**Test Plan**

**Account Creation/Login Tests**

**Test:** Coach Account Creation

**Initial Conditions:** Coach must be authorized by treasurer/admin

**Test Inputs:** coach email, name, phone number, address, new password, and authorization code issued by treasurer

**Expected Result:** coach account created and entered into system

**Test:** Member Account Creation

**Initial Conditions:** N/A

**Test Inputs:** member email, name, phone number, address, new password

**Expected Result:** member account created and entered into system

**Test:** User Login

**Initial Conditions:** User must have an account in the system

**Test Inputs:** Valid user login credentials(email and password)

**Expected result:** User guided to the correct UI based on their account role

**Treasurer User Tests**

***Financial Information Menu Tests***

**Test:** New Income Statement Creation

**Initial Conditions:** N/A

**Test Inputs:** Mouse click on “Create Income Statement” button

**Expected Result:** User enters income statement editing menu, prompted to enter new revenues/expenses for the income statement

**Test:** Editing Income Statement

**Initial Conditions:** User must have existing income statement

**Test Inputs:** Mouse click on “Edit Income Statement” button

**Expected Result:** User enters income statement editing menu, prompted to enter new revenues/expenses for the income statement

**Test:** Saving Income Statement

**Initial Conditions:** User must be in income statement editing menu

**Test Inputs:** Mouse click on “Save Income Statement” button

**Expected Result:** income statement updated with newly added revenues/expenses

**Test:** Monthly Profit List Display

**Initial Conditions:** There must be financial information to display

**Test Inputs:** Mouse click on “View Monthly Profits” button

**Expected Result:** Display a page containing an organized list of each month’s profit and income statement

**Test:** Monthly Expenses List Display

**Initial Conditions:** There must be monthly expenses to display

**Test Inputs:** Mouse click on “View Monthly Expenses” button

**Expected Result:** Display a page containing a list of expenses organized by month and priority

***Member Logging and Sorting Menu Tests***

**Test:** Member List Display

**Initial Conditions:** There must be member accounts to display

**Test Inputs:** Mouse click on “View Member List” button

**Expected Result:** Display a page containing a list of members and their information (name, phone number, address, classes they have attended, payments due) as well as a button to apply a discount

**Test:** Sorting Members by Attendance Frequency

**Initial Conditions:** User must be in member list menu

**Test Inputs:** Command to sort members by attendance frequency

**Expected Result:** Members are listed from highest attendance frequency to lowest

**Test:** Sending Members Discounts

**Initial Conditions:** User must be in top 10 most frequent attenders or not missed payment for 3+ months

**Test Inputs:** Mouse click on “Apply Discount” button

**Expected Result:** 10% discount applied to member’s most recent class, member notified of discount, and financial information updated

**Test:** Sorting Members by Payments Due

**Initial Conditions:** User must be in member list menu

**Test Inputs:** Command to sort members by payments due

**Expected Result:** Members are listed from most payments due to least

**Test:** Sending Overdue Payment Fees

**Initial Conditions:** User must have more than 1 payment due

**Test Inputs:** Mouse click on “Apply Penalty” button

**Expected Result:** Penalty fee issued to member, member notified, member payments due updated

***Coach Logging Menu Tests***

**Test:** Coach List Display

**Initial Conditions:** There must be coach accounts to display

**Test Inputs:** Mouse click on “View Coach List” button

**Expected Result:** Display a page containing a list of coaches and their information (name, phone number, address, classes they have coached, payments owed, schedule)

**Test:** Editing Coach Schedule

**Initial Conditions:** N/A

**Test Inputs:** Mouse click on “Edit Schedule” button for a certain coach

**Expected Result:** user prompted to add/remove classes to the coach’s schedule, coach and affected members notified of new or removed classes

**Coach User Tests**

***Practice Scheduling Tests***

**Test**: Sending Practice Reminders

**Initial Conditions**: Practice dates and member contact information are correctly entered in the system.

**Test Inputs:** Command to send out practice reminders.

**Expected Result**: All members receive reminders about the upcoming practice. The system logs the action and potentially tracks member responses or confirmations.

**Test:** Updating Practice Schedule

**Initial Conditions:** Existing practice schedule is available in the system.

**Test Inputs:** Modifications to the practice schedule (e.g., changes in time, date, location, or members).

**Expected Result:** The updated schedule is correctly reflected in the system. Notifications about the change are sent to all affected members.

**Test:** Communicating Schedule Changes

**Initial Conditions:** There's a need to cancel or reschedule a practice.

**Test Inputs:** Details of the cancellation or new schedule.

**Expected Result:** All members and treasurer are informed about the cancellation or the new schedule promptly. The system logs this communication.

**Test:** Recording Practice Attendance

**Initial Conditions:** Coach must have access to the coaching dashboard.

**Test Inputs:** Mouse click on “Record Attendance” button.

**Expected Result:** Coach enters attendance recording interface, prompted to select date and mark attendance for members present.

**Test:** Editing Practice Attendance

**Initial Conditions:** user must have access to the coaching dashboard and have existing recorded attendances.

**Test Inputs:** Mouse click on “Edit Attendance” button next to a specific practice date.

**Expected Result:** Coach enters attendance editing menu for that date, able to modify attendance records.

***Payment Reminder/Tracking Tests***

**Test:** Sending Payment Reminders

**Initial Conditions:** List of members with outstanding payments is up-to-date.

**Test Inputs:** Command to send out payment reminders to members with dues.

**Expected Result:** Members with pending payments receive reminders, detailing their outstanding amount and instructions for payment. The system tracks these notifications.

**Member User Tests**

***Member Attendance Tests***

**Test:** Display Schedule

**Initial Conditions:** N/A

**Test Inputs:** User clicks “View Class Schedule” button

**Expected Result:** Display a page containing a list of upcoming classes and their information (time, date, location, coach)

**Test:** Class Signup

**Initial Conditions:** User must be in schedule menu

**Test Inputs:** User clicks “Attend Class” button for a certain class in the schedule menu

**Expected Results:** User added to list of members for that class, given option to pay now through application, coach and treasurer notified accordingly

**Test:** Functionality of Attendance Logging

**Initial Conditions:** Member attends a session

**Test Inputs**: Member's ID or name as they check-in

**Expected Result:** Member's attendance is logged with the date and session details as well as payment status.

***Member Information Tests***

**Test:** Display classes a user has signed up for

**Initial Conditions:** User has signed up for at least one class

**Test Inputs:** User clicks “View My Schedule” button

**Expected Result:** Display a page containing a list of upcoming classes the user has signed up for with the option to cancel

**Test**: Sending Reminders for Upcoming Practices

**Initial Conditions**: Practice session scheduled

**Test Inputs**: Command to send reminders a specified number of days before the practice

**Expected Result**: All members receive reminders for the upcoming practice.

***Member Payment Tests***

**Test:** Display Payments Due

**Initial Conditions:** User has at least one payment due

**Test Inputs:** User clicks “View My Payments” button

**Expected Result:** Display a page with a list of the user’s outstanding payments as well as the associated class and the option to pay

**Test:** Functionality of Payment Status Update

**Initial Conditions:** Member makes a payment

**Test Inputs:** Member's ID or name and payment amount

**Expected Result**: Member's payment status is updated to reflect the new payment.