club.

The project has three types of users: admin/manager, trainer, and member. The admin/manager has access to all the features of the system, including managing member details, processing payments, viewing attendance, generating financial reports, and sending reminders and updates to members. Trainers can view and update their profiles and trainees' information. Trainers can also add, modify, and delete their own group fitness classes and personal trainer sessions. Members can view and update their profiles, book sessions with trainers, book exercise classes, and make payments.

Overall, the Agile and Scrum approach ensures that the project is delivered efficiently, with a high level of collaboration and transparency between the project team and stakeholders. This approach also provides flexibility to adapt to changing requirements and priorities, allowing the project team to deliver a system that meets the needs of Lincoln Fitness Club and its members. In summary, the project team has delivered a proof of concept for the Fitness Club Management System that will automate many manual tasks and provide an efficient way for Lincoln Fitness to manage its members and trainers.

User Stories

The three tables below illustrate the user stories the team had completed each sprint. The team had completed all the tickets planned from the release planning. However, the user stories that were originally listed from the release planning may appear dissimilar to those presented in these tables. This is due to the division of larger user stories into smaller ones to enhance workflow during the sprint and enable early detection and resolution of any issues. This methodology has proven to be effective in facilitating the delivery of high-quality results within predetermined timelines.

Sprint 1

User Story	Description	Acceptance Criteria
Admin/Manager - Admin Portal	As an Admin/Manager of Lincoln Fitness I want to be able to access the admin portal so that I can perform all the required tasks	AC1: After I log into my admin account, an admin portal page should appear with these buttons - Members, Trainers, Trainer Sessions, Exercise Sessions, Financial Report and Notifications AC2: Once logged in, my name should appear on the top right corner with a log out button to log out.
Admin/Manager - Register new members	As an Admin/Manager of Lincoln Fitness I want to be able register new members so that I can help members who have difficulties to register by themselves	AC1: A "Register New Member" button on the members page in the Admin Portal. AC2: When this button is clicked, a pop-up will appear to say "Registration is successful. Please click the back button to return to the admin portal"

Admin/Manager - View personal trainer's classes	As an Admin/Manager of Lincoln Fitness I want to be able to view a list of classes offered by each trainer so that I can make informed decisions about scheduling and staffing	AC1: The system should provide a list of trainers along with the personal training sessions they offer. AC2: The list should be searchable by trainer, class name and class time. AC3: The list should show the remaining capacity for each class. AC4: The list should show the date and time each class is offered.
Admin/Manager - Update member details	As an Admin/Manager I want to be able to add, update and delete member information so that I can store and manage member details	AC1: Admin can add a new member by submitting basic member information AC2: Admin can search for a member and edit member details, including name, DOB, phone number, emails, address and emergency contact AC3: A button provides for members, when he clicked it , a confirmation window will has him to confirm, after member confirmed, system will change the member to be inactive, and force him logout.
Admin/Manager - Generate financial report for the club	As an Admin/Manager I want to be able to generate financial reports for the club so that I can track revenue and expenses	AC1: When I click on the Financial Report button on the admin portal, a filter with start date and end date should appear, along with report type options to generate financial reports, including revenue, expense, and balance. AC2: The revenue report should include two columns - revenue type and amount, with a total figure displayed below the table. The report should display the date range selected in the filter above the table. AC3: The expense report should include two columns - expense type and amount, with a total figure displayed below the table. The report should display the date range selected in the filter above the table. AC4: The balance report should include the revenue report and the expense report with a balance figure displayed below the tables. The report should display the date range selected in the filter above the table.
Member - member portal	As a member of Lincoln Fitness I want to be able to access the member portal so that I can perform all the required tasks	AC1: After I log into my member account, a member portal page should appear with these buttons - Profile, Trainers, Trainer Sessions, Group Classes, My Bookings and Notifications.

		AC2: Once logged in, my name should appear on the top right corner with a log out button to log out.
Member - Sign-up Process for Fitness Centre Website	As a member of Lincoln Fitness I want to be able to register for an account so that I can access the system and perform the necessary tasks.	AC1: The login page should have a Register Now button AC2: The registration form should contain personal information fields, including First Name, Family Name, Email, Gender and Date of Birth AC3: The registration form should contain payment information fields, including subscription, card number, card holder, expiration month, expiration year and CVV. AC4: The registration form should contain a Join Now button. AC5: The log in page should have an About us button. When clicked, a description should appear with a join us button which redirects to the registration form detailed in AC2, AC3 and AC4 AC6: Once joined, the member should be able to login with the registered email address with account type selected as Member. AC7: If any required fields are left blank when the Join Now button is pressed, an error message will be displayed stating "Please ensure that all required information is filled out correctly." AC8: Once joined, a pop-up window will be displayed stating "Welcome to Lincoln Fitness. Please click on the back button below the LU Fitness logo to navigate back to the Portal" AC9: A back button to allow the users to navigate back to the sign in page.
Member - Login Process for Fitness Centre Website	As a member of Lincoln Fitness I want to have the ability to log in so that I can access the system and perform the necessary tasks.	AC1: The login page should have an email login option AC2: Registered members can only login when the member account type is selected AC3: When a registered member selects the wrong account type to log in, an error message should appear on the login page. The error message should read "Login details incorrect. Please try again." AC4: Once logged in, my name should appear on the top right corner with a log out button to log out.

Member - View My Profile	As a member of Lincoln Fitness I want to be able to view my profile, so that I can access and view my personal information	AC1: When I click on the Profile button on the Member portal, it should direct me to the profile page. AC2: The profile page should contain personal information fields, including First Name, Family Name, Email, Gender and Date of Birth and Address AC3: The profile page should contain payment information fields, including subscription, card number, card holder, expiration month, expiration year and CVV. AC4: The back button redirects the user back to the member portal
Member - Update My Profile	As a member of Lincoln Fitness I want to be able to update my profile, so that I can keep my personal information up to date	AC1: The profile page should display all my existing personal information AC2: The profile page should allow personal information to be modified and saved when the update button is pressed AC3: A pop up should appear once the changes are saved and redirect back to the member portal AC4: The pop-up message should read "Your profile information has been updated successfully. Please click on the back button below the LU Fitness logo to navigate back to the Portal".
Member - View a list of trainers	As a member of LU fitness I want to view a list of trainers so that I can see trainers' information and attend their training sessions or exercise classes	AC1: When I click on the Trainers button on the admin portal, a table with a list of trainers will appear. AC2: The trainers table is to contain columns including Name, Gender, Qualification, Summary and Trainer Schedules AC3: Under the Trainer Schedules Column, each row should display "Schedules" which links to a page that displays the trainer's training sessions and exercise classes.
Trainer - Trainer Portal	As a trainer of Lincoln Fitness I want to be able to access the trainer portal so that I can perform all the required tasks	AC1: After I log into my trainer account, a trainer portal page should appear with these buttons - Profile, Member and My Schedule AC2: Once logged in, my name should appear on the top right corner with a log out button to log out.
Trainer - View and update profile	As a Trainer of LU Fitness I want to be able to view and update my profile so that members can know me better, and see what	AC1: When I click on the Profile button on the Trainer portal, it should direct me to the profile page. AC2: The profile page should contain personal information fields, including First Name, Family

	specialised training I can provide to them	Name, Email, Phone, Gender, Summary and Qualifications AC3: The profile page should display all my existing trainer information AC4: The back button redirects the user back to the trainer portal AC5: The profile page should allow trainer information to be modified and saved when the update button is pressed AC6: The pop-up message should read "Your profile information has been updated successfully. Please click on the back button below the LU Fitness logo to navigate back to the Portal".
Trainer - View trainee's information	As a Trainer of Lincoln Fitness I want to be able to view trainee's information so that I can know more about trainees.	AC1: When I click on "MySchedules" in the trainer portal, will show a list of my group fitness class and personal trainer sessions. AC2: The list will include details for start time, end time, capacity and remaining capacity. AC3: For each session or class, trainers can view a list of members who have booked this session or class by clicking the "view member" link. AC4: The member list will contain details on the member's name, gender, age, address, email, subscription status and booking history.

Sprint 2

User Story	Description	Acceptance Criteria
Admin/Manager - Create personal trainer sessions and group exercise classes	As an Admin/Manager of Lincoln Fitness I want to be able to create personal trainer sessions and group exercise classes so that I can let other users know about the details of these sessions and classes	AC1: When I click on the Exercise Classes item in the admin portal, a button called "Add a Group Exercise Class" is located on the bottom of the page. AC2: When the "Add a Group Exercise Class" button is clicked, it will redirect to a page titled "Adding a Group Exercise Class". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC3: When the "Add Group Fitness Class" button is pressed, it will redirect the user back to the Group Exercise Classes page, showing a list of classes, including the class that has just been added.

		AC4: When I click on the Trainer Sessions item in the admin portal, a button called "Add a Personal Training Session" is located on the bottom of the page. AC5: When the "Add a Personal Training Session" button is clicked, it will redirect to a page titled "Adding a Personal Training Session". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC6: When the "Add a Personal Training Session" button is pressed, it will redirect the user back to the Personal Trainer Sessions page, showing a list of classes, including the class that has just been added.
Admin/Manager - Delete personal trainer sessions and group exercise classes	As an Admin/Manager of Lincoln Fitness I want to be able to delete personal trainer sessions and group exercise classes so that I can move sessions and classes that are no longer available	AC1: When I click on the Exercise Classes item in the admin portal, a button called "Delete" is shown for each exercise class. AC2: When the "Delete" button is clicked, it will delete the exercise class. AC3: When I click on the Trainer Sessions item in the admin portal, a button called "Delete" is shown for each personal training session. AC4: When the "Delete" button is clicked, it will delete the personal training session.
Admin/Manager - Process payment from members	As an Admin/Manager I want to be able to process payments from members so that I can receive payments from my customers	AC1: Payment details are required when member sign up AC2: First payment and subscription will be processed on signing up AC3: Once registering and payment is settled, the customer can immediately have access to the booking system, and the subscription status should also be updated.
Admin/Manager – Process Payment Automatically and Daily	As an Admin/Manager I want to be able to have a way to renew the subscription and process payment automatically so that I can sit and relax	AC1: All active members will renew his/her subscription automatically AC2: only subscriptions that due in 7 days need to be automatically renew AC3: subscriptions history and payment history need to be inserted well after renewing.
Admin/Manager - Unsubscribe existing members	As an Admin/Manager I want to be able to delete existing members so that I can remove member information from the database when requested by members	AC1: A "Unsubscribe" button under the "Unsubscribe| Update" column of the member table for each member row, on the members page in the Admin Portal. AC2: When this button is clicked, it will unsubscribe member from the database and prevent the member from login in.

		AC3: When this button is clicked, under the "Unsubscribe| Update" column, the status should change from Subscribe to Unsubscribe.
Admin/Manager - View members booking history	As an Admin/Manager I want to be able to view members' booking history so that I can see member has attended which group classes and training sessions	AC1: In member's detail page, when admin/manager clicks on the booking history button of each member, it will jump to a member's booking history page. AC2: The member's booking history page should display a table including class name, class type (group class or trainer session) and class date/time, with member's name on the top of the table.
Admin/Manager - View subscription status/history	As an Admin/Manager I want to be able to view subscription status and history of members so that I can keep track of members' subscription.	AC1: In the members details page, when admin/manager clicks on the subscription status of each member, it will jump to a member subscription page. AC2: The member subscription page should display member's name and subscription status on the top of the page (active or expired). AC3: The member subscription page should display a table of member's subscription history including three columns - date of subscription, price, and subscription type (weekly or monthly). AC4: Admin/manager should be able to view a reminder on this page if member's subscription will expire in 7 days. If the membership has already expired, the subscription status of AC2 will just display "Expired" instead of a reminder on this page.
Member - View list of group classes	As a member I want to view all exercise sessions information so that I can find group class that I like, and make decisions about scheduling etc.	AC1: The system should provide a list of trainers along with the group class they offer. AC2: The list should be searchable by trainer, class name and class time. AC3: The list should show the remaining capacity for each class. AC4: The list should show the date and time each class is offered.
Member - Make a payment for gym subscription	As a member I want to be able to pay for my monthly subscription. so that I can start using the member system.	AC1: In the registration form, the member should be able to see the amount they need to pay for monthly subscription. AC2: The member should be able to enter the payment details, including card number, expiration date, and CVV code. AC3: The system should process the payment and update the member's subscription status.

		AC4: The member should be able to log in once subscription is completed.
Member - Book and pay for training session	As a member I want to be able to make a payment so that I can book and pay for the specialised training sessions.	AC1: The member should be able to view the available specialised training sessions. AC2: The member should be able to see the additional fees for each training session. AC3: The member should be able to select the training session they want to attend. AC4: The member should be able to enter the payment details, such as card number, expiration date, and CVV code. AC5: The system should process the payment and the member can see his/her booking in My booking.
Member -A Calendar to see the trainer sessions of this week	As a member I want to be able to see the group sessions in this week so that I can see what is interesting and what suits my schedule	AC1: All Group sessions from this Monday to this Sunday need to be shown. AC2: Sessions in each day should have different colour to be distinguished easily. AC3: Sessions need to be display correctly displayed in which day in a week and which time from a day. AC4: Sessions can be click and process booking. AC5: Next Week's sessions and Previous Sessions need to be displayed as well

Sprint 3

User Story	Description	Acceptance Criteria
Admin/Manager - Modify personal trainer sessions and group exercise classes	As an Admin/Manager of Lincoln Fitness I want to be able to modify details for personal trainer sessions and group exercise classes so that I can keep information for these sessions and classes up to date.	AC1: When I click on the Exercise Classes item in the admin portal, a button called "Modify" is located for each exercise class AC2: When the "Modify" button is clicked, it will redirect to a page titled "Updating a Personal Trainer Session". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC3: When the "Modify Session" button is pressed, it will redirect the user back to the Group Exercise Classes page, showing a list of classes, including the class that has just been updated.

		AC4: When I click on the Personal Trainer session item in the admin portal, a button called "Modify" is located for Personal Trainer session. AC5: When the "Modify" button is clicked, it will redirect to a page titled "Updating a Personal Trainer session". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC6: When the "Modify Class" button is pressed, it will redirect the user back to the Personal Trainer page, showing a list of personal trainer sessions, including the personal trainer session that has just been updated.
Admin/Manager - Send news/updates to member	As an Admin/Manager I want send news/updates to member so that all the latest course promotions or cancelation could be known ASAP by our members.	AC1: News/Updates will have a title to let admin input AC2: Users can be selected one or All AC3: News/Updates will have a text area as content to let admin input. AC4: A send button and cancel but will allow admin to send or cancel the editing. AC5: Once send button clicked, system will check if the title, content, and receivers are all input or selected AC6: Only all input requires are satisfied then a window will pop up to ask Admin are you sure to send it? Yes, then send; No, then cancel, back to editing.
Admin/Manager - Sends reminders to member when their subscription is due	As an Admin/Manager I want to be able to send reminders to members when their subscription is due so that they are prompted to renew their membership and we can maintain accurate revenue projections	AC1: The manager should be able to see which members' subscription is due in the member's list. AC2: The manager should be able to send emails to the member's email address by clicking mail icon on the list. *Not working on free PythonAnywhere account.
Admin/Manager - View Popular Exercise Class	As an Admin/Manager I want to be able to view class popularity so that I can make data-driven decisions on class schedule	AC1: Admin can view the popular exercise class list through “The popular Exercise Class” in the admin portal. AC2: The page will show the number of members enrolled in each class which will allow for determining the popularity of each class AC3: There is a time filter that allows the admin views popular classes within a selected time period.

		AC4: The list is sortable by clicking an icon near the column title.
Admin/Trainer - members list display in pagination	As a Manager/Trainer when I go to members list feature, I want to see the data of member list is paginated so that so that not all the members are loaded in one time	AC1: Only ten members loaded each time. And the rest of members are loaded by button provided. AC2: Initially display first ten members. For example, if there are 25 members in our gym, so initially the first page will load first ten members. AC3: Click previous button, the data wont refresh if user at the first page, it works only when user at a second or later page. AC4: Click next page button, the data will be refreshed to load another 10 members. The page won't refresh user at the last page.
Admin/Trainer- member list can fuzzy search by member name	As a Manager/Trainer when I go to members list feature, I want to search by member's name, so that I could find the member in quick	AC1: All data will be displayed If nothing input and click the search button AC2: If letters are input and button is clicked, the data will be filtered by the input letters, not case sensitive, both works with first name and family name
Member - Book an Exercise Class	As a Member I want to be able to book a group class so that I can secure my spot and attend the class.	AC1: The member should be able to book a spot in the class. AC2: The member should be able to see if the class is fully booked or if there are still available spots. AC3: The system should confirm the booking and show the member their booked class. AC4: The system should reduce the available spots for the class by one. AC5: The member should not be able to book a spot in a class that is already fully booked.
Member - Cancel my bookings	As a Member I want to be able to cancel my booking for a class before the class starts. so that I can free up the slot for other members and avoid being charged for a no-show.	AC1: In member portal after login, when I click on the "My bookings" button, it will jump to a page of member's booking history which should display a list of all booked class. AC2: For each upcoming class, I should be able to see a "Cancel" button if the class will start in more than 24 hours (personal session) or 4 hours (group class). AC3: I should be able to click on the 'Cancel' button to withdraw from a class, which should be allowed up to a certain amount of time before the class starts as described in AC2.
Member - View my bookings	As a Member I want to be able to check my booking history.	AC1: There will be a "My bookings" button in the member portal, it will jump to the member's booking group and personal sessions records.

	so that I can keep track of my past workouts and future classes/sessions.	AC2: This booking page displays a table of booked classes, with the class name, trainer name, and start/end time.
Members- Receive and read the Notifications	As a member I want to receive the notifications from our club so that all latest schedule changes, promotions could be delivered to me.	AC1: All notifications that sent to me must to be correctly displayed in a list AC2: All notifications that not sent to me must not be displayed AC3: Notifications details need to be displayed correctly after clicked one notification from the list.
Trainer - Create personal trainer sessions and group exercise classes	As a Trainer of Lincoln Fitness I want to be able to create personal trainer sessions and group exercise classes so that I can let other users know about the details of these sessions and classes	AC1: When I click on the Exercise Classes item in the trainer portal, a button called "Add a Group Exercise Class" is located on the bottom of the page. AC2: When the "Add a Group Exercise Class" button is clicked, it will redirect to a page titled "Adding a Group Exercise Class". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC3: When the "Add Group Fitness Class" button is pressed, it will redirect the user back to the Group Exercise Classes page, showing a list of classes, including the class that has just been added. AC4: When I click on the Trainer Sessions item in the trainer portal, a button called "Add a Personal Training Session" is located on the bottom of the page. AC5: When the "Add a Personal Training Session" button is clicked, it will redirect to a page titled "Adding a Personal Training Session". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC6: When the "Add a Personal Training Session" button is pressed, it will redirect the user back to the Personal Trainer Sessions page, showing a list of classes, including the class that has just been added.
Trainer - Modify personal trainer sessions and group exercise classes	As a Trainer of Lincoln Fitness I want to be able to modify details for personal trainer sessions and group exercise classes so that I can keep information	AC1: When I click on the Exercise Classes item in the trainer portal, a button called "Modify" is located for each exercise class AC2: When the "Modify" button is clicked, it will redirect to a page titled "Updating a Personal Trainer Session". This page will include Session

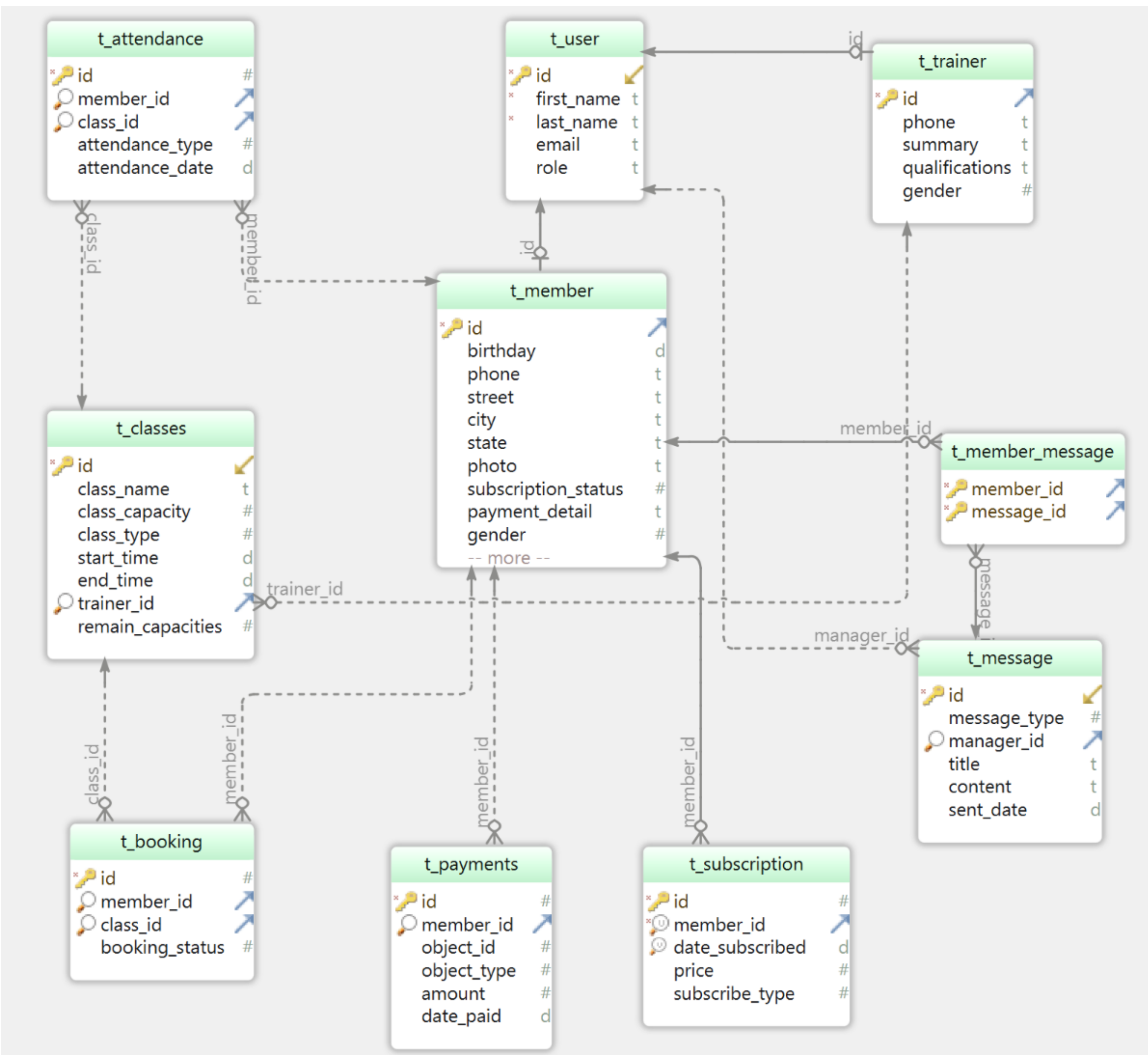
	for these sessions and classes up to date.	Name, Trainer, Date, Start time, End time and Capacity. AC3: When the "Modify Session" button is pressed, it will redirect the user back to the Group Exercise Classes page, showing a list of classes, including the class that has just been updated. AC4: When I click on the Personal Trainer session item in the admin portal, a button called "Modify" is located for Personal Trainer session. AC5: When the "Modify" button is clicked, it will redirect to a page titled "Updating a Personal Trainer session". This page will include Session Name, Trainer, Date, Start time, End time and Capacity. AC6: When the "Modify Class" button is pressed, it will redirect the user back to the Personal Trainer page, showing a list of personal trainer sessions, including the personal trainer session that has just been updated.
Trainer - Delete personal trainer sessions and group exercise classes	As a Trainer of Lincoln Fitness I want to be able to delete personal trainer sessions and group exercise classes so that I can move sessions and classes that are no longer available	AC1: When I click on the Exercise Classes item in the trainer portal, a button called "Delete" is shown for each exercise class. AC2: When the "Delete" button is clicked, it will delete the exercise class. AC3: When I click on the Trainer Sessions item in the trainer portal, a button called "Delete" is shown for each personal training session. AC4: When the "Delete" button is clicked, it will delete the personal training session.

End Project Report

Solution

Data Model

The database consists of ten tables, including t_user, t_member, t_trainer, t_subscription, t_payments, t_attendance, t_classes, t_message, t_member_message, and a few more. Each table contains data about the gym, including users, members, trainers, subscriptions, payments, attendance, classes, and messages. The t_user table stores information about all gym users, while the t_member and t_trainer tables store data specific to members and trainers, respectively. The t_subscription, t_payments, and t_attendance tables keep track of subscriptions, payments, and attendance, respectively. The t_classes table stores information about the classes offered by the gym, while the t_message and t_member_message tables contain data related to messages sent between members and the gym.



End Project Report

1. **Table: t_user**

The table contains information about all gym users, including manager, members and trainers. It contains fields for user ID, name, email, DOB, and role.

- id (Primary Key): Unique identifier for each user.
- first_name: The first name of the user.
- last_name: The last name of the user.
- email: The email address of the user, this will be used for login.
- role: the role of each user (0- manager, 1- trainer, 2- member)

2. **Table: t_member**

The Member table stores information about all the members registered with the fitness club. It includes fields for member ID, date of birth, phone number, address, photo, subscription, payment, emergency contact and gender.

- id (Primary Key): Unique identifier for each member.
- birthday DATE: the birthday date of the member.
- phone: The phone number of the member.
- street: The street address of the member.
- city: The city of the member.
- state: The state of the member.
- subscription_status: The current status of the member ‘s subscription (0- normal, 1- unsubscribed, 2- NA).
- payment_detail: Member’s saved bank information for automatic transfer.
- gender: The gender of each member (0- male, 1- female)

3. **Table: t_trainer**

The Trainer table stores information about all the trainers in the fitness club. It includes fields for trainer ID, photo, summary, qualification and gender.

- id (Primary Key): Unique identifier for each trainer.
- phone: The phone number of the trainer.
- summary: A summary of the trainer's previous working experience.
- qualifications: Information about the trainer's qualifications, such as certifications and areas of expertise.
- gender: The gender of each trainer (0- male, 1- female,2-neutral)

4. **Table: t_subscription**

The Subscriptions table contains information about the gym's subscriptions. It includes fields for the subscription ID, member ID, date subscribed, subscription type, and monthly price.

- id (Primary Key): Unique identifier for each subscription.
- member_id (Foreign Key): The ID of the member who subscribed to the subscription.
- date_subscribed: The date the member subscribed to the subscription.
- subscription_type: The type of subscription (0- weekly, 1- monthly).
- monthly_price: The monthly price of the subscription.

5. **Table: t_payments**

The Payments table contains information about each transaction made by members. It includes fields for the payment ID, member ID, object ID, object type, amount, and date paid.

- id (Primary Key): Unique identifier for each payment.
- member_id (Foreign Key): The ID of the member who made the payment.
- object_id (Foreign Key): The ID of the object (subscription or class booking) for which the payment was made.
- object_type: The type of object for which the payment was made (0- subscription, 1- class booking).
- amount: The amount paid.
- date_paid: The date the payment was made.

6. **Table: t_attendance**

The Attendance table contains information about members' attendance in classes. It includes fields for the attendance ID, member ID, trainer ID, class ID, attendance type, and attendance date.

- id (Primary Key): Unique identifier for each attendance record.
- member_id (Foreign Key): The ID of the member who attended the class.
- trainer_id (Foreign Key): The ID of the trainer who taught the class.
- class_id (Foreign Key): The ID of the classes including group or personal ones.
- attendance_type: The type of attendance including group class, personal training session and using gym only. (0- class, 1- gym, 2- session).
- attendance_date: The date of the attendance.

7. **Table: t_classes**

The Classes table stores information about the exercise classes offered by the fitness club. It includes fields for class ID, class name, trainer ID, class capacity, class start and end time, and class type.

- id (Primary Key): A unique identifier for each class.

- class_name: The name of the class.
- class_capacity: The maximum number of members who can attend the class (1- session, 30- class).
- class_type: The type of the class including group class and personal training session (0- class, 1- session)
- start_time: The start time of the class.
- end_time: The end time of the class.
- trainer_id: The ID of the trainer who leads the class.

8. **Table: t_message**

The Message table stores messages sent between members and the fitness club management. It includes fields for message ID, message type, manager ID, message content, and sent date.

- id (Primary Key): A unique identifier for each message.
- message_type: The type of the message including gym news and updates, as well as reminder when member's subscription is due (0- news, 1- updates, 2- reminder).
- manager_id: The ID of the user who sent the message.
- content: The text content of the message.
- sent_date: The date and time the message was sent.

9. **Table: t_member_message**

The member_message table acts as a bridge table between the user (member) and message tables, as it represents a many-to-many relationship between them. It stores data about which members received which messages.

- member_id (Primary Key): A unique identifier for each member.
- message_id (Primary Key): The id of the message sent.

10. **Table: t_booking**

The Booking table serves as a bridge table that tracks member bookings for group classes and personal training sessions. It includes fields for booking ID, member ID, class ID and booking status.

- id (Primary Key): A unique identifier for each booking recoed.
- member_id: The ID of the member who made the booking.
- class_id: The ID of the class that was booked.
- booking_status: The status of the booking (0- cancelled, 1- attended, 2- booked).

GUI Design

The GUI design of the system includes multiple web pages for different functionalities. The landing page presents options for users to log in or sign up for an account. The login box is displayed in the center of the page, which draws the user's attention to it. It has a transparent background with a white border, giving it a clean look. After logging in, the portal has a simple and intuitive design that allows users to quickly access different features of the system. The portal page is organized as a menu with each option represented by a circle. The circles are evenly spaced and have a good size that allows users to click easily on them. On hover, each circle expands slightly, indicating to the user that it is clickable and also creating a visual effect that gives feedback to the user.

The colour scheme of our website includes blue, gold and shades of grey. The blue background of the header section provides a visual connection to the LU gym logo, while the grey background of the main content section provides a neutral backdrop that is easy on the eyes. The use of gold and white for the text and other elements ensures that they stand out and are easy to read. The combination of these colours creates a visually appealing website design and helps to draw attention to the most important parts of the page.

The interface design follows a flat design style, which is characterized by the use of minimalistic and simple design elements such as flat colours, clean typography, and simple shapes. This style aims to create a clean and uncluttered look that emphasizes usability and functionality over aesthetic embellishments. This is achieved through the use of solid colours for all the page elements and the lack of any three-dimensional or realistic visual effects. The colour palette is limited to gold and white. The typography is clean and easy to read, with a simple sans-serif font used throughout.

Overall, our GUI design is clean and simple, with a consistent layout and colour scheme throughout the pages. Navigation is easy, with clear menus and options for users to move between pages. The use of icons and visual aids, such as the edit, delete and sort icons in the tables, help to make the system intuitive and easy to use.

Validation and Verification

To ensure the Fitness Club Management System meets the requirements and standards set by PO, the project team conducted thorough validation and verification testing throughout the project development lifecycle.

Validation and Verification are important steps in software development that ensure the system meets the functional requirements and, in this case, adheres to PO's standards of the program. To achieve this, the project team followed the acceptance criteria of each user story and tested whether the system met those criteria. Observations were recorded, notes were taken, and each test was marked as passed or failed based on whether the system met acceptance criteria. This helped to ensure that the system met the functional requirements of the project.

The project team used Github to plan and organize the testing progress. Github Issues were created for each user story, and the acceptance criteria for that story were listed in the issue. As tests were conducted for each user story, they were logged in the corresponding Github Issue. This made it easy for team members to see which tests had been conducted and the results of those tests. Additionally, Github Projects were used to organize and track the testing progress for each sprint.

In addition to functional testing, regression testing was performed to verify the system after each sprint to ensure that new features and updates did not break existing functionality. This helped to ensure that the system met the required specifications of the program.

Furthermore, usability testing was conducted to evaluate the system's ease of use, user-friendliness, and overall user experience. The project team members were asked to act as test users and provide feedback on the system's usability. Through this testing, the team was able to identify areas where the system could be improved, such as providing clearer instructions and more intuitive navigation.

Overall, the testing process enabled the project team to identify and fix several issues in the system before release. By performing Validation and Verification and utilizing Github for planning and organization, the project team was able to ensure that the system met both the functional requirements and standards set by the PO.

Technical Details

1. Spilt app.py and extracted queries.py.

The app.py file has been split into individual route.py files for each team member, using the Blueprint feature. This allows for independent work on route files, reducing conflicts when merging code to the main branch on GitHub. Additionally, we have adopted an object-oriented approach in our queries.py file where SQL commands are stored based on their relevance, such as UsersQueries.py for user-related queries and Subscriptions.py for subscription queries. This structure enhances code readability in the route.py file and facilitates code management through query reusability.

2. Front-end form validations.

To ensure proper data validation, we have implemented front-end form validations for various functions such as user registration, membership addition, and trainer profile update. These validations require mandatory input of specific data types, such as only letters being allowed for names or expiration dates being greater than the current date. In the case of trainer profile updates, all requirements must be met.

3. Calendar and sending messages

The scheduling system partitions the gym's operating hours, which range from 8:00 to 18:00, into ten discrete time zones, each representing an hour of the day. These time zones are implemented as an array of 0-9, where each index is indented proportionally. When a class is scheduled for a given time slot, a bar is generated on the calendar that displays the class name. The position of the bar on the calendar is determined by the proportional indentation of the time slot, while its length represents the duration of the class. For instance, if a class is scheduled to run from 8:00 am to 9:00 am, the bar will be located at the 0th index of the array, which corresponds to the beginning of the day, with a length proportional to the class duration. Additionally, members can click on the bar to book a class, and a pop-up message will inform them whether the booking was successful. The system also ensures that a member cannot book the same class twice.

To facilitate communication among members, the system incorporates the Message Box jQuery library, which has been customized to meet the gym's requirements. The library allows members to create and send messages, which consist of a title, a dropdown list, and content. The system enforces validation rules for each message, including mandatory entry of a title and content, as well as the selection of at least one member as the recipient. By implementing the Message Box library, the front-end workload has been significantly reduced, enabling the team to concentrate on improving the system's functionality.

4. Daily job to process payment automatically

We have created a daily job in PythonAnywhere to process payments automatically. The Tasks feature in PythonAnywhere enables us to schedule the dailyJob.py file to run our code automatically at a user-defined time each day in the US time zone. This job processes subscription and payment histories for active members who have not had a subscription for the past 24 days (31 days minus 7 days due time), streamlining the payment process.

Review of the Final Artefact

The Fitness Club Management system could be summarized into 8 different features with functionalities. Members Management, Trainer Management, Class & Session Management, Search Management, Booking Management, Payment Management, Notification Management and Reports.

For the Members Management, the system allows the members to update their information; to view their booking history and upcoming booked class; to book or cancel an exercise class or personal training session; to pay the subscription fee and personal trainer fee; to unsubscribe themselves from the system.

For the Trainer Management, the system allows trainers to add a new class/session; update personal information; view members in their class/session.

For the Class & Session management, the system allows trainer/manager to modify the class/session information; allow all system users to view Class/session information, such as class name, trainer name, and start/end time, remaining capacity.

For the Search Management, the system allows members and admin to search a class/session by using search bar.

For the Booking Management, the system allows members to book a class/session, the system has real-time booking confirmation function, after a click to “book”, the system will send the member information to the trainer. a booking can also be cancelled if the class has not started yet.

For the Payment Management, the system will allow members to book a personal training session by paying the training fee, and to keep the account active, members could pay the monthly subscription fee. We also provide daily task to check if there are member's subscriptions are due in next seven days and renew the subscription and process payment automatically by using tool called Tasks from Python Anywhere itself.

For the Notification Management, the system will send notifications to members such as subscription expiration, it also allows admin/manager to send a message to users. Members can only view messages.

For the last part of the system features – Reports. The system provides two reports, one is a popular exercise class list, which ranks the exercise class by its amount of member booking, second report is the fitness club finance report which show profit, expenses, balance of club's finance situation within select period.

Team Performance

The team demonstrated good communication, punctuality, and organisation skills, and excelled in setting up environments, establishing repositories, and ensuring all team members were familiar with the DevOps procedures. Below are some of the things that went really well during the sprints.

- Enablement work: The team completed necessary preparatory work before Sprint 1, such as setting up environments and getting everyone up to speed with devops procedures, which helped streamline the development process.
- DevOps Procedure Documentation: The team created detailed documentation of DevOps procedures, which helped ensure consistency and avoid errors during development and deployment.
- Teamwork and Communication: The team communicated effectively, collaborated well, and demonstrated good teamwork, which helped prevent misunderstandings, improve productivity, and foster a positive working environment.
- Good Scrum and Agile practice: The team followed Scrum and Agile principles, which allowed them to deliver working software increments in each sprint, provided flexibility to adapt to changing requirements, and encouraged collaboration with the Product Owner and stakeholders.
- Code review and detailed testing: The team performed thorough code reviews and testing, which helped identify and fix bugs early on, improve code quality, and ensure the system was functional and user-friendly.
- Kanban board and well-defined stages: The team utilized a Kanban board and well-defined stages, which helped keep track of tasks, identify blockers, and ensure that the team was making progress towards the sprint goals.

Lessons Learned

The team experienced some challenges with duplicated work and workload imbalance. To enhance team performance in the future, the team should consider pre-communication to prevent duplication of work, allocate tasks more effectively to balance workload, develop more specific acceptance criteria, and break down larger stories into smaller ones to improve workflow.

The team learned that effective communication is crucial in avoiding duplicated work and ensuring everyone is on the same page. Clear and well-defined acceptance criteria are necessary to avoid misunderstandings and delays. Balancing workload and breaking down big stories into smaller ones can improve workflow and efficiency. Additionally, identifying dependencies early and ensuring thorough testing and code review can minimise the risk of delivering a flawed product. Below are the key lessons learnt.

- Communication is key: Effective communication is crucial to ensure that everyone is on the same page and working towards the same goal. This includes pre-communication to avoid duplicate work, clear and well-defined acceptance criteria, and better coordination to avoid overwriting each other's work.
- Balance workload: It is important to ensure that tasks are allocated in a balanced manner to avoid overloading some team members while leaving others with less work. This can be achieved by better task allocation and planning.
- Thorough testing and code review: The team should prioritize thorough testing and code review to identify and fix bugs early, which minimizes the risk of delivering a flawed product.
- Define and refine acceptance criteria: The team should ensure that the acceptance criteria are specific, well-defined, and constantly refined to avoid misunderstandings and delays.
- Split big stories into smaller stories: Breaking down big stories into smaller stories can help to improve workflow, increase efficiency, and identify potential issues earlier.

Github Repository

Below is the link to the Github Repository

https://github.com/LUMasterOfAppliedComputing/comp693-2023-project1-group6

PythonAnywhere Link

Below is the link to Lincoln Fitness website hosted via PythonAnywhere

http://comp639group6.pythonanywhere.com/

Test User Details

Below is a list of test users for each role. It should be noted that no passwords have been configured for these users in the context of the Proof of Concept (POC) to facilitate quicker testing process. For the definitive implementation, passwords will be created for each user.

Email Address	Role	Password
admin@lincolnfitness.ac.nz	Admin	None
Blair.Wilmot@lincolnuni.ac.nz Trina.Greenwood@sina.com	Trainer	None
Barbara.Turner@hotmail.com Kayla.Slater@gmail.com GretchenProctor@ke.com LawandaVelazquez@acr.org CarlaRandall@ss.com	Member	None