

**Gym Management and Administration System**

This Report is submitted in partial fulfilment of the requirements for the

BSc (Honours) Information Systems and Information Technology (DT249) to the

School of Computing,

College of Sciences & Health, Dublin Institute of Technology.

**Author:** Tomas Mikoliunas – D14123810

**Supervisor:** Cathy Ennis, School of Computing

**Date:** 12 December, 2016

**Declaration of Own Work**

I, **Tomas Mikoliunas** hereby confirm that the project I now submit for Honours Project Module, titled **Gym Management and Administration System** is my own research work and was written by me following my research. The work is new and has not been submitted for any previous award. I further confirm that the work has not been taken from the work of others save and to the extent that such work has been identified, cited and acknowledged within the text of my proposal and presentation.

**Signed:**

**Date:**

# Acknowledgement

I would like to thank my supervisor, Cathy Ennis, for her help and guidance provided during the course of the project, who provided much needed support, encouragement and understanding through the long days require to complete my project.

Table of Contents

[Acknowledgement 2](#_Toc469315764)

[Abstract 4](#_Toc469315765)

[1. Introduction 5](#_Toc469315766)

[1.1. Background 6](#_Toc469315767)

[1.2. Project Objectives 6](#_Toc469315768)

[1.2.1. Limitation of existing systems 7](#_Toc469315769)

[1.3. Project Structure 7](#_Toc469315770)

[2. Technology choices and evaluation 8](#_Toc469315771)

[2.1. HTML for the web application 8](#_Toc469315772)

[2.1. CSS for defining the layout of HTML documents 9](#_Toc469315773)

[2.2. JavaScript, JQUERY for the dynamic web development 10](#_Toc469315774)

[2.3. AJAX for the asynchronous and synchronous calls without reloading the web application 10](#_Toc469315775)

[2.4. Spring MVC framework for the model-view-controller architecture development and dependency injection 11](#_Toc469315776)

[2.5. MySQL Relational Database Management System (RDBMS) 12](#_Toc469315777)

[2.6. JDBC Template, a database connection mechanism and for SQL queries execution 13](#_Toc469315778)

[2.7. Spring Security – Password Hashing or Password Encoding 14](#_Toc469315779)

[2.8. Apache Tomcat web application server 14](#_Toc469315780)

[3. Application implementation and design 15](#_Toc469315781)

[4. Testing 15](#_Toc469315782)

[4.1. Unit testing 15](#_Toc469315783)

[4.2. Unit integration testing 16](#_Toc469315784)

[4.3. Validation testing 17](#_Toc469315785)

[4.4. Performance testing 17](#_Toc469315786)

[4.5. User testing 18](#_Toc469315787)

[5. Conclusion 19](#_Toc469315788)

[6. References 19](#_Toc469315789)

# Abstract

This Project will look at Gym Management System which will be easy to use gym club management system. It is my intention to develop a secure which will consist of two modules: Main Web Application and a Database. Objective of my project is to provide an easy to use Gym Management System package. In terms of success criteria what would it contribute in business, it is my intension to provide the system. The web application will be password protected, so only authorized person could use this system.

The report comprises of an introduction which describes the project aim, objectives and scope followed by a technology overview, requirements detailing and finally the technical and design implementation.

# Introduction

As a gym owner, health club operator, or as a freelance personal trainer, without a doubt, one or more following questions crosses the mind daily…:

* How much membership plans did our gym sell yesterday/last week, etc.?
* Which one of our membership plans or products is selling the most?
* What is my gym retention rate?
* Who cancelled today and was the total losses?
* Etc…

An age old business adage states *“what gets measured gets improved”*. Therefore, we should measure the things that are important to your business [1].

As a gym owner who is running the business should be aware of when he/she was in danger of posting a bad week, month, or quarter way before the numbers were crunched by accountants. Having these statistics in advance is very good for a business. .

This Project will look at Gym management system which is an easy-to-use gym and health club membership management system. This software will help the gym administrator to keep records of members and their memberships, and have quick and easy communication between administrator and gym members. This Gym Management Software also includes user-friendly administrative interface, booking system, point-of-sale, accounting, which will help in management of the club, for example, payment and due collection, create, view, edit, delete member details or membership plans, member plan expiration and accounts status, photo capture supported, members schedule history and a range of reports that help in the management of the club.

I am attracted to this project for a number of reasons. The best gym management software should allow you to create, edit and easily find a member in the system. In the end, it should convert all tedious manual organising into automated, responsive cloud-based software. In addition to all the basic membership management features, my Club Management Software has dashboard that help to see how the organization operates over time, provides online sign ups, online booking system, inventory and point-of-sale system. Software’s strong focus on events and also the ability to see the success the organization long term is great for the business.

This is when I came up with an idea. This is still very new to me and my goal is to leverage from this college project in order to become a good analytic and developer.

## Background

My Gym Management Software is a complete gym and recreation facility system program which looks after all the members, memberships and activities. It is designed for gyms and health clubs.

My Gym Management Software provides lots of functions such data entry of club members, keeping records of all the things about club member’s fees, plan, bought or reserved products, physical fitness which help to provide good quality of services to club members from Gym administrators. Data to be stored in the database.

This system structure is become very simple to understand because of Data Flow Diagram provided by me. Context level Diagram and some chart are also available in the case study.

## Project Objectives

The Objective of my project is to provide the Software for helping gym owners, health club operators, freelance personal trainers, etc. better grow their businesses and promote their causes. Its goal is to make a complete membership solution that contains many enterprise features.

In terms of success criteria what would it contribute in business, it is my intension to provide system that would simplify the business of the club:

* Manage and maintain the member history easily
* Manage memberships management and packages
* Easy-to-use software which handles the customer-staff relationship in an effective manner
* Provide fast check-in process for members
* Online sign ups
* Online booking system
* Inventory and point-of-sale system
* E-mail campaigns
* Help to reduce the overload paperwork
* Minimize the human mistake and error, due to computerization

These will all be presented in a modern and intuitive interface.

The web application is a password protected, so only authorized person could use the system.

### Limitation of existing systems

Recall from one gym club owner: “*I have two gyms right now operating with a membership management software called Gym-master.. But I can only use it for a database reference for member retention rather than a real-time visualizer for all aspects of my business*” [1].

Also feedbacks followed with:

* The existing System such as Gym-master is not as much as user (Customer) friendly
* The communication with members is not well in Existing System because all the data is handled by Gym manager
* Members don’t get full accessibility to Gym centre and all permissions are allowed only for Gym manager
* The system cannot take effort out of finances and debt collection
* Systems are time taking software and cannot be easily install in Operation Systems

## Project Structure

This report is divided into a number of chapters, each chapter will detail a specific area of the project.

The early sections detail the technology that was used and why it was used, followed by a look at some of the alternative applications available today.

The report will then go into the specific details involved in the application design and construction with many relevant code examples. Finally, the report will show my conclusion and what I felt, I have learned over the course of this project.

# Technology choices and evaluation

For me to create an application for a Gym Management System which would be deployed on a web server and would be displayed on the web browser, a number of technologies were required, each involving their own specific learning challenges. While we did cover some of these technologies over the course of this program, the outcome was more foundational level knowledge. After day to day use and practice I would now present the following technologies were utilized to develop my intended project:

* HTML used as the format of the web application
* CSS the style language that defines layout of HTML documents
* JavaScript, JQUERY used in web development to manipulate the web page elements and to make it dynamic
* AJAX which is also a technique for creating fast and dynamic web pages, and which allows web pages to updated asynchronously by exchanging small amounts of data with the server behind the scenes, without reloading the web page
* Spring MVC framework which provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications
* Spring Security
* JDBC Template, which is a powerful mechanism to connect to the database and execute SQL queries
* MySQL, a database technology, Relational Database Management System (RDBMS)
* Apache Tomcat web server and servlet container that is used to serve Java applications

## 2.1. HTML for the web application

For the supporting Gym Management System web application, it was required that I further increase my knowledge of HTML coding, reading book “HTML and CSS: Design and Build Websites” [2], with an endless supply of resources available online: w3shcools.com, tutorialspoint.com, etc.

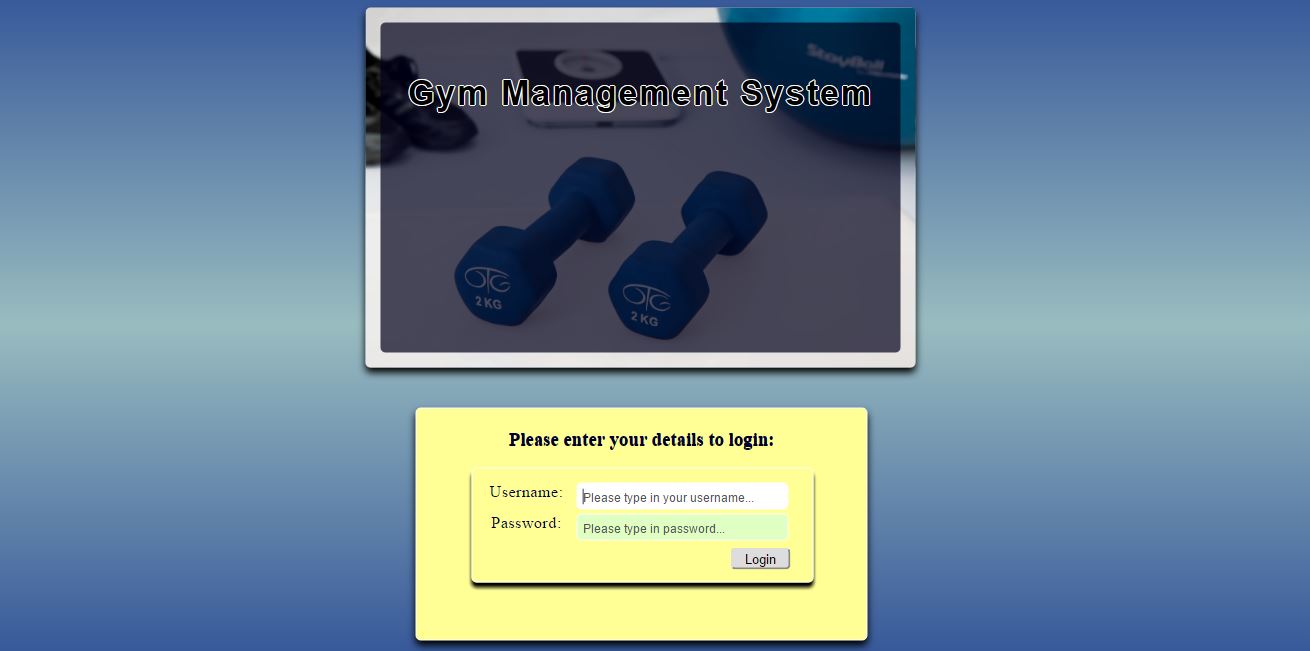


Figure . Login page

## CSS for defining the layout of HTML documents

I found that I was quickly able to build framework of my site, the more polished elements were achieved using cascading style sheets. Cascading Style Sheets or CSS allows for the styling aspects of a web-page to be controlled from a central file. By referencing a style sheet many web-pages can be styled from a central source. For my site I have written many files for typography, forms, buttons, navigation and other interface components.

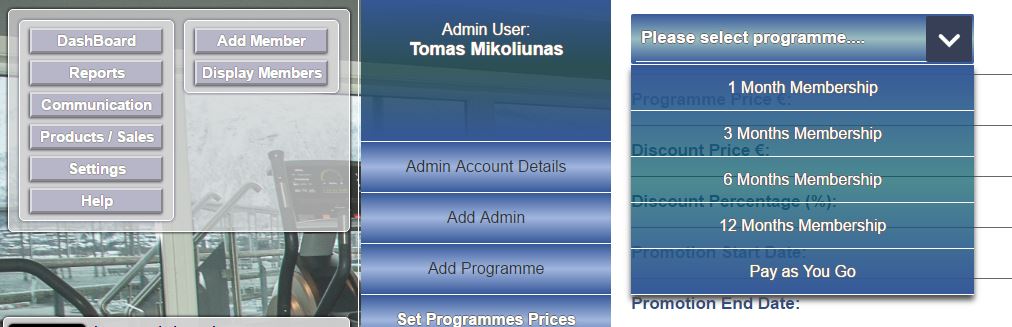


Figure . CSS style

## JavaScript, JQUERY for the dynamic web development

JavaScript and JQUERY were used to control web application and to run on the client side by the web browser. They support object-oriented programming and procedural programming. Thanks to my source books *“JavaScript step by step”* [3] and *“JavaScript and JQUERY: the missing manual”* [4]*,* where I have significantly deepened my knowledge about this programming language and its library. The first requirement for the web app was to define the content of the web page. This was done using HTML and CSS. Using just two languages like HTML and CSS you can create static web pages.

Incorporating JavaScript and JQUERY into a web application allows you to improve the user’s experience of the web application by converting it from a static page into one that can interact with them. With the usage of JQUERY I have added animation into the web app which either attract attention to a specific part of the page or which make the page easier to use. I have provided responses within the web application to various actions, like drop-down menus, sliding windows, etc.

## AJAX for the asynchronous and synchronous calls without reloading the web application

Using just two languages like HTML and CSS you can create static web pages that will be accessible regardless of which device is used to access the page. These static pages can interact with your visitor through the use of forms. Once a form is filled out and submitted a request is sent back to the server where a new static web page is constructed and eventually downloaded into the web browser.

The big disadvantage of web pages like this is that the only way that your visitor has of interacting with the page is by filling out the form and waiting for a new page to load.

After reading the book “Foundations of Ajax” I found that one of the great advantages, the AJAX technology does that by converting your static page into one that can interact with your application users without them needing to wait for a new page to load every time they make a request. Basically, AJAX is about updating parts of a web page, without reloading the whole page. I found it very useful for my one-page web application, where I have loaded different parts of the main page using AJAX calls to the server, which allowed me to update that asynchronously by exchanging data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

The example shows the ‘form’ submit with AJAX call to the server:

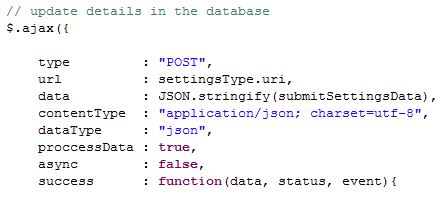


Figure . Submit the form

This technology was new to me, and I feel strongly enhanced my knowledge for the web development capabilities.

## Spring MVC framework for the model-view-controller architecture development and dependency injection

The framework like Spring MVC helped me to understand the way of building in Java the 3-tier web application. The MVC pattern results in separating the different aspects of the application (input logic, business logic and UI logic). MVC consists of Model, View and Controller. Model will be not only the data, but also the methods that will change this data.

I was new to Spring framework and since I had relevant knowledge of Java programming language and after my research on how your Java applications can benefit from this great framework and how it is in demand among the companies, I decided to learn this framework as it would help me to develop a great web application and also would be the advantage in my future job search.

During the research lots of readers suggested me *“Spring in Action”* [5]*,* as one of the best books to learn Spring. Seems like a worth reading book. I was looking for a common book to learn the both Spring MVC for web development and Spring core for using Spring framework as DI and IOC container, so *“Spring in Action”* is definitely a good choice.

Following is the list of few great benefits of using Spring framework:

* Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat [6]
* Testing an application written with Spring is simple because environment-dependent code is moved into this framework. Furthermore, by using JavaBean-style POJOs, it becomes easier to use dependency injection for injecting test data
* Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC into consistent, unchecked exceptions.

.Spring aims to make enterprise Java development easier and to promote loosely coupled code. Vital to this is dependency injection (DI) and aspect-oriented programming (AOP). DI is a way of associating application objects such that the objects don’t need to know where their dependencies come from or how they’re implemented. Rather than acquiring dependencies on their own, dependent objects are given the objects that they depend on. Because dependent objects often only know about their injected objects through interfaces, coupling is kept low. AOP enables you to centralize logic that would normally be scattered throughout an application in one place – an aspect. When Spring wires your beans together, these aspects can be woven in at runtime, effectively giving the beans new behaviour.

The Spring web MVC framework is designed around a DispatcherServlet that handles all the HTTP requests and responses. The request processing workflow of the Spring web MVC DispatcherServlet (see Figure 4).

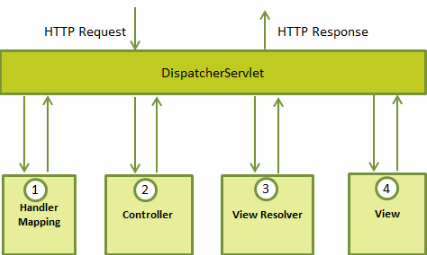


Figure . Workflow of the Spring web MVC.

## MySQL Relational Database Management System (RDBMS)

I chose MySQL as the database system as it is a popular, well documented and open-source system.

The MySQL database will store and manage all of member accounts, details etc.:



Figure . Tables in 'gym' database

“*In*[*computer science*](https://en.wikipedia.org/wiki/Computer_science)*, a database connection is the means by which a*[*database server*](https://en.wikipedia.org/wiki/Database_server)*and its*[*client*](https://en.wikipedia.org/wiki/Client_(computing))*software communicate with each other. The term is used whether or not the client and the server are on different machines*.” [7].

To establish and maintain communication with the database, the JDBC Template (Java Database Connectivity) of the Spring framework was used. This going to be discussed in the next section.

## JDBC Template, a database connection mechanism and for SQL queries execution

The JdbcTemplate class executes SQL queries, update statements and stored procedure calls, performs iteration over ResultSets and extraction of returned parameter values. It also catches JDBC exceptions. Instances of the JdbcTemplate class are thread-safe once configured. So you can configure a single instance of a JdbcTemplate and then safely inject this shared reference into multiple DAOs. A common practice when using the JdbcTemplate class is to configure a DataSource in your Spring configuration file, in my case **spring-servlet.xml** (see Figure 6) and then dependency-inject that shared DataSource bean into your DAO classes [8].

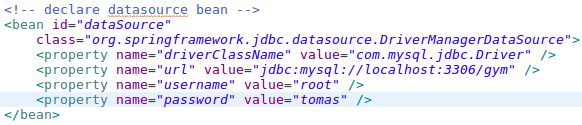


Figure . Data Source for MySql connection

We can perform all the database operation by the help of JDBC Template class such as insert, update, delete, and retrieval of the data from the database.

The example below (see Figure 7) shows simple SQL query to insert new Admin user into database:

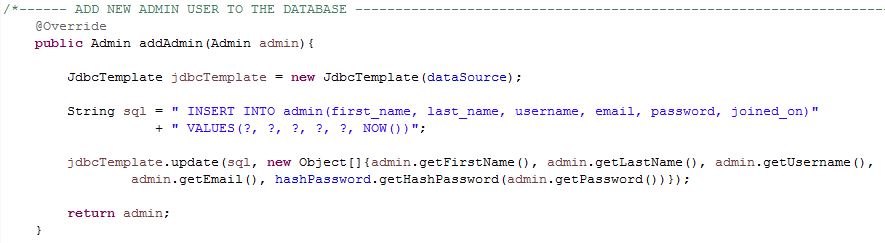


Figure . Insert new Admin user into database.

## Spring Security – Password Hashing or Password Encoding

This is very useful feature of Spring where I was able to integrate Spring Security with a Spring MVC framework. This feature gave ability to encode the login passwords for the admin and customers and store it into the database.

Saving your password in a simple string form in database is never a good idea, instead one must save his password in some sort of encoded or hashed format. To make password purely protected we must have some algorithm to identify how to save encoded password in database and how to authenticate user using an encoded password stored in database.

To encode a password using **'BCryptPasswordEncoder'** [9] spring-security provides ‘**encode()’** function of **'BCryptPasswordEncoder'** class (see Figure 8):



Figure . 'encode()' function for password encryption

Using the **‘BCryptPasswordEncoder’** and **‘encode()’** function, the password received has been encrypted.

And I test it that the password was encrypted with the **‘Logger’** class provided by Spring (see Figure 9):

C:\Users\Tomas\Documents\Koledzas\Honours_Degree_Project\IPR-2_Assignment_(5%)\encrypted_password.JPG

Figure . encrypted password

## Apache Tomcat web application server

Apache Tomcat [6] is a web server and servlet container that is used to serve Java applications. Tomcat is an open source implementation of the Java Servlet and Java Server Pages technologies, released by the Apache Software Foundation [10].

# Application implementation and design

In this section, I will detail the design of the application, this will cover the technical implementation as well as the visual components.

# Testing

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is a set of activities that can be planned in advance and conducted systematically.

For this reason a template for software testing a set of steps into which we can place specific test case design techniques and testing methods should be defined for the software process.

The testing process involved a number of different techniques and tools, in this chapter I will cover the main ones used, followed by the details of the actual user testing that was performed.

Testing is carried in two phases. First phase is carried during the software engineering, development, during the module creation. Second phase is after the completion of software. This is system testing which verifies that the whole set of programs working together.

In the first phase testing process should receive some inputs, processes them, and provides some outputs. It includes several activities, which should be carried out in different stages of software development. With the testing I was focusing on the possible unexpected outputs. This way I could correct the code as I went.

## Unit testing

Unit testing was performed to ensure smaller components of the application performed according to specification, this would then be followed by a level of integration testing to ensure the individual components worked together as expected.

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

## Unit integration testing

This testing is performed on the program’s structure when new modules being added while at the same time constructing test to uncover errors associated with interfacing the modules.

* Interfaces for the modules should be tested in the way that these interfaces perform their functions and make sure that they do what they are expected to do and also they do not do anything else which would lead to breaching of object-oriented rules.

Spring is a great framework to develop enterprise Java web applications. It provides a lot of features and one of them is the **TestContext** Framework, which helps us to implement integration unit tests easily in our applications [5].

I have created a model class with name **Member**. It has the following attributes, **firstName, lastName, address, email, dateOfBirth, email, password**, and accessor methods for them.

I also created **Service** and **DAO** classes as follows, in order to perform persistence operations with my domain model.

**MemberDAO** is an interface which defines persistence operation over **Member** instances like **addMember, memberProfile, updateMember, deleteMember**.

**MemberDAOImplementation** is an implementation of **MemberDAO** interface which employs **NamedParameterJbdcTemplate** bean of Spring in order to implement persistence operations via JDBC API [6].

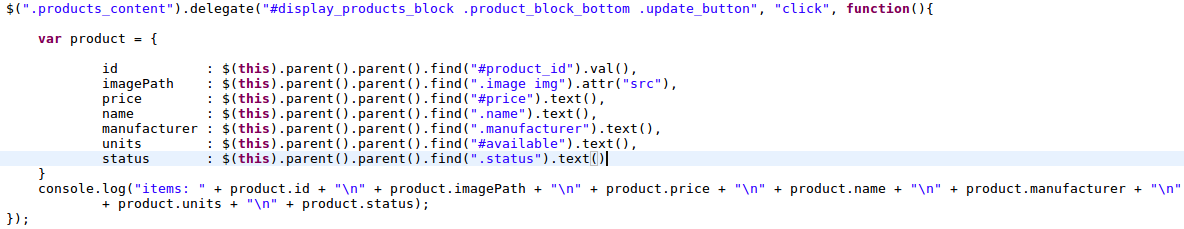
Then I wrote a transactional integration unit test where I have included test methods to test adding new Member instance, deletion of an existing one and finding by its id.

**@RunWith** annotation belongs to Junit, and is used to tell my STS IDE which Runner class, SpringJUnit4ClassRunner.class in this case, to use to run test methods defined in the class.

Therefore, SpringJUnit4ClassRunner starts a new transaction at the beginning of each test method execution, and then rolls back it at the end. The reason to rollback instead of commit is that those changes performed on the database within each test method should not adversely affect execution of other integration tests. However, any service method call which expects an active transaction to work during its execution is satisfied with that active transaction spanning the test method. It is possible to see how transaction is created and then rolled back from the log messages shown below.

Finally I have run the tests with JUnit and observed the results. The Junit tests passed as follows.

## Validation testing





## Performance testing

Performance testing is designed to test the run-time performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted. However, it is not until all system elements are fully integrated that the true performance of a system can be ascertained.

* Data collection technique: stop watch. For this it’s important to have a clear definition of when a task starts and stops
* Time on task - Time taken to complete a task
* Number of tasks completed in a given time
* Time spent recovering from errors. Number of errors. Number of subsequent moves to rectify error
* Number of clicks required for task

## User testing

After I have developed my application, it got to a point where I felt the application was ready, I felt confident I had tested all scenarios and had fail safes in place to handle all situations.

While I was indeed confident with my application, a level of actual user testing would be required. I felt I knew how the application would and should work, however it was unlikely I had tested all scenarios, even if the application was bug free, it might have still been rejected by users due to other reasons.

To get know my application rating evaluated, along with finding outstanding bugs, I enlisted the help of some family and friends. I selected 7 people, each were given a link to the my application which is hosted in the red hat cloud provider named as ‘Openshift’ at: <http://gym-tomas111.rhcloud.com>. Each person was asked to complete a set tasks provided by me, evaluate different features that I have, just to talk which of them most successful comparing with the time that I put for work implementing them and which of them the users liked the most, so I can put their feedback in the document and measure them, i.e. *‘I found this really easy to use, or I found the layout very good, I found it very functional*’.

* How did you find the registration/add new member process?
* How easy was to accomplish basic tasks the first time you encounter the design?
* Did you find the user interface easy to understand?
* How would you rate the effectiveness of the application?
* Would you recommend this application to others?

Each participant was also asked to make a note of anything they didn’t like or found difficult to understand.

# Conclusion

* Learn the Apache Tomcat web server working functionalities, application deployment and managing the lifecycle of the application within the server
* Learn the Spring and Spring MVC framework, which is the most popular application development framework for enterprise Java
* Learn the JDBC Template class, which simplifies the use of JDBC and helps to avoid common errors.
* Deepen my knowledge and understanding of Java programming language, also JavaScript, JQUERY, AJAX technologies, HTML, CSS

# References

1. GYMMASTER [Online]. Available from: <https://www.gymmastersoftware.com/> [Accessed on: 04 November 2015].
2. CAPTERRA TECHNOLOGY BLOG [Online]. Available from: <http://blog.capterra.com/top-7-free-open-source-membership-management-software-products/> [Accessed on: 04 November 2015].
3. DUCKET, J. (2013) *HTML and CSS: Design and Build Websites*. 1st Ed. Published by Wiley, Inc.
4. SUEHRING, S. (2013) *JavaScript step by step*. Third Ed. O’Reilly Media, Inc.
5. MCFARLAND, D. S. (2012) *JavaScript and JQuery: the missing manual*. Second Ed. O’Reilly Media, Inc.
6. From Wikipedia. *Database Connection*. Available from: <https://en.wikipedia.org/wiki/Database_connection/> [Accessed: 03 November 2015].
7. From Spring.docs.io. *Integration testing*. Available from: <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/integration-testing.html> [Accessed: 05 September 2016].
8. From Java Code Geeks. *Unit integration tests*. Available from: <https://examples.javacodegeeks.com/enterprise-java/spring/write-transactional-unit-tests-spring/> [Accessed: 05 September 2016].

# References

|  |  |
| --- | --- |
| [1] | Capterra, “Open Source Membership Management Software Products,” 21 10 2016. [Online]. Available: http://blog.capterra.com/top-7-free-open-source-membership-management-software-products/. |
| [2] | D. JOHN, HTML and CSS: Design and Build Websites, 1st Ed. ed., Wiley, Inc., 2013. |
| [3] | S. S., JavaScritp step by step, Third ed., O'Reilly Media, Inc., 2913. |
| [4] | M. D., JavaScript and JQUERY: the missing manual, Second ed., O'Reilly Media, Inc., 2012. |
| [5] | W. CRAIG, Spring in Action, Third ed., Manning Publications Co., 2011. |
| [6] | A. T. 8, “Apache Tomcat 8,” [Online]. Available: https://tomcat.apache.org/tomcat-8.0-doc/introduction.html. [Accessed 01 06 2016]. |
| [7] | M. Tutorial, “MySql Tutorial,” [Online]. Available: http://www.mysqltutorial.org/. [Accessed 01 06 2016]. |
| [8] | A. R. SCHUTTA NATHANIEL, Foundations of Ajax, 1st. ed., Apress, 2006. |