

Iteration 3: Toronto Premium Sports Club (TPSC) Archive

Danri He 500765982
Dennis Frolov 500776941
Elena Zubko 500768733
Diana Cazan 500680249
Christian Thomas 500761706

Ryerson University
CPS 406
Dr. Vojislav B.Mišić
Friday, April 6, 2018

Toronto Premium Sports Club (TPSC) Product Backlog

Requirements	Story	Initial Priority	Task	Estimate
Track Club Finances	Treasurer needs to be able to see the club's current profit	Med	Create the income statement and add the revenue portion (members payments, any other income that may come).	2
			Add the expense portion to the income statement (coach's payments, hall expenses, any other expenses).	2
			Create a list for the year, logging each month's profit to compare changes in revenue/expenses in order to find ways to maximize profit.	3
	Treasurer needs the current month's account payables (e.g. members who have paid in advance)	Med	For efficiency, this can be listed in the income statement under revenues. Since there is no mention of a return policy, this can be counted as a revenue as soon as the transaction is completed.	3
Log and Sort Members	Keep a log of the members	Med	Create a list of the members who have attended the class, and those who will be attending the class. Each member will have their name, phone number, whether they paid or not, and address.	1

			Create a members objects to keep track of the list of members and their following information	2
	Sort the members by whether they have paid or not. Provide a discount charge based on amount of unpaid/paid classes.	High	Using the list of members, organize the members by the amount of times they paid/not paid.	2
			If the member does not skip a payment for 3 months, he will receive a complimentary discount for 10% off for one class.	1
	Sort the members by their frequency of attendance in the club. Provide a discount based on amount of practices attended.	Low	Using the list of members, organize the members by the amount of times they have attended practice at the club.	2
			Based on the frequency level, the member will have a loyalty discount. The top 10 people on the list of most practices attended will receive a complimentary discount of 10% off for one class. This will help ensure member retention.	3
User Interaction with Application	The coach should be able to communicate to the members about future practices	High	Create a user interface where a message can be sent out to all or some members, with details of future practices or changes to any practices. A reminder message can also be sent out for a member who has skipped a payment.	4
	The treasurer should be able to manage coach list and schedule. The coach should be able to manage the members list (i.e. adding or	Med	Create user interfaces for different character to modify the list of participants. And the corresponding character will receive notification upon	3

	removing)		changes.	
	Members need to be able to view schedule through calendar.	Med	Create calendar interface that members can click and view schedule practices.	1

Throughout the following iteration we have applied the ‘Code- like- hell’ model. This allowed us to get out all of our ideas quickly during a restricted time period. However, the cons of this model were visible towards the end of the iteration as it was difficult to combine the requirements together from iteration two because we did not think of future problems. This limited us to some of the high priority tasks due to lack of foreseeing how the program should come together. However, we were able to finish majority of the tasks towards the end as we changed our approach on the project to the ‘Scrum’ model. We should’ve converted to this model at a sooner time to prevent the problems that occurred.



NOTE: The following chart represents our actual effort versus the ideal effort. We have completed all tasks for the following iteration over the span of 24 days.

Test Report

1. Introduction

In this document, we give the plan and results for testing of some of the key modules of the Toronto Premium Sports Club application.

2. Unit Testing Methodology

2.1 Selection of units: The most important object types for the test cases were selected. In the test environment, we developed formal test cases where we demonstrate the testing methodology by discussing only three object types.

2.2 Test Scripts: Each test case corresponds to a method or action in the following java program.

2.3 Fixing of Defects: The programmers fixed the defects found. The test cases were successfully complete when the program was executed.

2.4 Test Script Enhancement: As testing proceeded, some new test cases were added to fit the case of every action/ method in the program.

3. The Testing Tool

We used test cases to check the actual results of methods and actions:

- a) Buttons clicked
- b) Methods checked with entered parameters
- c) Alerts

4. Test Performed

We tested the methods and actions of following three object types: (A) Members, (B) Coaches, and (C) Treasurers

(A) Testing actions/ methods for the Members object type.

Operation Performed	Condition Tested	Actual Result
Member views notification	The GUI will display any messages the coach has sent to the specific recipient or to all members.	Test Passed.
Calendar button is clicked.	Once button is clicked the	Test Passed.

	GUI should display a calendar.	
Date on calendar is clicked.	A schedule of classes on the following day should display.	Test Passed.
Calendar button is clicked again.	Calendar should disappear and GUI should display original homepage.	Test Passed.
Member selected payment option.	Program should redirect to panel where the members may pay for their enrolled classes.	Test Passed.
Member makes payments by inputting required fields.	Payments go through.	Test Passed.
User selects Logout button.	The application should return to login homepage.	Test Passed.

(B) Testing actions/methods for the Coaches object type.

Operations Performed	Condition Tested	Actual Result
Message button is clicked.	A message area should appear on the screen allowing coaches to enter recipient, subject, and message.	Test Passed.
Coach sends messages, updates, alerts, and discount codes.	Pop-up should display notifying the user the message has been sent. Coach may send to member and treasurer.	Test Passed.
User views notification button.	GUI should display notification area where coaches can see any announcements, sender, and date.	Test Passed.
Members button is clicked.	GUI should display members.	Test Passed.

Option to sort by 'Paid'.	Members should be sorted based on if they have paid or not.	Test Passed.
Coach adds member.	GUI should redirect to panel where user can input all the information about the new member.	Test Passed.
Add member without inputting proper fields.	A pop-up should appear to notify user has invalid inputs.	Test Passed.
Add member with inputting proper fields.	GUI should automatically redirect to members panel where the new member is displayed on the list.	Test Passed.
Coach removes member.	A pop-up should appear notifying the corresponding member has been removed. The member should automatically be removed from the members list.	Test Passed.
Calendar button is clicked.	GUI should display calendar where coach can view the club's schedule.	Test Passed.
Coach selects Logout.	The application should return to login homepage.	Test Passed.

(C) Testing action/ methods for the Treasures object type.

Operations Performed	Condition Tested	Actual Result
Finance button is clicked.	GUI should display two tabs with income statement and history.	Test Passed.
Treasurer inputs parameters for expenses for selected month on the income page.	The parameters are written into an expense file.	Test Passed.
Treasurer inputs parameters for expenses for	The parameters are written into a revenue file.	Test Passed.

corresponding month.		
Treasurer selects corresponding month and clicks 'Display'.	The parameters of the expense and revenue file are read and generates an income statement using those values.	Test Passed.
History tab is selected.	GUI should display the entered month's revenue, expense, and total value of the year.	Test Passed.
Treasurer receives notifications.	A notification area should display on the page where announcements, sender, and date are displayed.	Test Passed.
Treasures adds coaches under the manage button.	A panel appears where treasures can manage coaches by adding or removing them.	Test Passed.
Schedule from calendar is viewed.	A schedule of classes on the following day should display.	Test Passed.
'Logout' button is selected.	The application should return to login homepage.	Test Passed.

5. Results

The test cases allowed us to obtain results by compiling and running the application for each case. The following results were given by the number of tests actually executed, if the expected results were successful, if the expected results were not successful, and the number of tests that did not go to completion.

For first test suite (testing in Members)

7 tests: success: 7, failure: 0, error: 0

For second test suite (testing in Coaches)

7 tests: success: 7, failure: 0, error: 0

For third test suite (testing in Treasurers)

9 tests: success: 9, failure: 0, error: 0

