|  |
| --- |
| Anglia Ruskin University |
| Group Design Project |
| Group 4 |

|  |
| --- |
|  |

Department: Computing & Technology

Course Pathway: BSc Computer Science

Module: EJ230003S – Group Design Project

SID:

0805203/1

0806857/1

0819563/1

0912913/1

**06 May 2010**

# Contents

[1. Contents 2](#_Toc260920727)

[2. Current System 5](#_Toc260920728)

[2.1. Interview 5](#_Toc260920729)

[2.2. Floor plan 6](#_Toc260920730)

[2.3. Scenario 7](#_Toc260920731)

[2.4. Rich Picture 8](#_Toc260920732)

[2.5. Use Case 9](#_Toc260920733)

[2.6. Current Data Flow Diagram 10](#_Toc260920734)

[2.6.1. Data Dictionary 10](#_Toc260920735)

[2.6.2. Data Flow Diagram 12](#_Toc260920736)

[3. Proposed System 14](#_Toc260920737)

[3.1. Version Control 15](#_Toc260920738)

[3.2. Gantt chart 15](#_Toc260920739)

[3.3. Critical Path 17](#_Toc260920740)

[4. Hardware and Networking Solutions 18](#_Toc260920741)

[4.1. Hardware Required 18](#_Toc260920742)

[4.2. Network Topology 19](#_Toc260920743)

[5. Entity Realtionship Diagrams 20](#_Toc260920744)

[5.1. Version 1.0 20](#_Toc260920745)

[5.1.1. Database Schema 20](#_Toc260920746)

[5.1.2. Initial Entity Relationship Model 21](#_Toc260920747)

[5.1.3. Chen Diagram 22](#_Toc260920748)

[5.2. Version 2.0 23](#_Toc260920749)

[5.2.1. Database Schema 23](#_Toc260920750)

[5.2.2. Initial Entity Relationship Model 24](#_Toc260920751)

[5.2.3. Chen Diagram 25](#_Toc260920752)

[5.3. Version 3.0 26](#_Toc260920753)

[5.3.1. Database Schema 26](#_Toc260920754)

[5.3.2. Initial Entity Relationship Model 27](#_Toc260920755)

[5.3.3. Chen Diagram 28](#_Toc260920756)

[6. Software Functions 29](#_Toc260920757)

[7. Website Functions 36](#_Toc260920758)

[8. Coding 38](#_Toc260920759)

[8.1. Example of database table handler class (Member.cs) 39](#_Toc260920760)

[8.2. Example of a form that interfaces table data (frm\_member.cs) 47](#_Toc260920761)

[8.3. Example of listing collection of entities (frm\_member\_list.cs) 58](#_Toc260920762)

[8.4. Example of mysql database connection utils (mySqlConn.cs) 64](#_Toc260920763)

[8.5. Example of custom message box (frm\_message\_box.cs) 72](#_Toc260920764)

[8.6. Example of custom utils class (utils.cs) 79](#_Toc260920765)

[9. Testing Methods 82](#_Toc260920766)

[9.1. Black Box Testing 82](#_Toc260920767)

[9.2. White Box Testing 83](#_Toc260920768)

[9.2.1. Login 83](#_Toc260920769)

[9.2.2. Main Menu 84](#_Toc260920770)

[9.2.3. Staff Functions 86](#_Toc260920771)

[9.2.4. Managements 91](#_Toc260920772)

[9.2.5. Login Option 98](#_Toc260920773)

[9.3. Usability Testing 99](#_Toc260920774)

[9.3.1. Visibility of System Status 99](#_Toc260920775)

[9.3.2. Match Between System and the Real World 101](#_Toc260920776)

[9.3.3. User Control and Freedom 102](#_Toc260920777)

[9.3.4. Consistency and Standards 103](#_Toc260920778)

[9.3.5. Help Users Recognize, Diagnose, and Recover From Errors 105](#_Toc260920779)

[9.3.6. Recognition Rather Than Recall 109](#_Toc260920780)

[9.3.7. Fexibility and Minimalist Design 111](#_Toc260920781)

[9.3.8. Aesthetic and Minimalist Design 112](#_Toc260920782)

[9.3.9. Help and Documentation 113](#_Toc260920783)

[9.3.10. Skills 115](#_Toc260920784)

[9.3.11. Pleasurable and Respectful Interaction with the User 116](#_Toc260920785)

[9.3.12. Privacy 117](#_Toc260920786)

[9.4. Website Testing 118](#_Toc260920787)

[9.4.1. Website Test Plan – Firefox 119](#_Toc260920788)

[9.4.2. Website Test Plan - Chrome 120](#_Toc260920789)

[9.4.3. Website Test Plan - Internet Explorer 121](#_Toc260920790)

[9.4.4. Website Test Log – Firefox 123](#_Toc260920791)

[9.4.5. Website Test Log – Chrome 124](#_Toc260920792)

[9.4.6. Website Test Log - Internet Explorer 125](#_Toc260920793)

[10. Overview 126](#_Toc260920794)

[11. Appendix 128](#_Toc260920795)

[11.1. Bibliography - References 128](#_Toc260920798)

[11.1.1. References from documentation 128](#_Toc260920799)

[11.1.2. References from code 129](#_Toc260920800)

[11.2. Licenses 132](#_Toc260920801)

[11.2.1. Free Stock Icons 132](#_Toc260920802)

[11.2.2. Women at Fitness Club Image 132](#_Toc260920803)

[11.3. Class Documentation 132](#_Toc260920804)

[11.4. Initial interface designs 132](#_Toc260920805)

[11.5. Course Logbook 132](#_Toc260920806)

# Current System

## Interview

**1. Do they use any software solution currently?**

No. But they keep the member details in a single access 2003 tale.

Everything else is booked manually.

**2. What kind of information handled/received by staff interactions during work?:**

* Induction booking in manual folder
* He also said member data input happens always after actual induction, because it initiates the card access process
* Class booking requests
* Trainer booking requests
* Trainer details
* Resource details (like minivan renting)

**3. What data do you record about members?**

- FirstName   
- LastName   
- SID   
- MembershipType (Student, Staff, Gym Member, SportsFederation, Gym + SportsFederation, Gym Alumni, Gym Shorterm, Gym SummerSchool, Community Monthly, Community PAYG)   
- Gender   
- Expire Date   
- Renewal (YES, NO)   
- Address   
- Email   
- Phone Number   
- Emergency Number   
- Nationality   
- Receipt Number   
- Money Taken (initials of the person who took the money)   
- Club (if applicable)

**4. What data do you record about trainers?**

-name  
-Date of birth  
-qualifications (what classes can they do)  
-inductioning (yes/no)  
-personal training (yes/no)

**5. What are the gym business hours (when is it open for member access)**

8am-10pm

**6. What are the shift and lunch times?**

- Shifts work from:   
- Early shift: 7:30 till 2:30/3:30   
- Late shift: 1pm till 8pm   
Lunch time: 1pm till 1:30pm

## Floor plan



**Gym Floor Plan 1.1**

## Scenario

Classes run up to 3 times a week

& Machines

10 Classes

Inductions

2 Managers

Customer Payments

200 Members

15 Fitness Instructors

2 Receptionists

Current System

Trees leisure centre & gym is independently run and has no current electronic system for recording time tables, memberships or payment. There is an access database simple containing membership contact details but everything else is recorded by hand and kept on paper. The staff books the inductions manually in an appointment book. There are 2 managers, 2 receptionists and 15 other members of staff, all the contact details and any other information is all stored on paper in the manager office. The gym has fitness machine which can be used by all of gym members, however a member can also book a personal trainer for 2 hour blocks throughout the day. The leisure centre also offers 10 other classes including yoga, spinning class and aerobics, a lot of the classes take place up to 3 times a week to meet the demand of the sym members. Each class can only have a certain number of members; each member can book themselves on a class however a certain number of spaces are always left empty to allow for walk-in’s. If a member wishes to attend any of the classes or book a personal trainer they either phone the reception desk to arrange times or physically visit the gym. At present the record of payment is all on paper and the only information recorded is the initials of receiving staff.

The trainers are employed specifically for the time constraints of the classes, therefore the system needs not to care about individual trainer availability as they sort out between each, if there is more than one trainer for a class.

Currently there are no access rights at all, anyone can start up the access database and modify any data or write anything into the manual records. We also propose at least three user access level: manager, staff and customer (for booking class over internet).

## Rich Picture

Database

Servers

Networking

Online system

Class Booking

Timetabling

Rotas

Back office systems

Log in/log out

Terminals

Implementation

Maintenance

Training

Machinery/equipment

Facilities

Classes available

Class users

Casual gym users

Regular gym users

Software

Existing Features

Market

Cost

Hardware

**Gym Management System**

## Use Case



## Current Data Flow Diagram

### Data Dictionary

**Data Flows**

|  |  |  |
| --- | --- | --- |
| **Ref** | Data | *Description* |
|  | address | = houseNumber + postCode |
|  | classBookingInfo | = { trainerName + className + roomNumber + date + time } |
|  | classBookingRequest | = trainerName + className + roomNumber + classStartDate + classStartTime + classLength |
|  | classStartDate | = date |
|  | date | = day + month + year |
|  | day | \*number between 1-31\* |
|  | gender | = [ Male | Female ] |
|  | name | = (nameTitle) + { [ forename | initial ] } + surname |
|  | nameTitle | = [“Mrs” | “Ms” | “Mr” ] |
|  | sessionType | = [ Induction | Personal Training ] |
|  | trainerBookingInfo | = { trainerName + sessionType + roomNumber + sessionStartDate + sessionStartTime + sessionLength } |
|  | trainerBookingRequest | = trainerName + sessionType + roomNumber + sessionStartDate + sessionStartTime + sessionLength |
|  | trainerName | = name |
|  | year | \*the current year\* |

**Data Stores**

|  |  |  |
| --- | --- | --- |
| **Ref** | Data | *Description* |
| **M1** | memberRecord | = { memberDetails } |
| **M2** | TrainerAppBook | = trainerBookingInfo |
| **M3** | ClassTimeTable | = classBookingInfo |
| **M4** | TrainerRecords | = { trainerDetails } |

**Forms Data Dictionary**

|  |  |  |
| --- | --- | --- |
| **Ref** | Data | *Description* |
| **Member Details Form** | | |
|  | expireDate | = date |
|  | memberDetails | = memberName + SID + membershipType + gender +   expireDate + renewal + address + email + phoneNumber +   emergencyNumber + nationality + receiptNumber +   moneyTaken + ( club ) |
|  | memberDetailsForm | = memberDetails |
|  | memberDetailsFormBlank | = memberDetailsForm  \*Blank form\* |
|  | memberDetailsFormFilled | = memberDetailsForm  \*Filled with valid data \* |
|  | membershipType | = [ Student | Staff | Gym Member | SportsFederation | Gym And SportsFederation | Gym Alumni | Gym Shortterm | Gym SummerSchool | Community Monthly | Community PAYG ] |
|  | moneyTaken | = { initials }  \* initials of the person who took the money\* |
|  | Renewal | = [ yes | no ] |
| **Trainer Details Form** | | |
|  | dateOfBirth | = date |
|  | inductioning | = [ yes | no ] |
|  | personalTraining | = [ yes | no ] |
|  | trainerDetails | = trainerName + dateOfBirth + qualifications + inductioning + personalTraining |
|  | trainerDetailsForm | = trainerDetails |
|  | trainerDetailsFormBlank | = trainerDetailsForm  \*Blank form\* |
|  | trainerDetailsFormFilled | = trainerDetailsForm  \*Filled with valid data \* |

### Data Flow Diagram

#### Level 0 Current Dataflow Diagram



#### Level 1 Current Dataflow Diagram

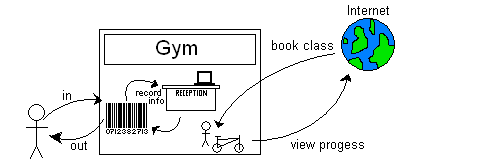


# Proposed System

The proposed system will be able to hold the member and staff details, which will be set and edited by the manager and viewed by the members of staff. The manager will also be able to add new classes to the system for example if the gym decided to start teaching a spin class then the information about the class would need to be added to the system so that members could be booked on that particular class. The program also needs to be able to book members on class induction and/or personal training session. There needs to be a facility that can record information about new members, and then record the member’s progress throughout their membership. For example it would include classes they have attended and target weight.

There needs to be access rights implemented within the gym to make sure that the system is secure and that all information is kept safe. Different security levels can be controlled by giving certain user names access to only certain areas of the system. For example fitness instructors can only view staff rotas, enrol members on a class, view class information, view member information and alter member information, where as the manager can do all of this and have the added privileges of editing staff information, staff rotas, class information and adding brand new classes to the system.

This will be implemented on to two computers situated at the front desk, for use by the receptionist and the fitness instructors. There will also be a computer in the manager’s office so that they can do work in private.



## Version Control

Complete project version control can be found here:

<http://code.google.com/p/anglia-gym/>

## Gantt chart

We have constructed two gantt charts, one of these gantt charts has been constructed One of these gantt charts was constructed in Microsoft project that particular gantt chart shows just the stages of the entire process from the start of the requirement analysis until the project has been completed and implemented. However it does to specify tasks it is just a top level plan.

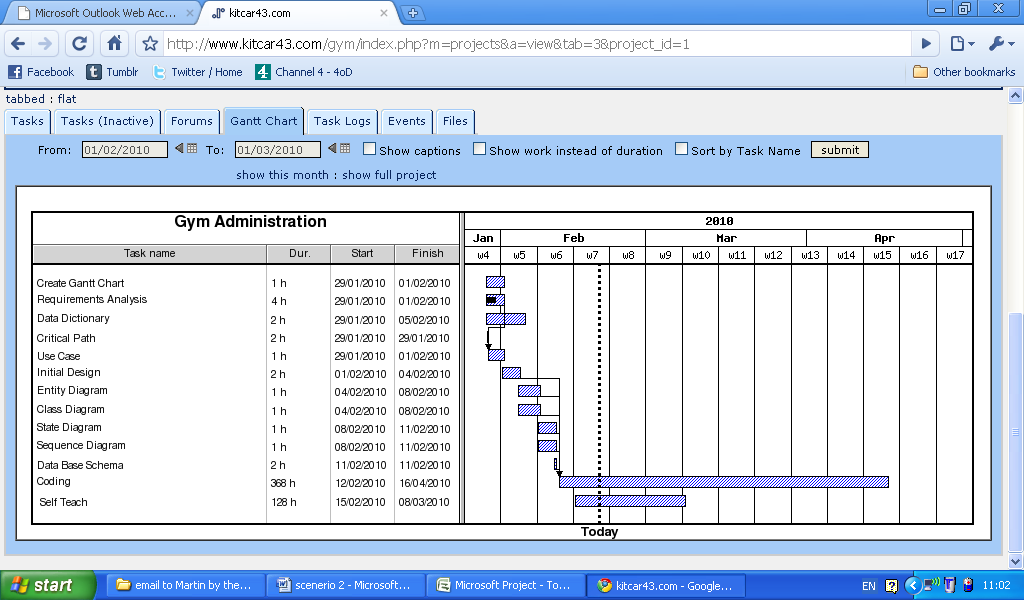
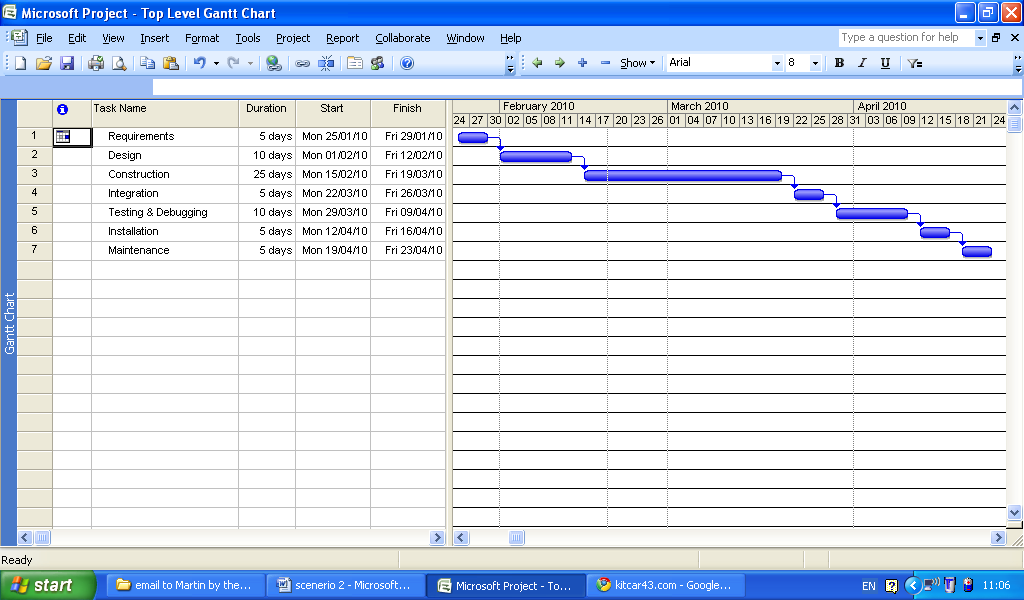
The other however was constructed using a piece of project management software called ‘dot project’, we have chosen this software because it is accessed online and can be accessed and edited by any member of the group. Each task is given a start and end date, it is also required that a member the group to be assigned to completing the task, more than one member can be assigned to a task. This means that if a new task is decided then any member can log on and assign themselves to that task. It keeps the plan up to date and allows any member of the group to check the time or place of a meeting. Dot Project also has the capability to upload documents and share them with each member of the group, a member can edit that document and upload a new version. This means that we can study any changes to make sure we don’t lose any important information.

**Microsoft Project**

Microsoft Project is a project management software program it is designed to assist project managers. The program allows you to track project progress, budgets, resources, developments and analyzing workloads.

The program also allows you to create critical path schedules from the information that is entered. There is also user options so that different classes of users can be defined to control different access levels.

Resources can easily be assigned to different tasks to make sure that deadlines are met & all users are aware of what needs to be done. Finances can easily be controlled which can improve cost estimates.

 ****

## Critical Path

15

5

0

5 Days

5 Days

10 Days

5 Days

25 Days

10 Days

5 Days

Design

8

Requirements

Construction

Integration

Testing &

Debugging

Installation

Maintenance

7

6

5

4

3

2

1

The critical path method or critical path analysis is a mathematically based algorithm for scheduling a set of project activities. It is an important tool for effective project management.

The technique for CPM includes the following:

1. A list of all activities required to complete the project
2. The time that each activity will take to complete.

|  |  |  |
| --- | --- | --- |
| Task | Length | Dependencies |
| Requirements | 5 Days |  |
| Design | 10 Days | Requirements |
| Construction | 25 Days | Design |
| Integration | 5 Days | Construction |
| Testing & Debugging | 10 Days | Construction |
| Installation | 5 Days | Testing & Debugging |
| Maintenance | 5 Days | Installation |

1. The dependencies between the activities.

Using these values CPM calculates the longest path of the planned tasks from start to finish. It also includes the start and finish times for each task what extending the length of the project. The process determines which tasks are critical and which tasks can be delayed.

The result of the critical path analysis allows managers to practice tasks to make sure that the completion of the task is time effective and target met.

Maintenance

Testing &

Debugging

Integration

Construction

Requirements

Design

Installation

1

2

3

47

57

6

7

8

# Hardware and Networking Solutions

As with any computer system, the existing hardware must be evaluated before any implementation of software or future hardware solutions. As the current system was so simple and paper based, the client really had no reason to have more than two computers. One for the upkeep of the database, and one for simple word processing, for posters and the creation of timetables etc.

With the introduction of the new system, with its desktop application and its web based application, the old hardware would be unable to keep up with the demands of the new software system. We have therefore devised the following solution for the client.

## Hardware Required

To enable the client to be entirely computer based, we have concluded that the following hardware options are to be sourced.

Server – Dell Poweredge T110 – An affordable and robust server, ideal for small businesses.

Desktop Computers – Dell Vostro Desktop Computer – perfect for a small business like this.(Dell, 2010)

Router – Cisco 800 Series – Ideal for small businesses. (Cisco, 2010a)

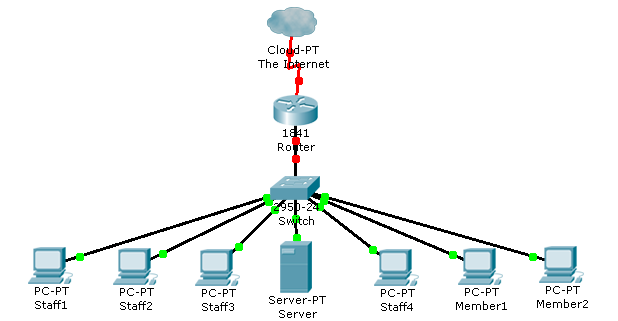
Switch – Cisco Catalyst Express 500 Series – once again, ideal for a small business.(Cisco, 2010b)

All of the above products have been chosen because they are ideally suited for a small business such as this. The Dell server is optimised for use with less than 25 other machines, and seeing this system would only require 4 (6 if the existing machines are to be replaced) currently, this server seems a better option than some of the higher costing ones. The Dell Vostro Desktop, like the server, is ideal for a small business. It is affordable, and the one chosen is the slim tower design, which means it will reduce the amount of space needed to store it.

The router and switch, the fundamental items behind collaborating the network for the client. Both of these have been selected from the broad range of products by Cisco, simply because of the manufacturer’s reputation within the networking community. (Cisco Systems, 2010) Whilst the comments are not corresponding to the chosen hardware in this scenario, they apply to the company as a whole. However the chosen products are perhaps more suited towards the small office than small gym, but we feel that they can cover the requirements substantially.

## Network Topology

One other fundamental thing within this system is how the computers, server, router and switch will be connected. Below is the proposed network topology for the client.



Topology created by Cisco Packet Tracer (Cisco, 2010c)

Member1 and Member2 are currently the old machines, but in the future these could be used by members to book their own classes and change their own details.

The IP addressing would consist of a single subnet, with an ideal network address of 192.168.1.0/24. This then could be adapted if any future expansion was considered, simply by adding more subnets.

# Entity Realtionship Diagrams

## Version 1.0

### Database Schema

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | STAFF | | **id\_staff** | | **id\_user \*** | | name | | address | | position | | NInumber | | contractType | | phoneNr |  |  | | --- | | EQUIPMENT\_BOOKINGS | | **id\_mbr\_eq\_booking** | | **id\_member \*** | | **id\_equipment \*** | | date\_start | | date\_due | | |  | | --- | | MEMBERS | | **id\_member** | | **id\_user \*** | | name | | address | | email | | type | | payment\_method | | phoneNr |  |  | | --- | | ROOMS | | **id\_room** | | name | | description | | size | | |  | | --- | | CLASSES | | **id\_class** | | name | | description | | type |  |  | | --- | | CLASS\_BOOKINGS | | **id\_class\_booking** | | **id\_class\_instance \*** | | **id\_member \*** | | booking\_date |  |  | | --- | | CLASS\_INSTANCE | | **id\_class\_instance** | | **id\_class \*** | | **id\_staff \*** | | start\_time | | end\_time | | frequency | |
| |  | | --- | | USERS | | **id\_user** | | login | | password | | profile | | |  | | --- | | PAYMENTS | | **id\_payment** | | **id\_member \*** | | date | | amount | | details | | |  | | --- | | EQUIPMENT | | **id\_equipment** | | name | | description | | id\_set | |

### Initial Entity Relationship Model



### Chen Diagram



## Version 2.0

### Database Schema

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | STAFF |  | | **id\_staff** |  | | **id\_user \*** |  | | firstName | 1 | | lastName | | birthdate |  | | address\_1 | 2 | | city | | county | | postalcode | | natinsnumber | 3 | | contract\_type | | position | | qualifications | | contract\_start | | contract\_finish | | address\_2 | 4 | | emerg\_contact\_name | | emerg\_contact\_telephone | | emerg\_contact\_relation | | nationality |  | | allergies |  | | medicalNotes |  | | |  |  | | --- | --- | | MEMBERS |  | | **id\_member** |  | | **id\_user \*** |  | | firstName | 1 | | lastName | | birthDate |  | | address\_1 | 2 | | city | | county | | postalCode | | type |  | | payment\_method |  | | is\_active |  | | address\_2 | 3 | | emerg\_contact\_name | | emerg\_contact\_relation | | emerg\_contact\_phone | | emerg\_contact\_mobile | | allergies |  | | medical\_notes |  | | picture |  | | |  | | --- | | CLASSES | | **id\_class** | | name | | description | | type |  |  | | --- | | CLASS\_BOOKINGS | | **id\_class\_booking** | | **id\_class\_instance \*** | | **id\_member \*** | | booking\_date |  |  | | --- | | CLASS\_INSTANCE | | **id\_class\_instance** | | **id\_class \*** | | **id\_staff \*** | | **id\_room \*** | | date | | start\_time | | end\_time | | frequency | |
| |  | | --- | | STAFF\_EQUIPMENT\_BOOKINGS | | **id\_staff\_eq\_booking** | | **id\_staff \*** | | **id\_equipment \*** | | date\_start | | date\_due | | |  | | --- | | PAYMENTS | | **id\_payment** | | **id\_member \*** | | date | | amount | | details | | |  | | --- | | EQUIPMENT | | **id\_equipment** | | name | | description | | id\_set | |
| |  | | --- | | MEMBERS\_EQUIPMENT\_BOOKINGS | | **id\_mbr\_eq\_booking** | | **id\_member \*** | | **id\_equipment \*** | | date\_start | | date\_due | | |  | | --- | | USERS | | **id\_user** | | login | | password | | profile | | active | | |  | | --- | | ROOMS | | **id\_room** | | name | | description | | size | |

### Initial Entity Relationship Model



### Chen Diagram



## Version 3.0

### Database Schema

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | STAFF |  | | **id\_staff** |  | | **id\_user \*** |  | | firstName | 1 | | lastName | | birthdate |  | | address\_1 | 2 | | city | | county | | postalcode | | phone !!! | | mobile !!! | | natinsnumber | 3 | | contract\_type | | position | | qualifications | | contract\_start | | contract\_finish | | address\_2 | 4 | | emerg\_contact\_name | | emerg\_contact\_telephone | | emerg\_contact\_relation | | nationality |  | | medical\_allergies | 5 | | medical\_notes | | |  |  | | --- | --- | | MEMBERS |  | | **id\_member** |  | | **id\_user \*** |  | | firstName | 1 | | lastName | | birthDate |  | | address\_1 | 2 | | city | | county | | postalCode | | email | | phone !!! | | mobile !!! | | type |  | | payment\_method |  | | is\_active |  | | address\_2 | 3 | | emerg\_contact\_name | | emerg\_contact\_relation | | emerg\_contact\_phone | | emerg\_contact\_mobile | | medical\_allergies | 4 | | medical\_notes | | medical\_doc\_name | | medical\_phone | | picture |  | | member\_number |  | | |  | | --- | | CLASSES | | **id\_class** | | name | | description | | type |  |  | | --- | | CLASS\_BOOKINGS | | **id\_class\_booking** | | **id\_class\_instance \*** | | **id\_member \*** | | booking\_date |  |  | | --- | | CLASS\_INSTANCE | | **id\_class\_instance** | | **id\_class \*** | | **id\_staff \*** | | **id\_room \*** | | date | | start\_time | | end\_time | | frequency | |
| |  | | --- | | PAYMENTS | | **id\_payment** | | **id\_member \*** | | date | | amount | | details | |  | |  | | --- | | EQUIPMENT | | **id\_equipment** | | name | | description | | id\_set | |
| |  | | --- | | EQUIPMENT\_BOOKINGS | | **id\_staff\_eq\_booking** | | **id\_equipment \*** | | **id\_staff \*** | | **id\_member \*** | | **id\_class\_instance \*** | | date\_start | | date\_due | | |  | | --- | | USERS | | **id\_user** | | login | | password | | profile | | active | | |  | | --- | | ROOMS | | **id\_room** | | name | | description | | size | |

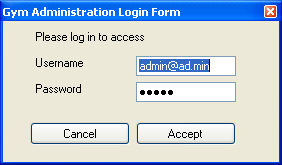
### Initial Entity Relationship Model



### Chen Diagram



# Software Functions



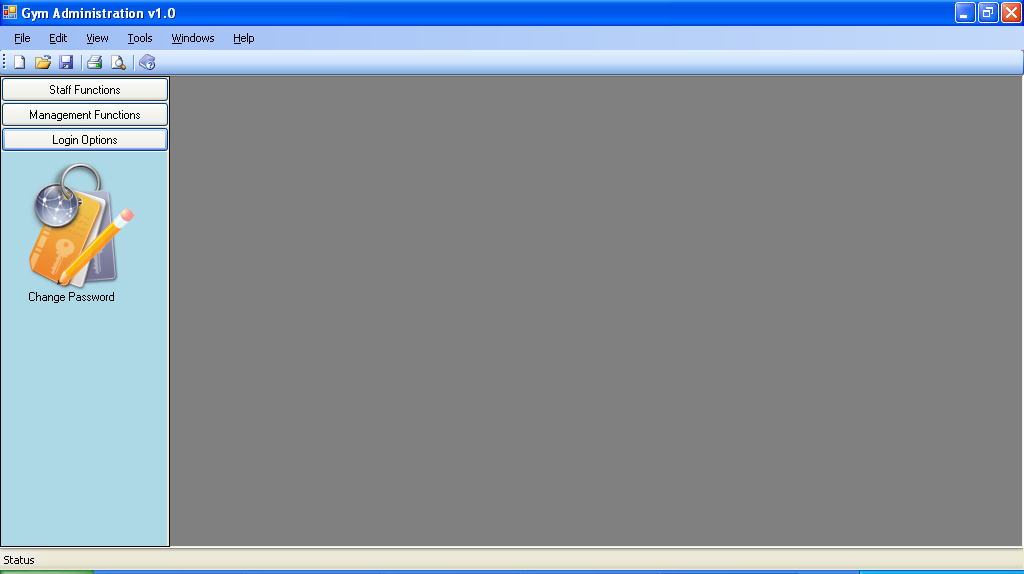
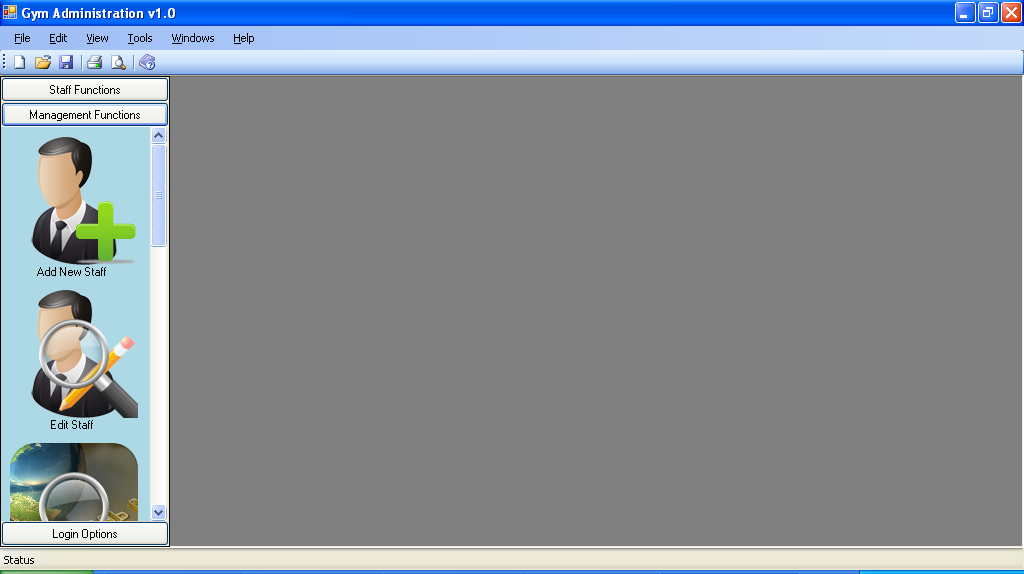
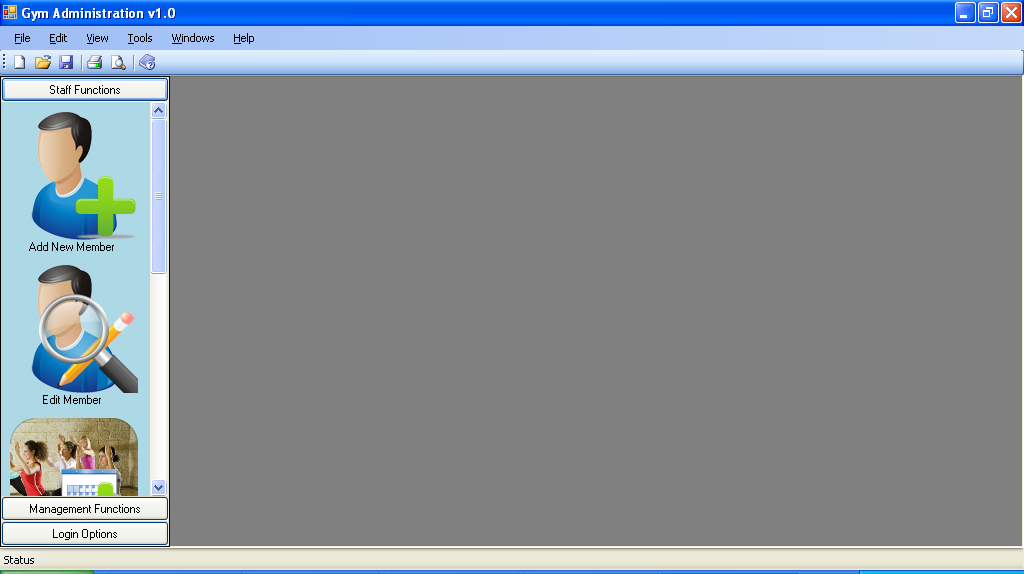
Staff Functions

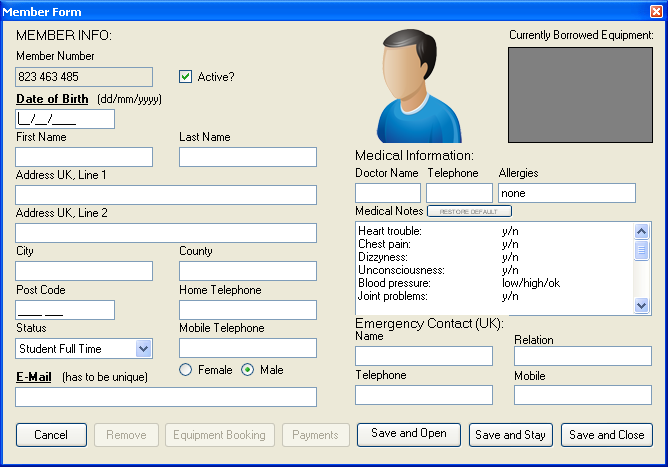
Login Options

Management Functions

Login Window

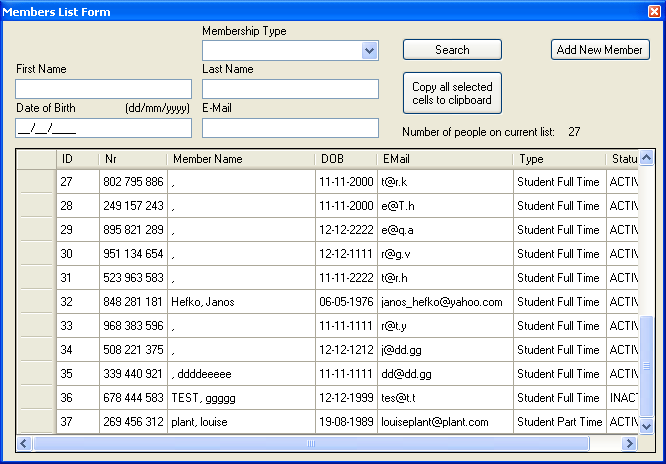
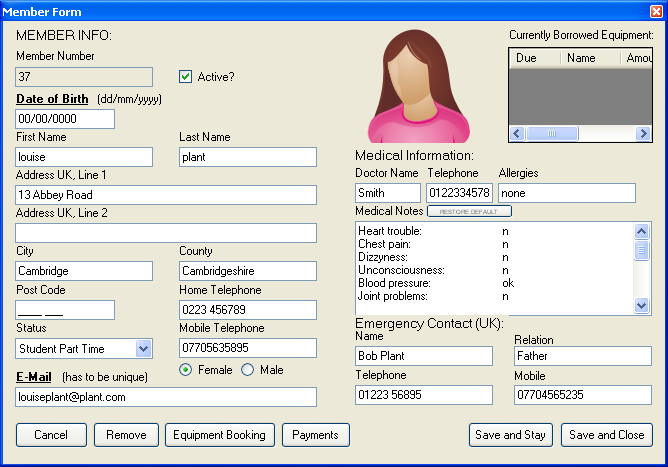
Menu





Staff Functions /Add New Member

Click Add New Member to open a blank Member Form



Staff Functions /Edit Member/View Member Information/Equipment Bookings

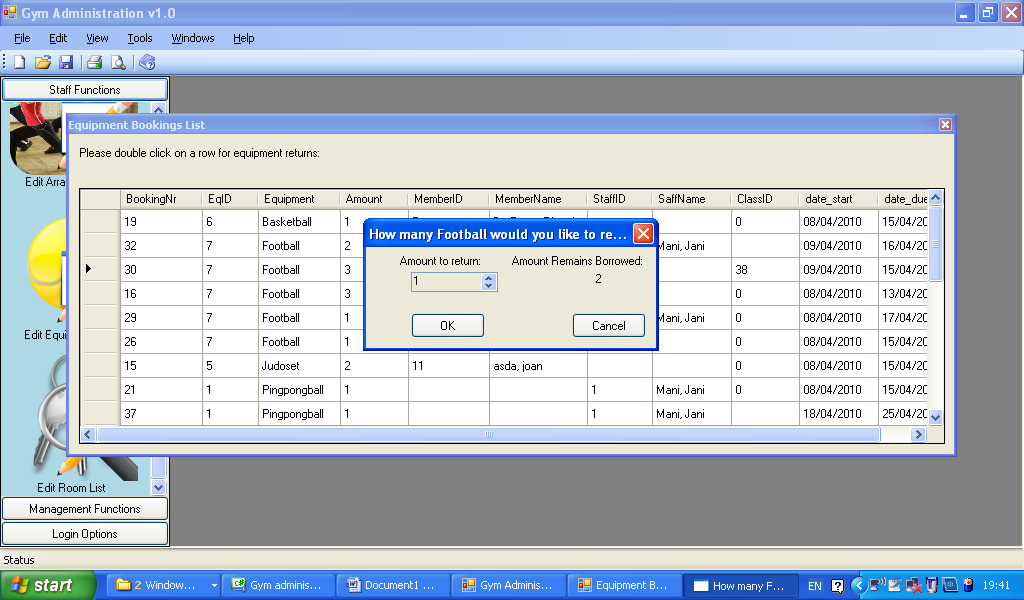
Save the information and then open the Member List

By filling in the fields and clicking search it makes it easier and quicker to find Members

Double click a record to view the Members Information

Staff Functions /Edit Member/Member List

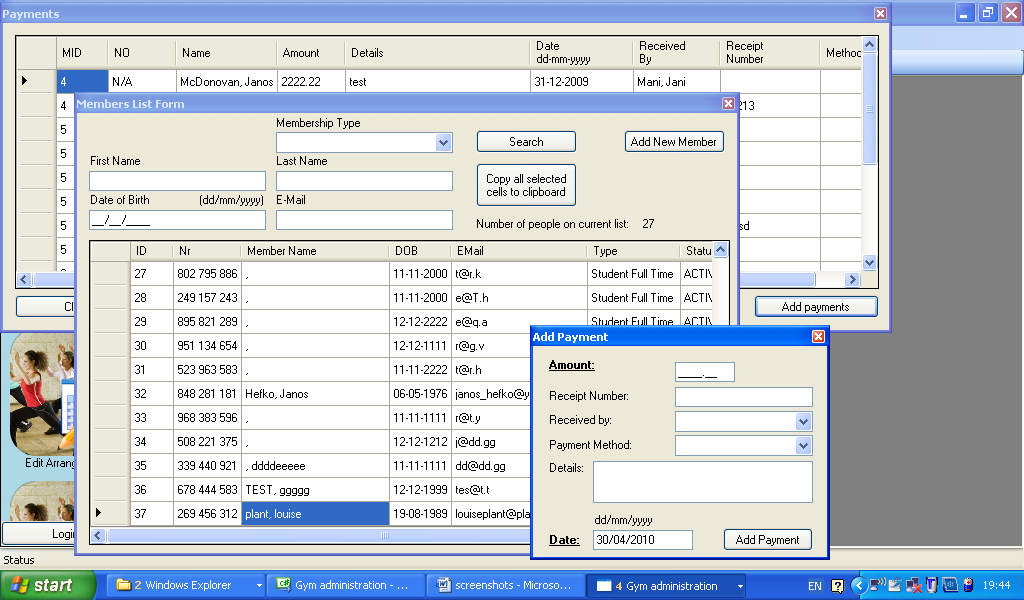
Equipment Bookings/Return Items



Click “add payments”

to select a member to add a payment

Payments List/Add Payments

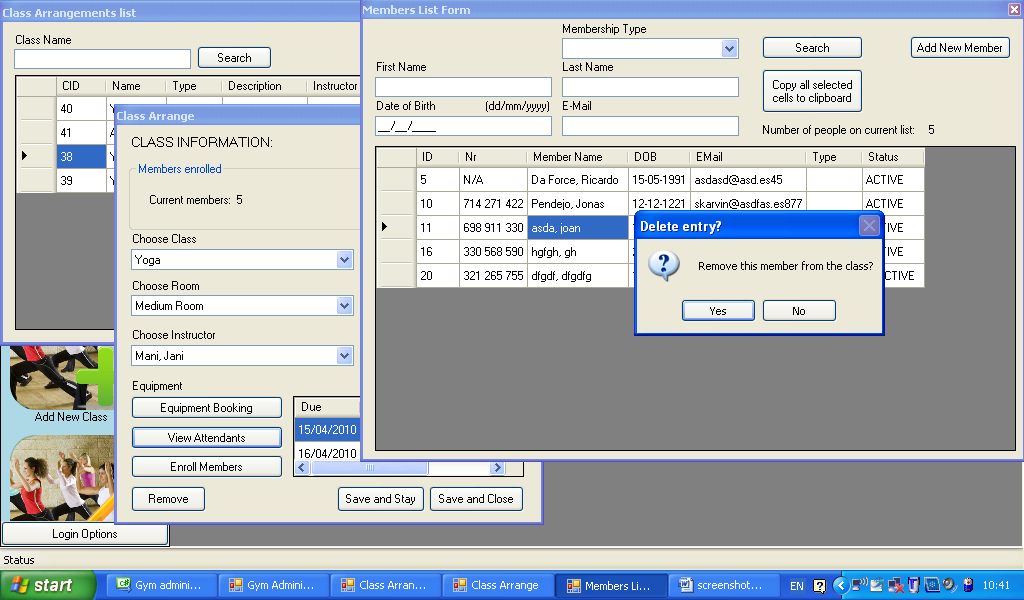


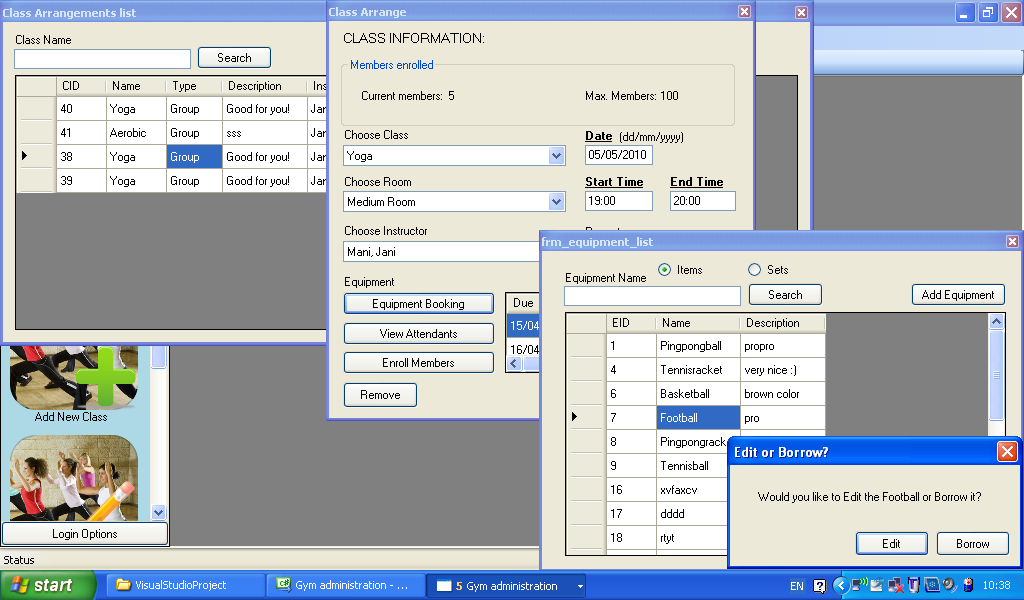
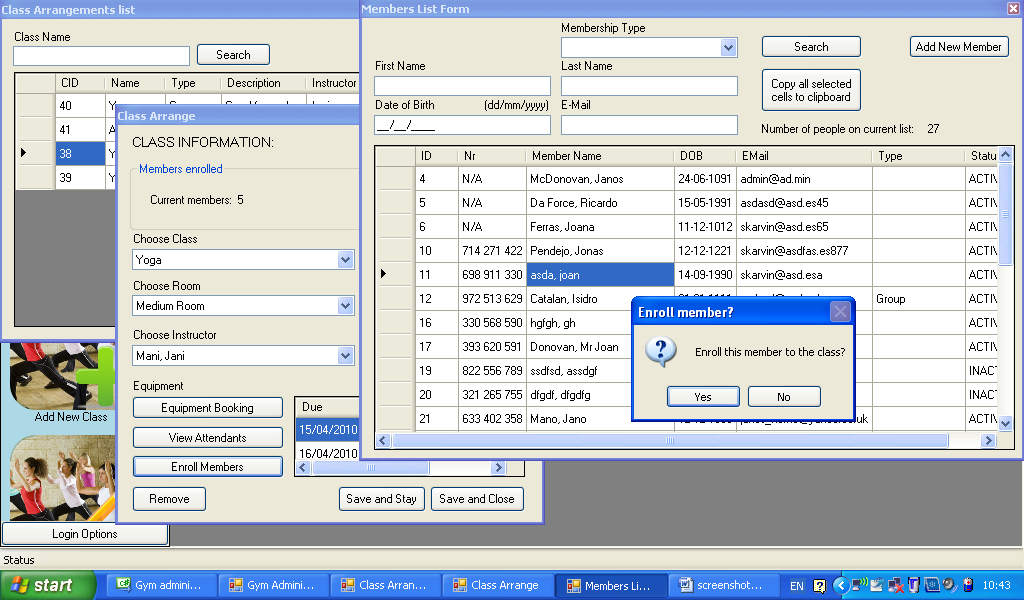
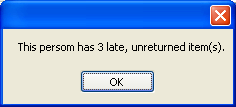
Click ‘add payments’ brings up the Member List so that the user can search for a Member.

Double click a booking

to return equipment.

Then double click the user that has made the payment to record the information.





Double click a member to remove them from the class.

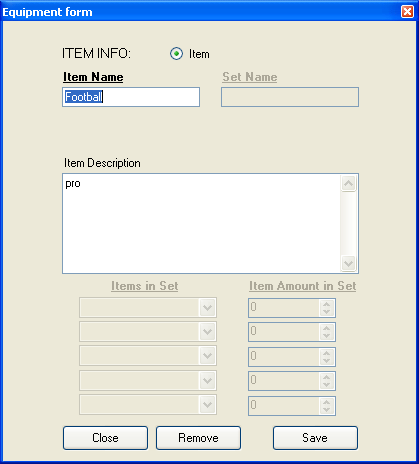
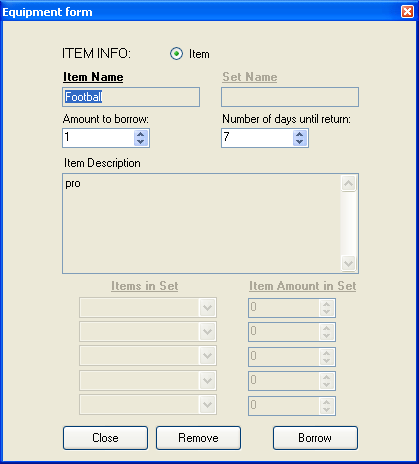
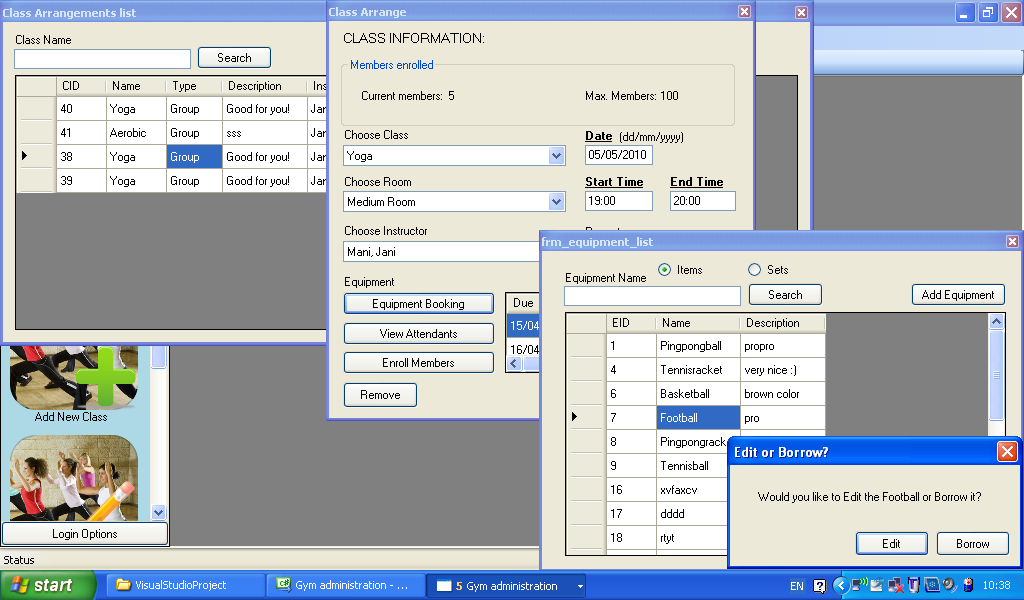
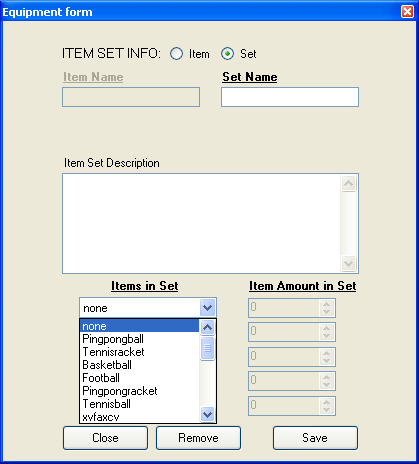
Click to bring up the list of members attending that class.

Once the user confirms their choice then a notice is displayed if they have equipment that needs returning.

Double click a Member you wish to enroll on the class.

Double click item or set to display options.

Add a Equipment Booking to this class.

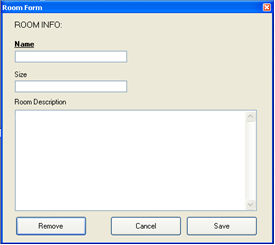


Staff Functions/Equipment List/Edit Equipment/Borrow Equipment/Add Equipment

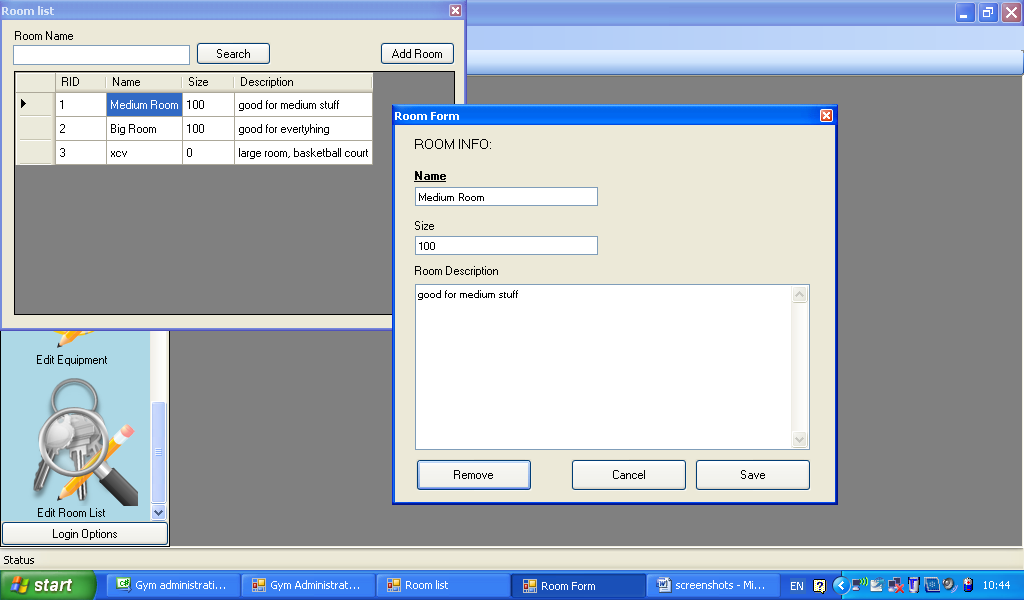
Adding equipment brings up a plant Equipment Form that the user can fill out and then save, the equipment will then appear on the Equipment List.

Borrowing equipment freezes the text fields and brings up the option for users to choose how many pieces of equipment they need.

Editing the equipment lets the user change the information stored in the text fields and then save those changes.

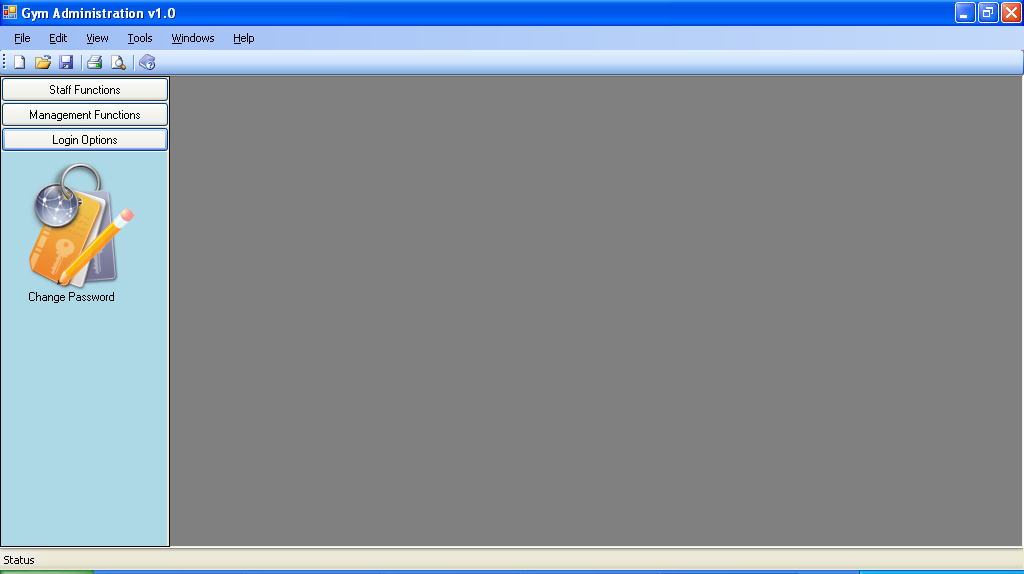
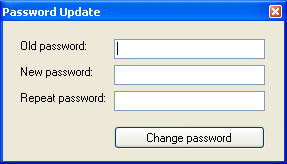


Clicking Add Room will open a empty Room Form that can then be filled in and saved. The room will then appear on the Room List.



Management Functions/Room List/Edit Room/Add Room

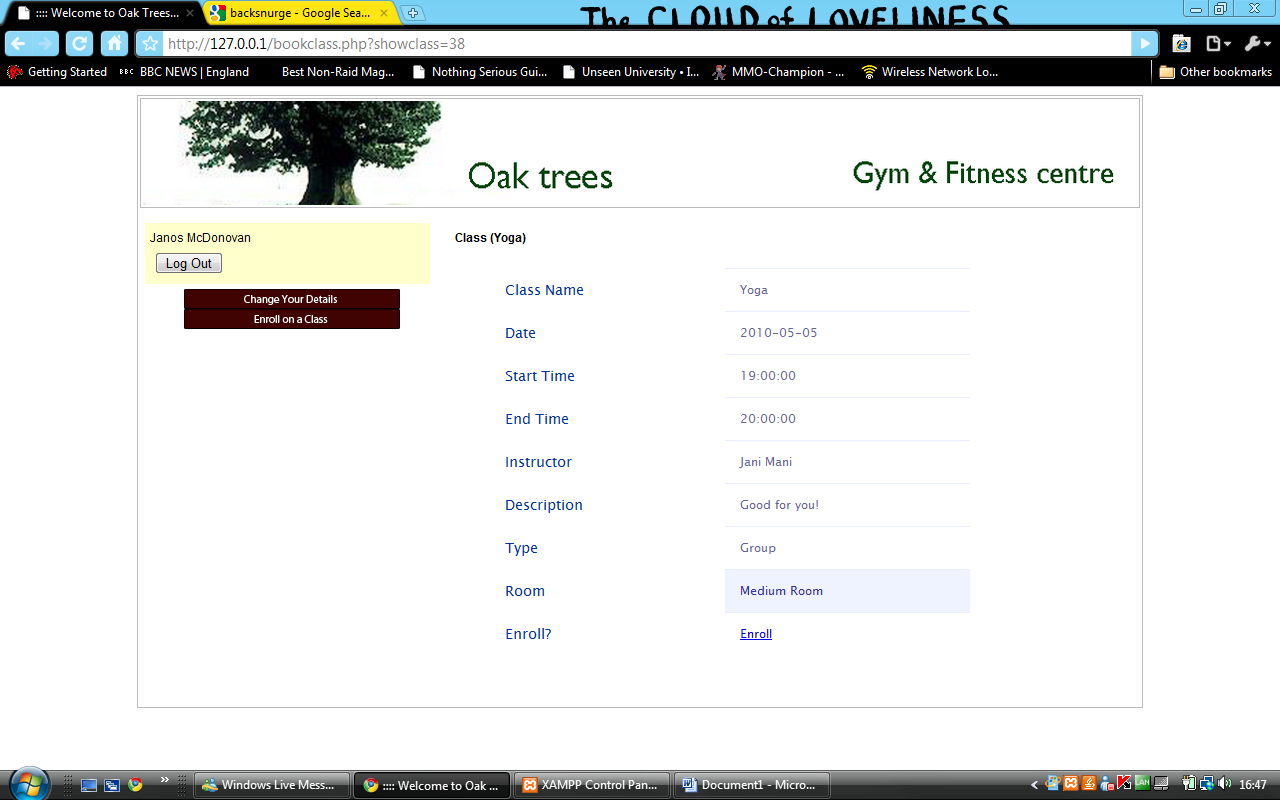
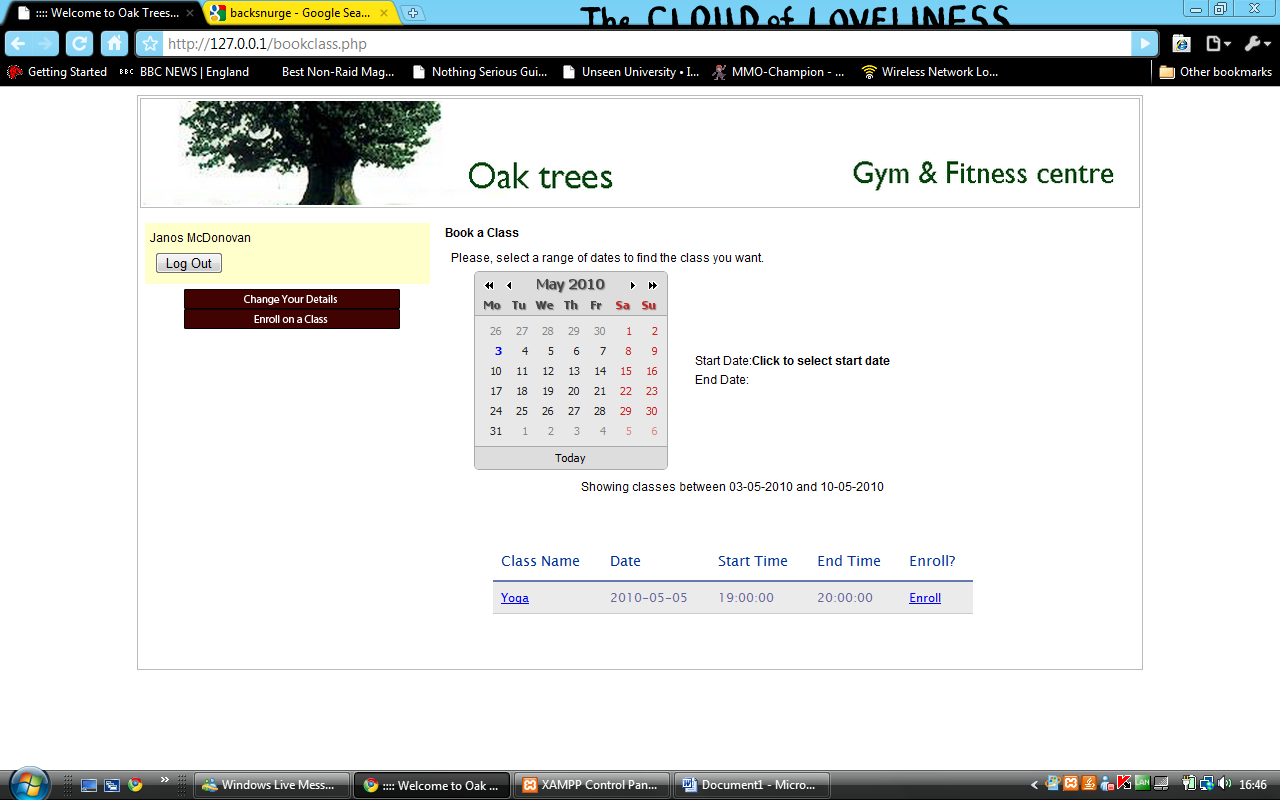
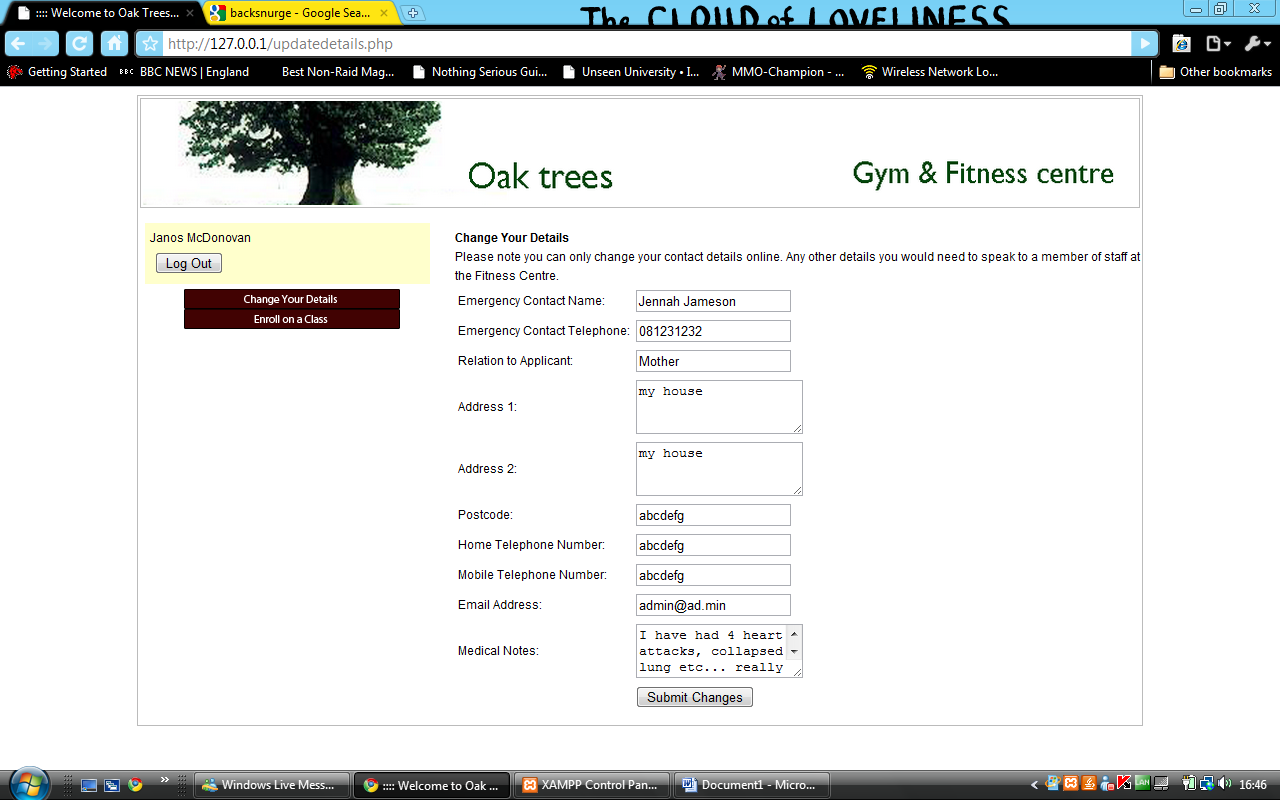
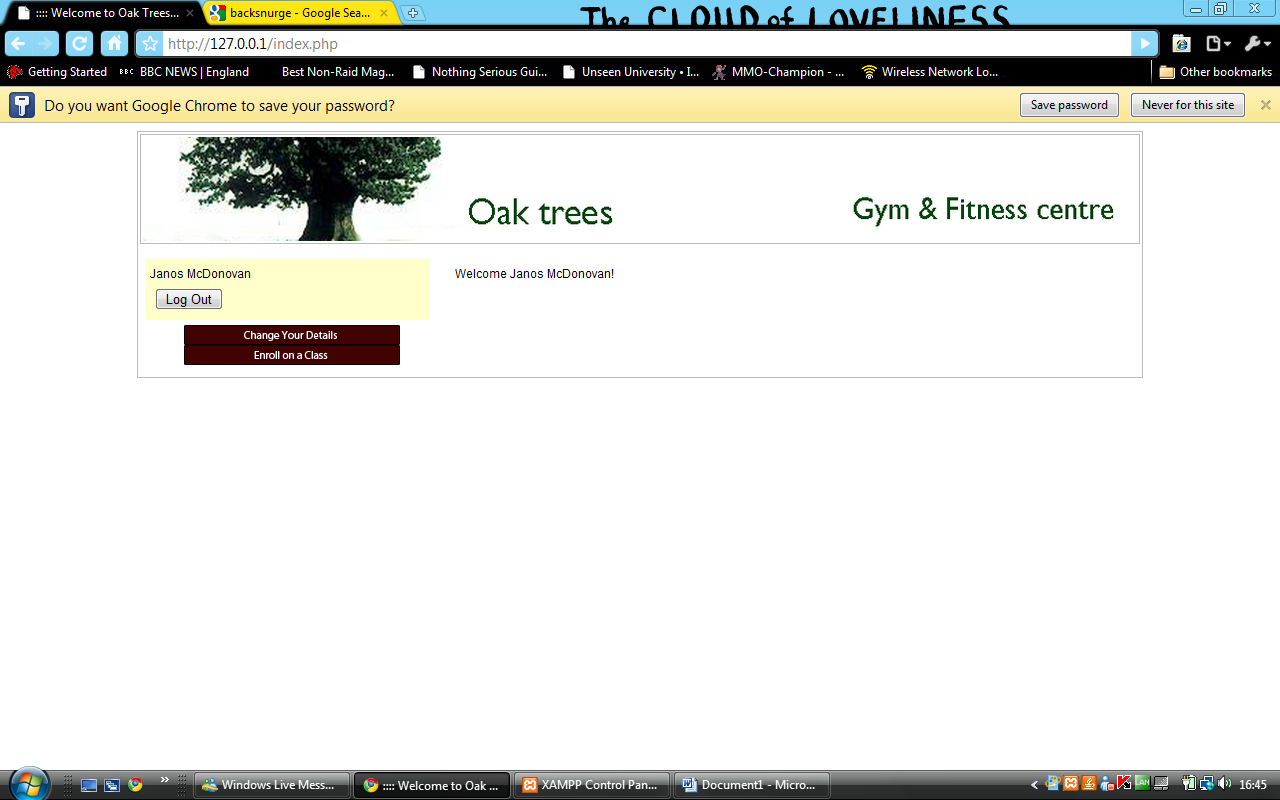
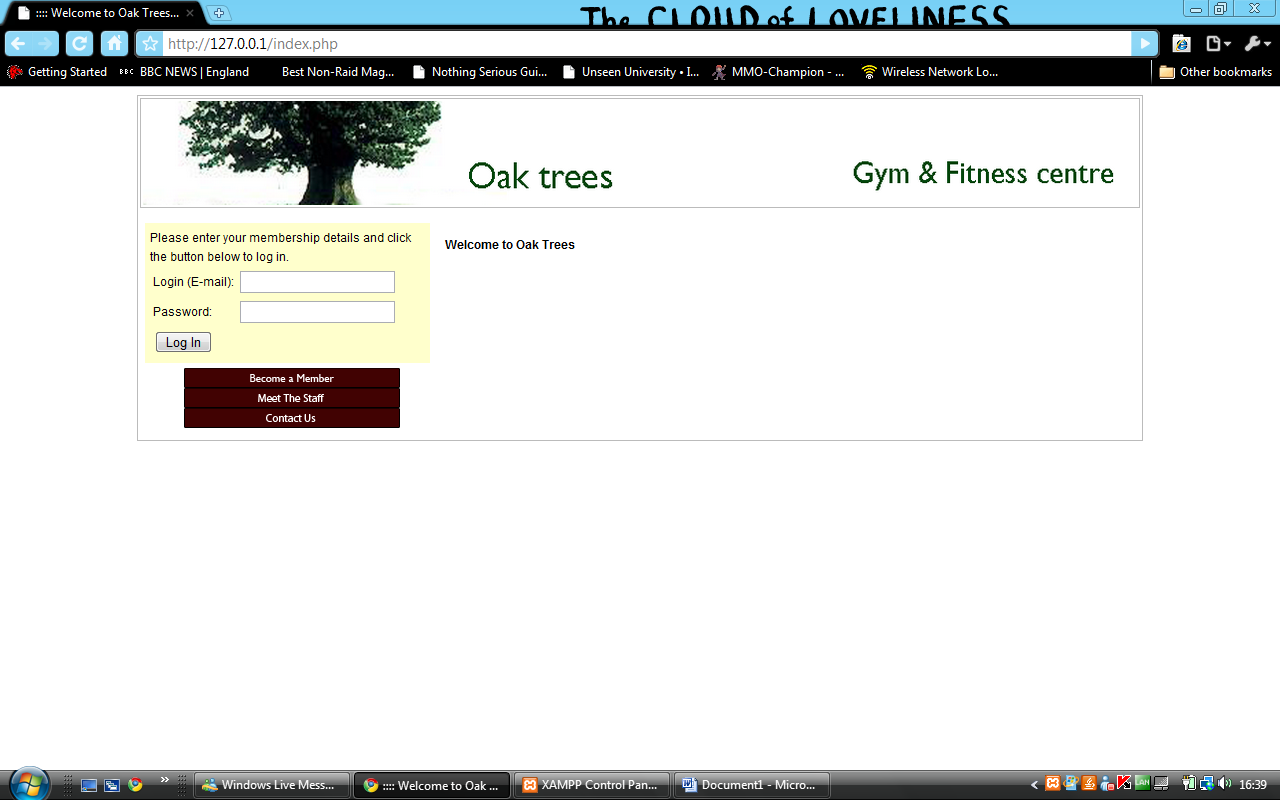
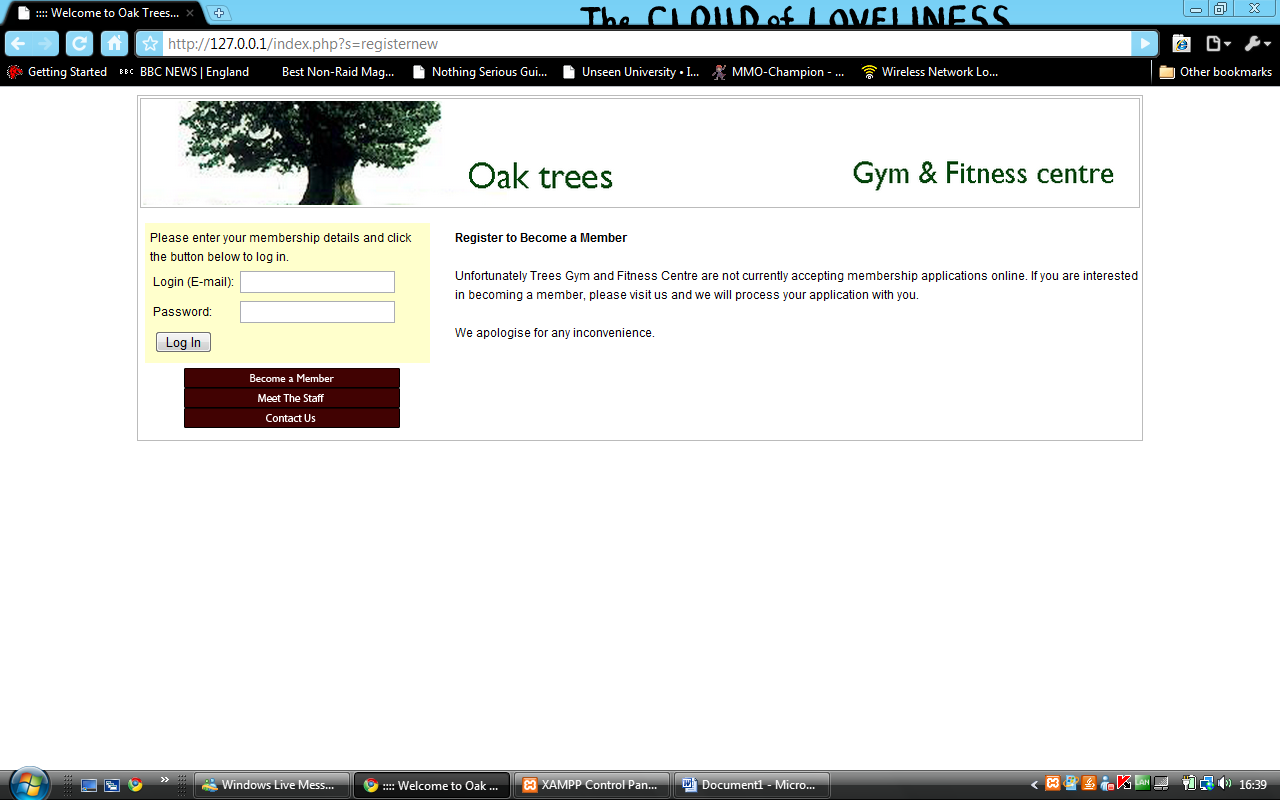
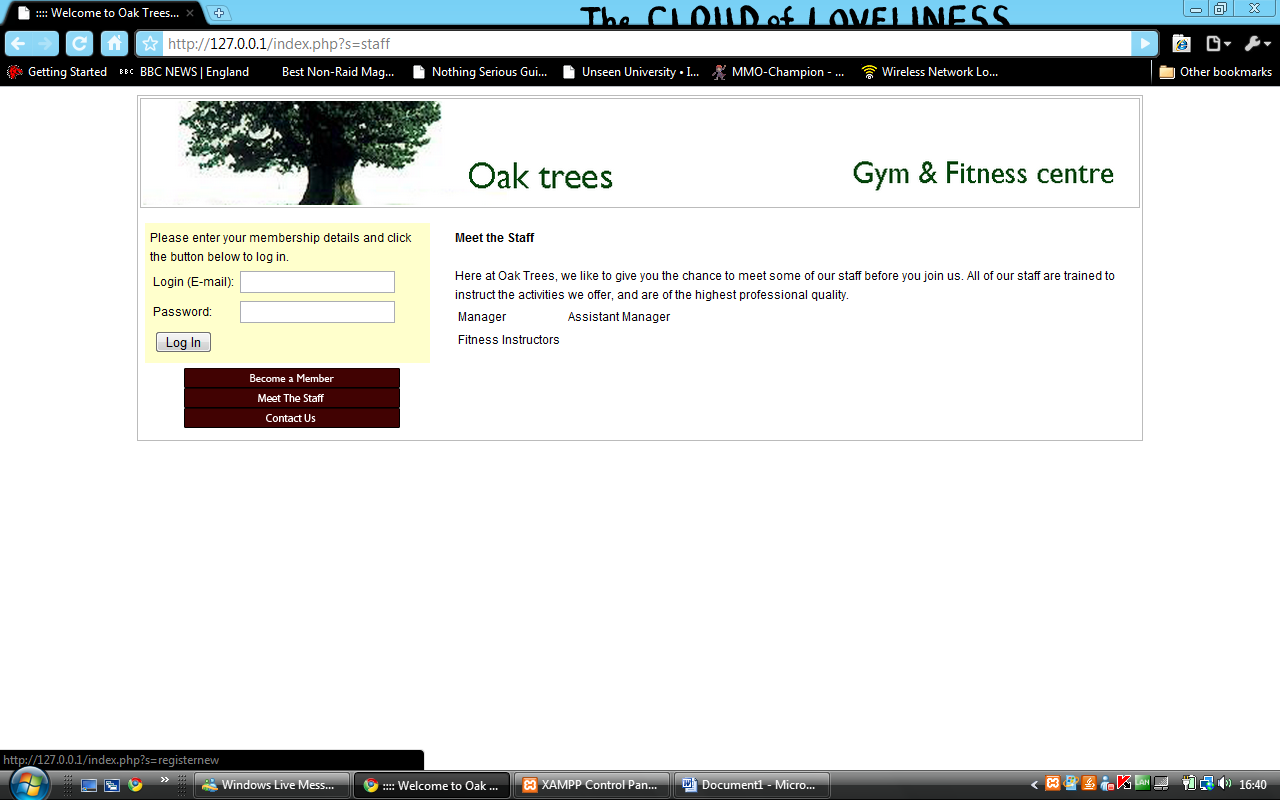
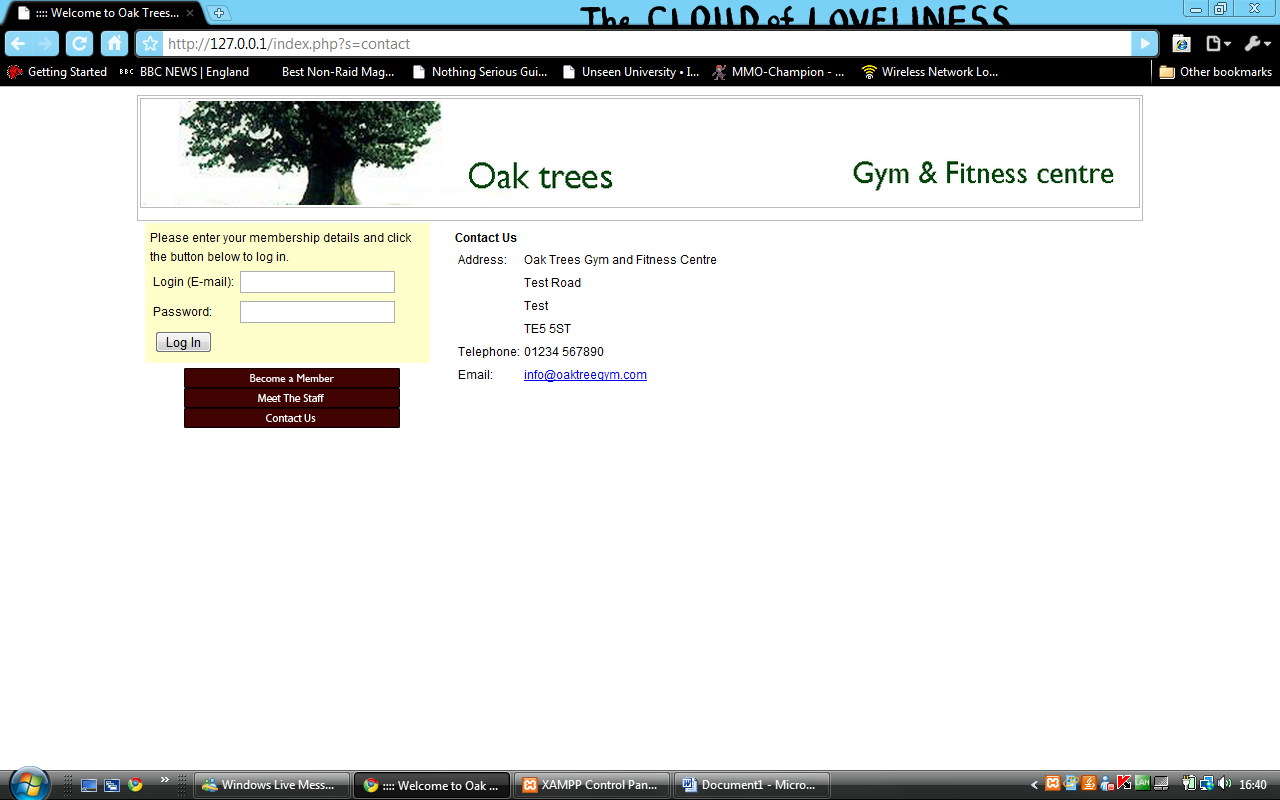
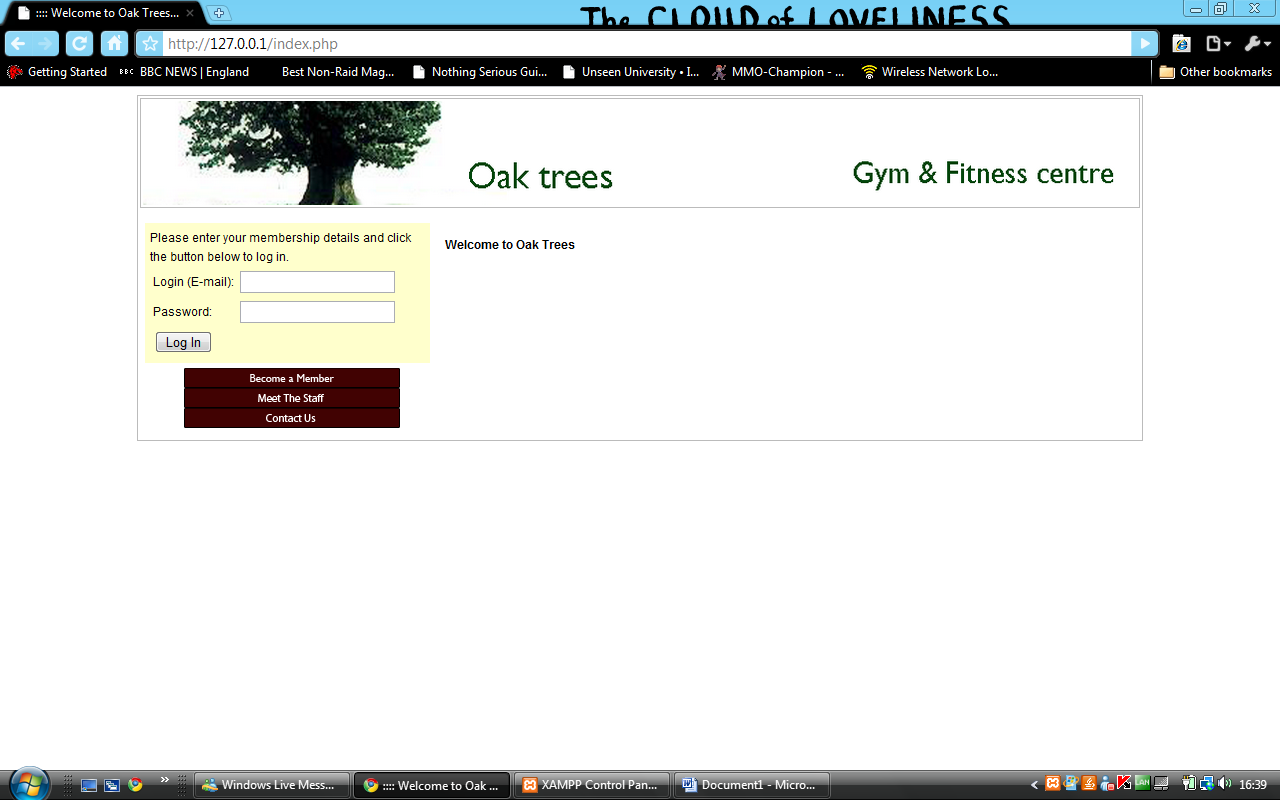
Double Click a room form the Room List and the information can then edited and saved. The room can also be removed from the list by clicking the button on the bottom left hand side.



Once logged in the user can choose this option to change their password.

Login Options/Change Password

# Website Functions

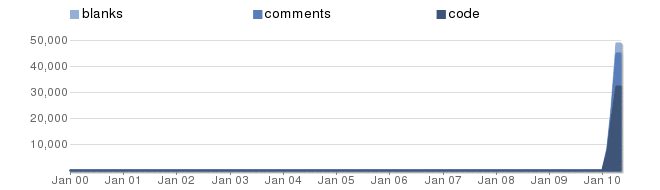


# Coding

Ohloh analyzes the project source code and determines the language of each line of code, excluding comments and blanks.

|  |  |  |
| --- | --- | --- |
| C# | 57% | anguages |
| PHP | 18% |
| XML | 12% |
| OTHER | 13% |

**Lines of Code**



**Lines of Code By Language**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Language** | **Code Lines** | **Comment Lines** | **Comment Ratio** | **Blank Lines** | **Total Lines** |
|  | [C#](https://www.ohloh.net/languages/17) | 18,458 | 7,448 | 28.8% | 1,871 | 27,777 |
|  | [PHP](https://www.ohloh.net/languages/2) | 5,889 | 3,192 | 35.2% | 1,325 | 10,406 |
|  | [XML](https://www.ohloh.net/languages/3) | 4,009 | 1,934 | 32.5% | 440 | 6,383 |
|  | [CSS](https://www.ohloh.net/languages/4) | 1,431 | 19 | 1.3% | 209 | 1,659 |
|  | [HTML](https://www.ohloh.net/languages/1) | 1,262 | 1 | 0.1% | 57 | 1,320 |
|  | [JavaScript](https://www.ohloh.net/languages/6) | 823 | 29 | 3.4% | 169 | 1,021 |
|  | [SQL](https://www.ohloh.net/languages/30) | 433 | 17 | 3.8% | 9 | 459 |

## Example of database table handler class (Member.cs)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Collections;

using System.Drawing;

using MySql.Data.MySqlClient;

using System.Data.SqlClient;

using System.Data;

using System.IO;

namespace Gym\_administration

{

/\*\*

\* @desc It holds data and modifying methods for the MEMBERS table.

\* Most closely associated form is frm\_member.

\* Most closely associated table is MEMBERS.

\* @params [none] Incoming parameters are described at the individual constructors.

\* @return [none] No directly returned data.

\* Returns of public methods are described at the individual methods.

\*/

class Member : Person

{

// A field with the same name from MEMBERS table

private int id\_member;

public int Id\_member

{

get { return id\_member; }

set { id\_member = value; }

}

// Field member\_number from MEMBERS table

private string memberNumber;

public string MemberNumber

{

get { return memberNumber; }

set { memberNumber = value; }

}

// Field type from MEMBERS table

private string type;

public string Type

{

get { return type; }

set { type = value; }

}

// Field is\_active from MEMBERS table

private bool isActive;

public bool IsActive

{

get { return isActive; }

set { isActive = value; }

}

// Field emerg\_contact\_name from MEMBERS table

private string emergContactName;

public string EmergContactName

{

get { return emergContactName; }

set { emergContactName = value; }

}

// Field emerg\_contact\_relation from MEMBERS table

private string emergContactRelation;

public string EmergContactRelation

{

get { return emergContactRelation; }

set { emergContactRelation = value; }

}

// Field emerg\_contact\_phone from MEMBERS table

private string emergContactPhone;

public string EmergContactPhone

{

get { return emergContactPhone; }

set { emergContactPhone = value; }

}

// Field emerg\_contact\_mobile from MEMBERS table

private string emergContactMobile;

public string EmergContactMobile

{

get { return emergContactMobile; }

set { emergContactMobile = value; }

}

// Field medical\_allergies from MEMBERS table

private string medicalAllergies;

public string MedicalAllergies

{

get { return medicalAllergies; }

set { medicalAllergies = value; }

}

// Field medical\_notes from MEMBERS table

private string medicalNotes;

public string MedicalNotes

{

get { return medicalNotes; }

set { medicalNotes = value; }

}

// Field id\_file from MEMBERS and FILE tabe for referencing from FILE table

private string id\_file;

public string Id\_file

{

get { return id\_file; }

set { id\_file = value; }

}

// Field file\_name from FILE table

private string fileName;

public string FileName

{

get { return fileName; }

set { fileName = value; }

}

// Variable for storing file path

private string filePath;

public string FilePath

{

get { return filePath; }

set { filePath = value; }

}

// Field medical\_doctor\_name from MEMBERS table

private string medicalDoctorName;

public string MedicalDoctorName

{

get { return medicalDoctorName; }

set { medicalDoctorName = value; }

}

// Field medical\_phone from MEMBERS table

private string medicalPhone;

public string MedicalPhone

{

get { return medicalPhone; }

set { medicalPhone = value; }

}

// Field ismale from MEMBERS table

private string gender;

public string Gender

{

get { return gender; }

set { gender = value; }

}

// a User (User.cs) object is stored here

private User clUser;

internal User ClUser

{

get { return clUser; }

set { clUser = value; }

}

/\*\*

\* @desc Default constructor.

\* Sets id\_member to -1 so the fact of this is a new member can be referenced.

\* Creates a new user parent class instance.

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public Member()

{

this.id\_member = -1;

this.clUser = new User();

}

/\*\*

\* @desc Constructor

\* Loads in all fields from a single row of the MEMBERS table.

\* @params [int] id\_member identifies the member uniquely.

\* @return [none] No directly returned data.

\*/

public Member(int id\_member)

{

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// Launch the query to return all fields from a single row of the MEMBERS table

List<Hashtable> lhResultset = conn.lhSqlQuery("Select \* from members m, users u where u.id\_user = m.id\_user AND m.id\_member = '" + id\_member + "'");

// Check if we found the member

if ((int)lhResultset.Count > 0)

{

// Fill in all member and parent user fields with table data

this.clUser = new User();

this.clUser.Id\_user = int.Parse(lhResultset[0]["id\_user"].ToString());

this.clUser.Login = lhResultset[0]["login"].ToString();

this.clUser.Password = lhResultset[0]["password"].ToString();

this.clUser.Profile = lhResultset[0]["profile"].ToString();

this.IsActive = (lhResultset[0]["is\_active"].ToString() == "True") ? true : false;

this.Id\_member = int.Parse(lhResultset[0]["id\_member"].ToString());

this.Address\_2 = lhResultset[0]["address\_2"].ToString();

this.Address\_1 = lhResultset[0]["address\_1"].ToString();

this.Birthdate = lhResultset[0]["birthdate"].ToString();

this.City = lhResultset[0]["city"].ToString();

this.County = lhResultset[0]["county"].ToString();

this.Email = lhResultset[0]["email"].ToString();

this.EmergContactMobile = lhResultset[0]["emerg\_contact\_mobile"].ToString();

this.EmergContactName = lhResultset[0]["emerg\_contact\_name"].ToString();

this.EmergContactPhone = lhResultset[0]["emerg\_contact\_phone"].ToString();

this.EmergContactRelation = lhResultset[0]["emerg\_contact\_relation"].ToString();

this.FirstName = lhResultset[0]["firstName"].ToString();

this.LastName = lhResultset[0]["lastName"].ToString();

this.MedicalAllergies = lhResultset[0]["medical\_allergies"].ToString();

this.MedicalDoctorName = lhResultset[0]["medical\_doctor\_name"].ToString();

this.MedicalNotes = lhResultset[0]["medical\_notes"].ToString();

this.MedicalPhone = lhResultset[0]["medical\_phone"].ToString();

this.MemberNumber = lhResultset[0]["member\_number"].ToString();

this.Id\_file = lhResultset[0]["id\_file"].ToString();

this.PostalCode = lhResultset[0]["postalcode"].ToString();

this.Type = lhResultset[0]["type"].ToString();

this.Mobile = lhResultset[0]["mobile"].ToString();

this.Phone = lhResultset[0]["phone"].ToString();

this.Gender = lhResultset[0]["gender"].ToString();

}

}

/\*\*

\* @desc Creates a new payment for the opened member

\* @params [decimal] amount: the amount payed

\* @params [string] date: the date of the payment

\* @params [string] desc: description/comments

\* @params [string] receiptNumber: receipt number

\* @params [string] paymentMethod: card/cash/cheque/bank transfer

\* @params [string] receivedBy: name of person, who received the payment

\* @return [bool] Returns true in case of success, false if there was a problem

\*/

public bool AddPayment(Decimal amount, string date, string desc, string receiptNumber, string paymentMethod, string receivedBy)

{

// The payment can be added to existing members only, not new members

if (Id\_member != -1)

{

// Create a payment object and copy into all payment data

Payment clPayment = new Payment();

clPayment.Amount = amount;

clPayment.Date = date;

clPayment.ClMember = this;

clPayment.Details = desc;

clPayment.ReceiptNumber = receiptNumber;

clPayment.PaymentMethod = paymentMethod;

clPayment.ReceivedBy = receivedBy;

// Save payment

if (clPayment.SavePayment())

return true;

else

return false;

}

else

return false;

}

/\*\*

\* @desc This method will save the object into the database

\* @return [bool] Returns true in case of success, false if there was a problem

\*/

public bool SaveMember()

{

// Convert date into mysql format

string mysqlDate = Utils.sGetMysqlDate(this.Birthdate);

string query;

// Check Birthdate format

if (mysqlDate == "0000-00-00")

{

MessageBox.Show("The Date of Birth is in incorrect format!");

}

// Check e-mail format

else if (Utils.bValidateEmail(this.Email) == false)

{

MessageBox.Show("The E-Mail address is incorrect!");

}

else

{

// First the user object is filled

clUser.IsActive = (this.IsActive) ? true : false;

clUser.Login = this.Email;

clUser.Password = mysqlDate;

clUser.Profile = "member";

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// If the User details were correctly saved

if (clUser.SaveUser())

{

// Check if there is a new picture to save

if ((this.FilePath != null) && (this.FilePath.Length > 1))

{

this.Id\_file = conn.uploadFileToDB(this.FilePath, this.FileName);

}

// The insert query is launched in case of existing members only, not new members

if (this.Id\_member == -1)

{

// Create insert query

query = "insert into `gym`.`members` (`id\_member`, `firstName`, `lastName`, `birthdate`, `address\_1`, `city`, `county`, `postalcode`, `type`, `id\_user`, `is\_active`, `address\_2`, `emerg\_contact\_name`, `emerg\_contact\_relation`, `emerg\_contact\_phone`, `emerg\_contact\_mobile`, `medical\_allergies`, `medical\_notes`, `id\_file`, `medical\_doctor\_name`, `medical\_phone`, `email`, `member\_number`, `phone`,`mobile`,`gender`) values " +

"(NULL, '" + this.FirstName + "', '" + this.LastName + "', '" + mysqlDate + "', '" + this.Address\_1 + "', '" + this.City + "', '" + this.County + "', '" + this.PostalCode + "', '" + this.Type + "', '" + clUser.Id\_user + "', '" + ((this.IsActive) ? "1" : "0") + "', '" + this.Address\_2 + "', '" + this.EmergContactName + "', '" + this.EmergContactRelation + "', '" + this.EmergContactPhone + "', '" + this.EmergContactMobile + "', '" + this.MedicalAllergies + "', '" + this.MedicalNotes + "', '" + this.Id\_file + "', '" + this.MedicalDoctorName + "', '" + this.MedicalPhone + "', '" + this.Email + "', '" + this.MemberNumber + "','" + this.Phone + "','" + this.Mobile + "','" + this.Gender + "')";

// Launch insert query

int id\_member = conn.InsertToDB(query);

// Check if the insert was successful

if (id\_member != -1)

{

this.Id\_member = id\_member;

MessageBox.Show("The new member has been added to the databse succesfully!");

return true;

}

else

{

MessageBox.Show("There was a problem adding the new user, please check your data!");

clUser.DeleteUser();

return false;

}

}

// This is a member update

else

{

// Create update query

query = "UPDATE members SET firstName = '" + this.FirstName + "', lastName = '" + this.LastName + "', birthdate = '" + mysqlDate + "', address\_1 = '" + this.Address\_1 + "', city = '" + this.City + "', county = '" + this.County + "', postalcode = '" + this.PostalCode + "', type = '" + this.Type + "', is\_active = " + ((this.IsActive) ? "1" : "0") + ", address\_2 = '" + this.Address\_2 + "', emerg\_contact\_name = '" + this.EmergContactName + "', emerg\_contact\_relation = '" + this.EmergContactRelation + "', emerg\_contact\_phone = '" + this.EmergContactPhone + "', emerg\_contact\_mobile = '" + this.EmergContactMobile + "', medical\_allergies = '" + this.MedicalAllergies + "', medical\_notes = '" + this.MedicalNotes + "', id\_file = '" + this.Id\_file + "', medical\_doctor\_name = '" + this.MedicalDoctorName + "', medical\_phone = '" + this.MedicalPhone + "', email = '" + this.Email + "', phone = '" + this.Phone + "', mobile = '" + this.Mobile +

"', gender = '" + this.Gender + "' " + " WHERE id\_member = '"+this.Id\_member+"'";

// Launch update query

int result = conn.DeleteOrUpdate(query);

// Check if the update was successful

if (result > 0)

{

MessageBox.Show("The member data has been updated succesfully!");

return true;

}

else

{

MessageBox.Show("There was a problem updating the user information, please check your data!");

clUser.DeleteUser();

return false;

}

}

}

// If the user saving was false, then it was becuase of duplicate e-mail at this point

else

{

MessageBox.Show("The e-mail already exists in the database! Please choose another one.");

return false;

}

}

return false;

}

/\*\*

\* @desc This method will set a member to be inactive in the database

\* @return [bool] Returns true in case of success, false if there was a problem

\*/

public bool RemoveMember()

{

// If an existing member is currently loaded in

if (this.Id\_member != -1)

{

// Set his active status to inactive

this.IsActive = false;

return this.SaveMember();

}

return false;

}

}

}

## Example of a form that interfaces table data (frm\_member.cs)

using System;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO;

namespace Gym\_administration

{

/\*\*

\* @desc Form Handler for members.

\* It is for adding or modifying a member.

\* @params [none] Incoming parameters are described at the individual constructors.

\* @return [none] No directly returned data.

\* Returns of public methods are described at the individual methods.

\*/

public partial class frm\_member : Form

{

Member clMember;

EquipmentBooked clEquipmentBooked;

frm\_member\_list frmMemberList;

/\*\*

\* @desc Default constructor for creating new member from main menu.

\* This is for loading from main menu,

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public frm\_member()

{

clMember = new Member();

InitializeComponent();

this.Text = "Add New Member Form";

txt\_membernum.Text = Utils.sGenerateNewMemberNumber();

txt\_membernum.ReadOnly = true;

// Members who don't yet exist can't book equipment

// This will be available once the "Save and Stay" is executed successfully

button\_equipmentbooking.Enabled = false;

// Members who don't yet exist can't book equipment

// This will be available once the "Save and Stay" is executed successfully

button\_payments.Enabled = false;

// Members who don't yet exist can't be deleted

button\_remove.Enabled = false;

clMember.Id\_file = "";

cmb\_type.SelectedIndex = 0;

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

rd\_male.Checked = true;

}

/\*\*

\* @desc Constructor for creating new member, that was opened from member list.

\* (To be able to refresh member list after saving the new member)

\* @params [frm\_member\_list] frmMemberList: by taking this parameter there will be a reference

\* to the member list so it can be refreshed after saving the new member

\* @return [none] No directly returned data.

\*/

public frm\_member(frm\_member\_list frmMemberList)

{

// Create reference to the parent form

this.frmMemberList = frmMemberList;

// create new member object

clMember = new Member();

InitializeComponent();

this.Text = "Add New Member Form";

txt\_membernum.Text = Utils.sGenerateNewMemberNumber();

txt\_membernum.ReadOnly = true;

// equipmentbooking, add payments and remove member buttons are hidden until saving (creating) the member

button\_equipmentbooking.Enabled = false;

button\_payments.Enabled = false;

button\_remove.Enabled = false;

// As this was opened from a member list there is no need to open a new one after closing

button\_saveOpen.Enabled = false;

clMember.Id\_file = "";

// Set a default type (Full time student)

cmb\_type.SelectedIndex = 0;

// Set default gender and image

rd\_male.Checked = true;

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

}

/\*\*

\* @desc Constructor for editing an existing member.

\* (To be able to refresh member list after saving the edited member)

\* @params [int] id\_member: identifies the member to modify

\* @params [frm\_member\_list] frmMemberList: by taking this parameter there will be a reference

\* to the member list so it can be refreshed after saving the edited member

\* @return [none] No directly returned data.

\*/

public frm\_member(int id\_member, frm\_member\_list frmMemberList)

{

InitializeComponent();

this.Text = "Edit Member Form";

// Create reference to the parent form

this.frmMemberList = frmMemberList;

// Load in member details for specified member

clMember = new Member(id\_member);

button\_equipmentbooking.Enabled = true;

button\_payments.Enabled = true;

button\_remove.Enabled = true;

button\_saveOpen.Enabled = false;

if (clMember.Id\_member < 1)

MessageBox.Show("The member could not be found");

else

{

// If the member was found, load in all member details into member object from database

vLoadBookedList();

txt\_firstName.Text = clMember.FirstName;

txt\_lastName.Text = clMember.LastName;

chk\_active.Checked = clMember.IsActive;

txt\_dob.Text = Utils.sGetCsharpDateFromMysqlDate(clMember.Birthdate);

txt\_address1.Text = clMember.Address\_1;

txt\_address2.Text = clMember.Address\_2;

txt\_city.Text = clMember.City;

txt\_county.Text = clMember.County;

txt\_emerg\_mobile.Text = clMember.EmergContactMobile;

txt\_emerg\_name.Text = clMember.EmergContactName;

txt\_emerg\_telephone.Text = clMember.EmergContactPhone;

txt\_emerg\_relation.Text = clMember.EmergContactRelation;

txt\_allergies.Text = clMember.MedicalAllergies;

txt\_doctor\_name.Text = clMember.MedicalDoctorName;

txt\_medical\_notes.Text = clMember.MedicalNotes;

txt\_doctor\_phone.Text = clMember.MedicalPhone;

txt\_membernum.Text = clMember.Id\_member.ToString();

txt\_pc.Text = clMember.PostalCode;

cmb\_type.Text = clMember.Type;

txt\_email.Text = clMember.Email;

txt\_mobile.Text = clMember.Mobile;

txt\_telephone.Text = clMember.Phone;

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// If there is a corresponing picture for this member, then load it in,

// else show default image, depending on gender

if (clMember.Gender == "male")

{

rd\_male.Checked = true;

if (clMember.Id\_file == "")

{

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

}

else

{

try

{

pictureBox1.Image = conn.loadImageFromDB(clMember.Id\_file);

}

catch (Exception e)

{

MessageBox.Show("Picture error, please assign anothe picture to this member!\r\n\r\n" + e);

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

}

}

}

else

{

rd\_female.Checked = true;

if (clMember.Id\_file == "")

{

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_female\_128;

}

else

{

try

{

pictureBox1.Image = conn.loadImageFromDB(clMember.Id\_file);

}

catch (Exception e)

{

MessageBox.Show("Picture error, please assign anothe picture to this member!\r\n\r\n" + e);

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_female\_128;

}

}

}

}

}

/\*\*

\* @desc This method refreshes the booked equipment list

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public void vLoadBookedList()

{

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// Create source for grid

BindingSource itemsSource = new BindingSource();

// Create query

string query = "SELECT DISTINCT eb.date\_due Due, e.name Name, eb.borrowedamount Amount, eb.id\_equipment EqID, eb.id\_eq\_booking BkID FROM equipment e, equipment\_bookings eb WHERE eb.id\_member = " + clMember.Id\_member + " AND (eb.isreturned = 0 OR eb.isreturned is NULL) AND eb.id\_equipment = e.id\_equipment ORDER BY Due";

// Launch query and load result into source

itemsSource.DataSource = conn.dtGetTableForDataGrid(query);

// Assign source to grid

dg\_eqbookings.DataSource = itemsSource;

dg\_eqbookings.AllowUserToAddRows = false;

dg\_eqbookings.ReadOnly = true;

// Check for unreturned items, if there is any, then report ir to the user!

if (dg\_eqbookings.RowCount > 0)

{

int lateItems = 0;

int rowIndex;

for (rowIndex = 0; rowIndex < dg\_eqbookings.RowCount; rowIndex++)

{

string eqDueDate = dg\_eqbookings.Rows[rowIndex].Cells[0].Value.ToString();

DateTime today = DateTime.Today;

DateTime due = DateTime.Parse(eqDueDate);

int result = DateTime.Compare(today, due);

if (result > 0)

lateItems++;

}

if (lateItems > 0)

MessageBox.Show("This persom has " + lateItems + " late, unreturned item(s).");

}

}

/\*\*

\* @desc Executes when the "Save and Stay" button is clicked

\* It calls for saving, then leaves the form open for further editing

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_saveStay\_Click(object sender, EventArgs e)

{

if (saveClick())

{

button\_equipmentbooking.Enabled = true;

button\_payments.Enabled = true;

button\_remove.Enabled = true;

}

}

/\*\*

\* @desc Executes when the "Save and Close", "Save and Open List" or "Save and Stay" button is clicked

\* It copies user input into member object and then calls for saving the member

\* @params [none] No input parameter.

\* @return [bool] Returns true in case of success.

\*/

private bool saveClick()

{

// Copy all form fields into member object fields

clMember.FirstName = txt\_firstName.Text;

clMember.LastName = txt\_lastName.Text;

clMember.IsActive = (chk\_active.Checked) ? true : false;

clMember.Birthdate = txt\_dob.Text;

clMember.Address\_2 = txt\_address2.Text;

clMember.Address\_1 = txt\_address1.Text;

clMember.City = txt\_city.Text;

clMember.County = txt\_county.Text;

clMember.EmergContactMobile = txt\_emerg\_mobile.Text;

clMember.EmergContactName = txt\_emerg\_name.Text;

clMember.EmergContactPhone = txt\_emerg\_telephone.Text;

clMember.EmergContactRelation = txt\_emerg\_relation.Text;

clMember.MedicalAllergies = txt\_allergies.Text;

clMember.MedicalDoctorName = txt\_doctor\_name.Text;

clMember.MedicalNotes = txt\_medical\_notes.Text;

clMember.MedicalPhone = txt\_doctor\_phone.Text;

clMember.MemberNumber = txt\_membernum.Text;

clMember.PostalCode = txt\_pc.Text;

clMember.Type = cmb\_type.Text;

clMember.Email = txt\_email.Text;

clMember.Phone = txt\_telephone.Text;

clMember.Mobile = txt\_mobile.Text;

if (rd\_male.Checked == true)

clMember.Gender = "male";

else

clMember.Gender = "female";

// Call for saving the member

return clMember.SaveMember();

}

/\*\*

\* @desc Executes when the "Payments" button is clicked

\* It displays the add payment form instantiated for this specific member

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_payments\_Click(object sender, EventArgs e)

{

frm\_payment\_list frmPayments = new frm\_payment\_list(clMember.Id\_member);

frmPayments.ShowDialog();

}

// Close/Cancel

private void button\_cancel\_Click(object sender, EventArgs e)

{

this.Close();

}

/\*\*

\* @desc Executes when the "Remove" button is clicked

\* It asks for confirmation and then calls for removing the member

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_remove\_Click(object sender, EventArgs e)

{

// If there are still equipments borrowed, the member can't be deleted!

if (dg\_eqbookings.RowCount > 0)

MessageBox.Show("You can't remove this member as the borrowed equipments has to be returned first!");

// else there are no outstanding borrowed equipments

else

{ // Confirm member removal

DialogResult result = MessageBox.Show("Are you sure?", "Delete entry", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Remove the member

if (clMember.RemoveMember())

{

// refresh parent member list if this was called from a member list

if (frmMemberList != null) this.frmMemberList.vLoadMemberList();

this.Close();

}

}

}

}

/\*\*

\* @desc Executes when the "Equipment Booking" button is clicked

\* It displays the equipment list for borrowing form instantiated for this specific member

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_equipmentbooking\_Click(object sender, EventArgs e)

{

frm\_equipment\_list frmEquipmentList = new frm\_equipment\_list(clMember.Id\_member, this);

frmEquipmentList.ShowDialog();

}

/\*\*

\* @desc Executes when a grid cell is double clicked on the borrowed equipment list

\* It loads in the equipment return dialog belonging to the cell

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void dg\_eqbookings\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

// Retrieve equipment booking details from grid row

string equipmentName = dg\_eqbookings.Rows[e.RowIndex].Cells[1].Value.ToString();

int borrowedAmount = int.Parse(dg\_eqbookings.Rows[e.RowIndex].Cells[2].Value.ToString());

int id\_eq\_booking = int.Parse(dg\_eqbookings.Rows[e.RowIndex].Cells[4].Value.ToString());

// Show return dialog for confirming amount to be returned

frm\_message\_box myMessageBox = new frm\_message\_box();

string result = myMessageBox.ShowBox(Utils.MB\_CUST4, "", "How many "+equipmentName+" would you like to return?",borrowedAmount.ToString());

// Reference how to use TryParse

//ref (Microsoft, 2006)

// Check the result of user input

double Num;

bool isNum = double.TryParse(result, out Num);

if (isNum)

{

// If there is something to return but not everything

if ((int.Parse(result) > 0) && (result != "Cancel"))

{

// Save the new amount into eq. booking

this.clEquipmentBooked = new EquipmentBooked(id\_eq\_booking);

this.clEquipmentBooked.BorrowedAmount = int.Parse(result);

this.clEquipmentBooked.IsReturned = false;

this.clEquipmentBooked.SaveEquipmentBooking();

}

// If all amount of this booking is to be returned

else

{

// Mark the booking as returned

this.clEquipmentBooked = new EquipmentBooked(id\_eq\_booking);

this.clEquipmentBooked.BorrowedAmount = 0;

this.clEquipmentBooked.IsReturned = true;

this.clEquipmentBooked.SaveEquipmentBooking();

}

// Refresh eq. booking list

this.vLoadBookedList();

}

}

/\*\*

\* @desc Executes when rd\_male radiobutton is checked

\* Sets default male image as background image if no picture is loaded

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void rd\_male\_Checked(object sender, EventArgs e)

{

if (pictureBox1.Image == null)

{

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

}

}

/\*\*

\* @desc Executes when rd\_female radiobutton is checked

\* Sets default female image as background image if no picture is loaded

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void rd\_female\_Checked(object sender, EventArgs e)

{

if (pictureBox1.Image == null)

{

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_female\_128;

}

}

/\*\*

\* @desc Executes when pictureBox1 is double clicked

\* Lets the user to load in a piture for the current member

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void pictureBox1\_DoubleClick(object sender, EventArgs e)

{

// Try to open an user selected image file

try

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.jpg; \*.jpeg; \*.gif; \*.bmp; \*.png)|\*.jpg; \*.jpeg; \*.gif; \*.bmp; \*.png";

if (open.ShowDialog() == DialogResult.OK)

{

// Store the file's path

clMember.FilePath = open.FileName;

// Create a fileinfo object for the file

FileInfo file = new FileInfo(clMember.FilePath);

// Store file's name

clMember.FileName = file.Name;

// Display the image on form

pictureBox1.Image = new Bitmap(clMember.FilePath);

// Delete the default image from background

this.pictureBox1.BackgroundImage = null;

}

}

catch (Exception)

{

throw new ApplicationException("Failed loading image");

}

}

/\*\*

\* @desc Executes mouse hovers over pictureBox1

\* It lets the user remove the association between member and its current picture

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

switch (e.Button)

{

// In case of right mouse click

case (MouseButtons.Right):

{

// If there is a picture currently loaded in and there is an association between the picture and the member

if ((this.pictureBox1.Image != null) && (clMember.Id\_file != null))

{

// Ask user confirmation

frm\_message\_box frmMessageBox = new frm\_message\_box();

string result = frmMessageBox.ShowBox(Utils.MB\_YESNO, "Would you like to delete the picture?", "Delete?");

if (result == "YES")

{

// remove all association between member and current picture

this.pictureBox1.Image = null;

clMember.Id\_file = "";

clMember.FileName = "";

clMember.FilePath = "";

// Warn the user that he has to save the modifications

MessageBox.Show("Image has been marked for deletion,\r\nyou must click on save for\r\nthe deletion to take effect!");

// Display default images as per genders

if (rd\_male.Checked)

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_male\_128;

else

this.pictureBox1.BackgroundImage = global::Gym\_administration.Properties.Resources.member\_female\_128;

}

}

break;

}

}

}

/\*\*

\* @desc Executes when the "Save and Close" button is clicked

\* If the saving is ok, then closes the member form

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_saveClose\_Click(object sender, EventArgs e)

{

// If saving the member was successful

if (this.saveClick())

{

// Refresh the list in parent window and close this one

if (this.frmMemberList != null) this.frmMemberList.vLoadMemberList();

this.Close();

}

}

/\*\*

\* @desc Executes when the "Save and Open" button is clicked

\* If the saving is ok, then closes the member form and opens up the member list

\* This button is never shown on a member form which was just called form a member list.

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_saveOpen\_Click(object sender, EventArgs e)

{

if (this.saveClick())

{

this.Dispose();

frm\_member\_list frmMemberList = new frm\_member\_list();

frmMemberList.ShowDialog();

}

}

// Fill in county if city iS Cambridge

private void txt\_city\_TextChanged(object sender, EventArgs e)

{

if (txt\_city.Text == "Cambridge")

txt\_county.Text = "Cambridgeshire";

}

// Restore default medical history text

private void button\_RestoreMedicalText\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("This will delete all current medical notes!\r\nAre you sure?\r\n\r\n(You must click on save for\r\nthe modifications to take effect!)", "Are you sure?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(frm\_member));

this.txt\_medical\_notes.Text = resources.GetString("txt\_medical\_notes.Text");

}

}

}

}

## Example of listing collection of entities (frm\_member\_list.cs)

using System;

using System.Collections.Generic;

using System.Collections;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Gym\_administration

{

/\*\*

\* @desc

\* Form Handler for listing members

\* @params [none] Incoming parameters are described at the individual constructors.

\* @return [none] No directly returned data.

\* Returns of public methods are described at the individual methods.

\*/

public partial class frm\_member\_list : Form

{

ClassInstance clClassInstance = null;

frm\_payment\_list frmPayments = null;

public bool viewAttendants;

/\*\*

\* @desc Default constructor for creating member list from main menu.

\* This is for loading from main menu,

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public frm\_member\_list()

{

InitializeComponent();

this.clClassInstance = new ClassInstance();

vLoadMemberList();

}

/\*\*

\* @desc Constructor for creating member list for adding new payment from payments list.

\* This is for loading from frm\_payment\_list.

\* @params [frm\_payments] frmPayments: This is for reference to the parent class.

\* @return [none] No directly returned data.

\*/

public frm\_member\_list(frm\_payment\_list frmPayments)

{

InitializeComponent();

this.frmPayments = frmPayments;

vLoadMemberList();

}

/\*\*

\* @desc Constructor for creating member list for adding new attendant

\* to a class instance or to view members who are currently attending

\* This is for loading from a class\_instance\_arrange.

\* @params [ClassInstance] clClassInstance: This is for reference to the parent class.

\* @params [bool] viewAttendants: This decides if viewing or adding attendants.

\* @return [none] No directly returned data.

\*/

public frm\_member\_list(ClassInstance clClassInstance, bool viewAttendants)

{

InitializeComponent();

this.clClassInstance = clClassInstance;

this.viewAttendants = viewAttendants;

vLoadMemberList();

}

/\*\*

\* @desc This method refreshes the member list

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public void vLoadMemberList()

{

string query = "";

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// Create source for grid

BindingSource itemsSource = new BindingSource();

// If this is not for viewing current attendants for a class instance

if (this.viewAttendants == false)

// Create query for displaying all members

query = "SELECT id\_member ID, member\_number Nr, CONCAT(lastName, ', ', firstName) 'Member Name', DATE\_FORMAT(birthdate,\"%d-%m-%Y\") DOB, email 'EMail', type Type, IF((is\_active= 0), 'INACTIVE','ACTIVE') Status FROM members ORDER BY ID ";

// If this is for viewing current attendants for a class instance

else

// Create query for displaying only members who attend this class instance

query = "SELECT m.id\_member ID, m.member\_number Nr, CONCAT(m.lastName, ', ', m.firstName) 'Member Name', DATE\_FORMAT(m.birthdate,\"%d-%m-%Y\") DOB, email 'EMail', type Type, IF((is\_active= 0), 'INACTIVE','ACTIVE') Status FROM members m, class\_bookings cb WHERE m.id\_member = cb.id\_member AND cb.id\_class\_instance = '" + this.clClassInstance.Id\_class\_instance + "' ORDER BY ID";

// Launch query and load result into source

itemsSource.DataSource = conn.dtGetTableForDataGrid(query);

// Assign source to grid

dg\_members.DataSource = itemsSource;

dg\_members.AllowUserToAddRows = false;

dg\_members.ReadOnly = true;

// Display current number of people on list

label\_numberOfPeople.Text = dg\_members.RowCount.ToString();

}

/\*\*

\* @desc Executes when a grid cell is double clicked on the member list

\* It loads in the member belonging to the cell for editing

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void dg\_members\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

try

{

// Get the member id at current cell

int id\_member = int.Parse(dg\_members.Rows[e.RowIndex].Cells[0].Value.ToString());

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// If this member list was launched from the payment list panel

if (this.frmPayments != null)

{

// Launch Add Payment panel for the selected member

frm\_add\_payment frmAddPayment = new frm\_add\_payment(id\_member);

frmAddPayment.ShowDialog();

// As this member list was launched from the payment list, refresh the payment list when this finishes

this.frmPayments.vloadDgPayments();

this.Close();

return;

}

// If this member list was launched from class instance list for adding new members

if (this.clClassInstance.Id\_class\_instance != -1 && this.viewAttendants == false)

{

// Confirm enrollment

DialogResult dialogResult = MessageBox.Show("Enroll this member to the class?", "Enroll member?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

// Check the room size

string query = "SELECT COUNT(\*) q FROM gym.class\_bookings WHERE id\_class\_instance = '" + this.clClassInstance.Id\_class\_instance + "'";

List<Hashtable> lhRes = conn.lhSqlQuery(query);

int currMembers = int.Parse(lhRes[0]["q"].ToString());

query = "SELECT r.size FROM gym.class\_instance ci, gym.rooms r WHERE ci.id\_room = r.id\_room AND ci.id\_class\_instance = '" + this.clClassInstance.Id\_class\_instance + "'";

lhRes = conn.lhSqlQuery(query);

int maxMembers = int.Parse(lhRes[0]["size"].ToString());

if (maxMembers < currMembers + 1)

{

MessageBox.Show("Sorry! This room does not allow more bookings!");

return;

}

Member clMember = new Member(id\_member);

this.clClassInstance.LclAttendants.Add(clMember);

this.clClassInstance.SaveClassInstance();

}

}

// If this member list was launched from class instance list for viewing attending members

if (this.clClassInstance.Id\_class\_instance != -1 && this.viewAttendants == true)

{

// Confirm removal

DialogResult dialogResult = MessageBox.Show("Remove this member from the class?", "Delete entry?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (dialogResult == DialogResult.Yes)

{

// Create delete query

string deleteClassBookingQuery = "delete from class\_bookings WHERE id\_member = '" + id\_member + "'" + " AND id\_class\_instance = '" + this.clClassInstance.Id\_class\_instance + "'";

// Launch delete query

int result = conn.DeleteOrUpdate(deleteClassBookingQuery);

// Check delete result

if (result > 0)

{

MessageBox.Show("The attendant has been removed from this class instance!");

vLoadMemberList();

}

else

{

MessageBox.Show("There was a problem updating the class booking information, please check your data!");

return;

}

}

}

// If this member list was launched from main menu just create an edit member form

else

{

frm\_member frmMember = new frm\_member(id\_member, this);

frmMember.ShowDialog();

}

}catch(Exception)

{

return;

}

}

/\*\*

\* @desc Executes when the Search button is clicked

\* It creates a new list of members based on search criteria given by the user

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_search\_Click(object sender, EventArgs e)

{

// Create mysql connection

mySqlConn conn = new mySqlConn();

conn.connect();

// Create source for grid

BindingSource itemsSource = new BindingSource();

// Create start of query

string query = "SELECT id\_member ID, member\_number Nr, CONCAT(lastName, ', ', firstName) 'Member Name', DATE\_FORMAT(birthdate,\"%d-%m-%Y\") DOB, email 'EMail', type Type, IF((is\_active= 0), 'INACTIVE','ACTIVE') Status FROM members WHERE 1 = 1 ";

// Check user input and create query for search

if (txt\_firstName.Text != "")

query += " AND firstName LIKE '%"+txt\_firstName.Text+"%'";

if (txt\_lastName.Text != "")

query += " AND lastName LIKE '%" + txt\_lastName.Text + "%'";

if (txt\_email.Text != "")

query += " AND email LIKE '%" + txt\_email.Text + "%'";

if (cmb\_type.SelectedIndex != -1)

{

query += " AND type LIKE '%" + cmb\_type.SelectedItem.ToString() + "%'";

}

string sDate = Utils.sGetMysqlDate(txt\_dob.Text);

if (sDate != "0000-00-00")

query += " AND birthdate = '" + sDate + "'";

query += " ORDER BY ID";

// Launch query and load result into source

itemsSource.DataSource = conn.dtGetTableForDataGrid(query);

// Assign source to grid

dg\_members.DataSource = itemsSource;

dg\_members.AllowUserToAddRows = false;

dg\_members.ReadOnly = true;

// Display amount of people currently on list

label\_numberOfPeople.Text = dg\_members.RowCount.ToString();

}

/\*\*

\* @desc Executes when the Copy button is clicked

\* It copies the contents of the currently selected grids onto the clipboard

\* with the optional delimiter

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_copy\_Click(object sender, EventArgs e)

{

// Create a string array of the size (amount) of selected cells

string[] saSelectedCellValues = new string[dg\_members.SelectedCells.Count];

int counter = 0;

string cellValue;

// Fill in all selected cell contents into string array

for (counter = 0; counter < (dg\_members.SelectedCells.Count); counter++)

{

cellValue = dg\_members.SelectedCells[counter].Value.ToString();

if (cellValue.Length != 0)

saSelectedCellValues[counter] = cellValue;

}

// Offer choice of delimiter

frm\_message\_box frmMessageBox = new frm\_message\_box();

string result = frmMessageBox.ShowBox(Utils.MB\_CUST2, "Which delimiter would you like to use? \r\n Its normally a comma (,) some mail uses semicolon (;)", "Mass e-mail delimiter selection", ",", ";");

char delimiter = result[0];

// Copy everything to clipboard

Clipboard.SetData(DataFormats.Text, string.Join(delimiter+" ", saSelectedCellValues));

// If let the user know about the result

if (Clipboard.ContainsData(DataFormats.Text))

MessageBox.Show(Clipboard.GetData(DataFormats.Text) + "\r\n\r\n is on the Clipboard now!");

else

MessageBox.Show("Nothing was selected");

}

/\*\*

\* @desc Executes when the Add Member button is clicked

\* It invokes the frm\_member panel for adding a new member

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_addNewMember\_Click(object sender, EventArgs e)

{

frm\_member frmMember = new frm\_member(this);

frmMember.ShowDialog();

}

}

}

## Example of mysql database connection utils (mySqlConn.cs)

using System;

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using System.Text;

// MySql Connector must be installed at this point

using MySql.Data.MySqlClient;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

using System.Data;

using System.IO;

namespace Gym\_administration

{

/\*\*

\* @desc It holds methods for managing mysql connections and queries

\* @params [none] Incoming parameters are described at the individual constructors.

\* @return [none] No directly returned data.

\* Returns of public methods are described at the individual methods.

\*/

class mySqlConn

{

string server;

string database;

string user;

string password;

string MyConString;

MySqlConnection connection;

/\*\*

\* @desc The constructor specifying the connection parameters

\*

\* @param string server

\* Server IP

\* @param string database

\* The database Name

\* @param string user

\* The username

\* @param string password

\* The password

\*/

public mySqlConn(string server, string database, string user, string password)

{

this.server = server;

this.database = database;

this.user = user;

this.password = password;

}

/\*\*

\* @desc The constructor by default with the connection information

\*

\*/

public mySqlConn()

{

this.server = "localhost";

this.database = "gym";

this.user = "gym";

this.password = "gym";

}

/\*\*

\* @desc Returns the connection string.

\*

\* @return [string]

\* The connection string returned

\*

\*/

public string GetMyConnString()

{

return this.MyConString;

}

/\*\*

\* @desc Connects to the databse using the specified query string

\*

\*/

public void connect()

{

this.MyConString = "server=" + this.server + ";User Id=" + this.user + ";password=" + this.password + ";Persist Security Info=True;database=" + this.database + ";";

try

{

this.connection = new MySqlConnection(this.MyConString);

}

catch (MySql.Data.MySqlClient.MySqlException ex)

{

switch (ex.Number)

{

case 0:

MessageBox.Show("Cannot connect to server. Contact administrator");

break;

case 1045:

MessageBox.Show("Invalid username/password when connecting to the database, please contact with an administrator.");

break;

default:

MessageBox.Show(ex.ToString());

break;

}

}

}

/\*\*

\* @desc Returns the connection to the database, if it is closed, it

\* opens it.

\*

\* @return [MySqlConnection]

\* The mysql connection object returned

\*

\*/

public MySqlConnection mycGetConnection()

{

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

return this.connection;

}

/\*\*

\* @desc Execute an arbitrary query string against the active database, very similar

\* to the PHP way. Do not use this function for INSERT, UPDATE, or DELETE queries.

\*

\* @param string query

\* The prepared statement query to run.

\*

\* @return [List<Hashtable>]

\* A List<HashTable> containing the data retrieved from the databse. This

\* date can be used with a foreach loop or directly htResultset[index][fieldName]

\*

\*/

public List<Hashtable> lhSqlQuery(string query)

{

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

// Create and populate a List.

List<Hashtable> resultset = new List<Hashtable>();

try

{

MySqlCommand command = this.connection.CreateCommand();

MySqlDataReader Reader;

command.CommandText = query;

Reader = command.ExecuteReader();

Hashtable resultset\_tmp;

int i;

while (Reader.Read())

{

resultset\_tmp = new Hashtable();

for (i = 0; i < Reader.FieldCount; i++)

{

// We check first if the key exists

if(!resultset\_tmp.ContainsKey(Reader.GetName(i).ToString()))

resultset\_tmp.Add(Reader.GetName(i).ToString(), Reader.GetValue(i).ToString());

}

resultset.Add(resultset\_tmp);

}

Reader.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

return resultset;

}

/\*\*

\* @desc This function performs an insert statement on the database

\* and returns the id of the the last record inserted

\* @params [string] query

\* @return [int] The id of the last record inserted

\* or the update

\*/

public int InsertToDB(string query)

{

string lastInsertId = "-1";

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

try

{

MySqlCommand command = this.connection.CreateCommand();

command.CommandText = query;

command.ExecuteNonQuery();

List<Hashtable> result = this.lhSqlQuery("SELECT LAST\_INSERT\_ID() id");

lastInsertId = result[0]["id"].ToString();

}

catch (Exception ex)

{

// We don't want to show that to the user

MessageBox.Show(ex.ToString());

}

this.connection.Close();

return int.Parse(lastInsertId);

}

/\*\*

\* @desc This function executes the specified query on the database and

\* returns a DataTable ready to use in a DataGridView component.

\* @params [string] query

\* @return [DataTable] The DataTable object if there is no data, the object

\* will be returned empty.

\*/

public DataTable dtGetTableForDataGrid(string query)

{

DataTable table = new DataTable();

MySqlDataAdapter MyDA = new MySqlDataAdapter();

try

{

MyDA.SelectCommand = new MySqlCommand(query, this.mycGetConnection());

MyDA.Fill(table);

}

catch (MySqlException e)

{

MessageBox.Show(e.ToString());

return new DataTable();

}

return table;

}

/\*\*

\* @desc This function Returns the datasource ready to be used in a combobox.

\* @params [string] query

\* @return [ArrayList] ArrayList containing the items

\*/

public ArrayList alGetComboFromDB(string query, string fieldId, string fieldValue)

{

ArrayList alMyItems = new ArrayList();

List<Hashtable> result = this.lhSqlQuery(query);

foreach (Hashtable record in result)

alMyItems.Add(new DictionaryEntry(record[fieldId].ToString(), record[fieldValue].ToString()));

return alMyItems;

}

/\*\*

\* @desc This function returns the id of the the modified

\* records affected by a Delete or Update sql statement

\* @params [string] query

\* @return [int] The number of modified rows by the delete

\* or the update

\*/

public int DeleteOrUpdate(string query)

{

int affectedRows = 0;

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

try

{

MySqlCommand command = this.connection.CreateCommand();

command.CommandText = query;

affectedRows = command.ExecuteNonQuery();

}

catch (MySql.Data.MySqlClient.MySqlException ex)

{

// We don't want to show that to the user

switch (ex.Number)

{

case 1451:

return -1;

default:

MessageBox.Show(ex.ToString());

break;

}

}

this.connection.Close();

return affectedRows;

}

/\*\*

\* @desc This function uploads a file to the database

\* and returns the id of the the last record inserted

\* @params [string] filePath: The path on the hard drive to the file to be inserted

\* @params [string] fileName: The inserted file's name

\* @return [int] The id of the last record inserted

\* or the update

\*/

public string uploadFileToDB(string filePath, string fileName)

{

// Create a mysql command object

MySql.Data.MySqlClient.MySqlCommand cmd;

cmd = new MySql.Data.MySqlClient.MySqlCommand();

int fileSize;

byte[] rawData;

FileStream fs;

string id\_file;

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

try

{

// Create a filestream object for the file

fs = new FileStream(filePath, FileMode.Open, FileAccess.Read);

// Retrieve the filesize

fileSize = int.Parse(fs.Length.ToString());

// Create a byte[] object for the raw data

rawData = new byte[fileSize];

// Copy the binary contents of the file into the byte[] object

fs.Read(rawData, 0, fileSize);

// Close the filestream

fs.Close();

// Add mysql connection to mysql command object

cmd.Connection = connection;

// Add insert file query with parameters to mysql command object

cmd.CommandText = "INSERT INTO file VALUES(NULL, @FileName, @FileSize, @File)";

// Add parameter values to mysql command object

cmd.Parameters.AddWithValue("@FileName", fileName);

cmd.Parameters.AddWithValue("@FileSize", fileSize);

cmd.Parameters.AddWithValue("@File", rawData);

// Launch query of mysql command object

cmd.ExecuteNonQuery();

// Create a mysql datareader object for holding results of query

MySqlDataReader Reader;

// Create query for checking insert query result

cmd.CommandText = "SELECT LAST\_INSERT\_ID() id";

// Launch query for checking insert query result

Reader = cmd.ExecuteReader();

// Read the reader

Reader.Read();

// Retrieve reader result

id\_file = Reader.GetValue(0).ToString();

MessageBox.Show("File Inserted into database successfully!", "Success!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

// Close the connection

connection.Close();

// Return last inserted file's ID

return id\_file;

}

// Error catching

catch (MySql.Data.MySqlClient.MySqlException ex)

{

MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Message,

"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return "-1";

}

}

/\*\*

\* @desc This function downloads an image from the database

\* @params [string] id\_file: the file ID in the database that has to be downloaded

\* @return [Image] an image ready to be loaded into a pictureBox

\*/

public Image loadImageFromDB(string id\_file)

{

// Create a mysql command object

MySql.Data.MySqlClient.MySqlCommand cmd;

cmd = new MySql.Data.MySqlClient.MySqlCommand();

// Create a mysql datareader object

MySqlDataReader Reader;

int FileSize;

byte[] rawData;

// The connection is forced when its not connected

if (this.connection.State.ToString() == "Closed")

{

this.connect();

connection.Open();

}

try

{

cmd.Connection = connection;

// Add query to find and download the image to mysql command object

cmd.CommandText = "SELECT file\_name, file\_size, file FROM file WHERE id\_file = '" + id\_file + "' ";

// Launch query

Reader = cmd.ExecuteReader();

// If there is 0 result

if (!Reader.HasRows)

throw new Exception("There are no BLOBs to save");

// Read the reader

Reader.Read();

// retrive the file size from reader data

FileSize = int.Parse(Reader.GetUInt32(Reader.GetOrdinal("file\_size")).ToString());

// Create a new byte[] object foe holding binary image data

rawData = new byte[FileSize];

// Add image data to the byte[] object from reader

Reader.GetBytes(Reader.GetOrdinal("file"), 0, rawData, 0, FileSize);

// Create memory stream that can be read into Image object as source

MemoryStream picStream = new MemoryStream(rawData);

// Close reader

Reader.Close();

// Close connection

connection.Close();

// Return the ready image from memory stream

return Image.FromStream(picStream);

}

// Catch the errors

catch (MySql.Data.MySqlClient.MySqlException ex)

{

MessageBox.Show("Error " + ex.Number + " has occurred: " + ex.Message,

"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

return null;

}

}

}

}

## Example of custom message box (frm\_message\_box.cs)

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Gym\_administration

{

/\*\*

\* @desc

\* Form Handler for custom dialogs

\* @params [none] Incoming parameters are described at the individual constructors.

\* @return [none] No directly returned data.

\* Returns of public methods are described at the individual methods.

\*/

public partial class frm\_message\_box : Form

{

frm\_message\_box newMessageBox;

static string Button\_id;

string Button1\_Value;

string Button2\_Value;

string Button3\_Value;

int Counter1\_PrevValue;

/\*\*

\* @desc Constructor.

\* Initializes the dialog to its default state

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public frm\_message\_box()

{

InitializeComponent();

label\_counter1.Hide();

counter1.Hide();

label\_counter2.Hide();

label\_borrowedcounter.Hide();

button\_1.Hide();

button\_2.Hide();

button\_3.Hide();

}

/\*\*

\* @desc Constants for default button values

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public class MBButton

{

public const UInt32 MB\_OK = 0x00000000;

public const UInt32 MB\_OKCANCEL = 0x00000001;

public const UInt32 MB\_ABORTRETRYIGNORE = 0x00000002;

public const UInt32 MB\_YESNOCANCEL = 0x00000003;

public const UInt32 MB\_YESNO = 0x00000004;

public const UInt32 MB\_RETRYCANCEL = 0x00000005;

public const UInt32 MB\_CANCELTRYCONTINUE = 0x00000006;

public const UInt32 MB\_CUST1 = 0x00000007;

public const UInt32 MB\_CUST2 = 0x00000008;

public const UInt32 MB\_CUST3 = 0x00000009;

public const UInt32 MB\_CUST4 = 0x00000010;

public const UInt32 MB\_HELP = 0x00004000;

}

/\*\*

\* @desc Default return values

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

public enum MBReturn

{

OK = 1,

CANCEL = 2,

ABORT = 3,

RETRY = 4,

IGNORE = 5,

YES = 6,

NO = 7,

CLOSE = 8,

HELP = 9,

TRYAGAIN = 10,

CONTINUE = 11,

}

/\*\*

\* @desc It sets up the default and custom dialog layout variations

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void SetButtonLayout(UInt32 Type, string Button1, string Button2, string Button3)

{

switch (Type)

{

case MBButton.MB\_OK:

{

button\_3.Show();

button\_3.Text = "Ok";

Button3\_Value = MBReturn.OK.ToString();

break;

}

case MBButton.MB\_OKCANCEL:

{

button\_2.Show();

button\_2.Text = "OK";

button\_3.Show();

button\_3.Text = "Cancel";

Button2\_Value = MBReturn.OK.ToString();

Button3\_Value = MBReturn.CANCEL.ToString();

break;

}

case MBButton.MB\_ABORTRETRYIGNORE:

{

button\_1.Show();

button\_1.Text = "Abort";

button\_2.Show();

button\_2.Text = "Retry";

button\_3.Show();

button\_3.Text = "Ignore";

Button1\_Value = MBReturn.ABORT.ToString();

Button2\_Value = MBReturn.RETRY.ToString();

Button3\_Value = MBReturn.IGNORE.ToString();

break;

}

case MBButton.MB\_YESNOCANCEL:

{

button\_1.Show();

button\_1.Text = "Yes";

button\_2.Show();

button\_2.Text = "No";

button\_3.Show();

button\_3.Text = "Cancel";

Button1\_Value = MBReturn.YES.ToString();

Button2\_Value = MBReturn.NO.ToString();

Button3\_Value = MBReturn.CANCEL.ToString();

break;

}

case MBButton.MB\_YESNO:

{

button\_2.Show();

button\_2.Text = "Yes";

button\_3.Show();

button\_3.Text = "No";

Button2\_Value = MBReturn.YES.ToString();

Button3\_Value = MBReturn.NO.ToString();

break;

}

case MBButton.MB\_RETRYCANCEL:

{

button\_2.Show();

button\_2.Text = "Retry";

button\_3.Show();

button\_3.Text = "Cancel";

Button2\_Value = MBReturn.RETRY.ToString();

Button3\_Value = MBReturn.CANCEL.ToString();

break;

}

case MBButton.MB\_CANCELTRYCONTINUE:

{

button\_1.Show();

button\_1.Text = "Cancel";

button\_2.Show();

button\_2.Text = "Try";

button\_3.Show();

button\_3.Text = "Continue";

Button1\_Value = MBReturn.CANCEL.ToString();

Button2\_Value = MBReturn.TRYAGAIN.ToString();

Button3\_Value = MBReturn.CONTINUE.ToString();

break;

}

case MBButton.MB\_HELP:

{

button\_3.Show();

button\_3.Text = "Help";

Button3\_Value = MBReturn.HELP.ToString();

break;

}

case MBButton.MB\_CUST1:

{

button\_3.Show();

button\_3.Text = Button1;

Button3\_Value = Button1;

break;

}

case MBButton.MB\_CUST2:

{

button\_2.Show();

button\_2.Text = Button1;

button\_3.Show();

button\_3.Text = Button2;

Button2\_Value = Button1;

Button3\_Value = Button2;

break;

}

case MBButton.MB\_CUST3:

{

button\_1.Show();

button\_1.Text = Button1;

button\_2.Show();

button\_2.Text = Button2;

button\_3.Show();

button\_3.Text = Button3;

Button1\_Value = Button1;

Button2\_Value = Button2;

Button3\_Value = Button3;

break;

}

case MBButton.MB\_CUST4:

{

label\_counter1.Show();

counter1.Value = 1;

Counter1\_PrevValue = 1;

counter1.Show();

counter1.Minimum = 1;

counter1.Maximum = int.Parse(Button1);

label\_counter2.Show();

label\_borrowedcounter.Show();

label\_borrowedcounter.Text = (int.Parse(Button1)-1).ToString();

button\_2.Show();

button\_2.Text = "OK";

button\_3.Show();

button\_3.Text = "Cancel";

Button2\_Value = MBButton.MB\_CUST4.ToString();

Button3\_Value = MBReturn.CANCEL.ToString();

break;

}

}

}

/\*\*

\* @desc Default dialog invoker

\* @params [Uint32] Type: Selector of one of the default dialog layouts

\* @params [string] txtMessage: Dialog message

\* @params [string] txtTitle: Dialog title

\* @return [string] Button\_id: Return value of selected button

\*/

public string ShowBox(UInt32 Type, string txtMessage, string txtTitle)

{

// Create message box

newMessageBox = new frm\_message\_box();

// Set layout

newMessageBox.SetButtonLayout(Type, "", "", "");

// Set title

if (txtTitle == "")

newMessageBox.Text = "Message";

else

newMessageBox.Text = txtTitle;

// Set message

newMessageBox.label\_Message.Text = txtMessage;

// Show dialog

newMessageBox.ShowDialog();

// Return result

return Button\_id;

}

/\*\*

\* @desc Dialog invoker with one custom button

\* @params [Uint32] Type: Selector of one of the custom dialog layouts

\* @params [string] txtMessage: Dialog message

\* @params [string] txtTitle: Dialog title

\* @params [string] Button1: Button text

\* @return [string] Button\_id: Return value of selected button

\*/

public string ShowBox(UInt32 Type, string txtMessage, string txtTitle, string Button1)

{

newMessageBox = new frm\_message\_box();

newMessageBox.SetButtonLayout(Type, Button1, "", "");

if (txtTitle == "") newMessageBox.Text = "Message";

else newMessageBox.Text = txtTitle;

newMessageBox.label\_Message.Text = txtMessage;

newMessageBox.ShowDialog();

return Button\_id;

}

/\*\*

\* @desc Dialog invoker with two custom buttons

\* @params [Uint32] Type: Selector of one of the custom dialog layouts

\* @params [string] txtMessage: Dialog message

\* @params [string] txtTitle: Dialog title

\* @params [string] Button1: Button text

\* @params [string] Button2: Button text

\* @return [string] Button\_id: Return value of selected button

\*/

public string ShowBox(UInt32 Type, string txtMessage, string txtTitle, string Button1, string Button2)

{

newMessageBox = new frm\_message\_box();

newMessageBox.SetButtonLayout(Type, Button1, Button2, "");

if (txtTitle == "") newMessageBox.Text = "Message";

else newMessageBox.Text = txtTitle;

newMessageBox.label\_Message.Text = txtMessage;

newMessageBox.ShowDialog();

return Button\_id;

}

/\*\*

\* @desc Dialog invoker with three custom buttons

\* @params [Uint32] Type: Selector of one of the custom dialog layouts

\* @params [string] txtMessage: Dialog message

\* @params [string] txtTitle: Dialog title

\* @params [string] Button1: Button text

\* @params [string] Button2: Button text

\* @params [string] Button3: Button text

\* @return [string] Button\_id: Return value of selected button

\*/

public string ShowBox(UInt32 Type, string txtMessage, string txtTitle, string Button1, string Button2, string Button3)

{

newMessageBox = new frm\_message\_box();

newMessageBox.SetButtonLayout(Type, Button1, Button2, Button3);

if (txtTitle == "") newMessageBox.Text = "Message";

else newMessageBox.Text = txtTitle;

newMessageBox.label\_Message.Text = txtMessage;

newMessageBox.ShowDialog();

return Button\_id;

}

/\*\*

\* @desc Executes when button 1 is clicked on

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_1\_Click(object sender, EventArgs e)

{

// Pass the result to return method

Button\_id = Button1\_Value;

this.Close();

}

/\*\*

\* @desc Executes when button 1 is clicked on

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_2\_Click(object sender, EventArgs e)

{

// If this is an item return dialog, then the result is to be the amount returned

if (Button2\_Value == MBButton.MB\_CUST4.ToString())

Button2\_Value = label\_borrowedcounter.Text;

// Pass the result to return method

Button\_id = Button2\_Value;

this.Close();

}

/\*\*

\* @desc Executes when button 1 is clicked on

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void button\_3\_Click(object sender, EventArgs e)

{

// Pass the result to return method

Button\_id = Button3\_Value;

this.Close();

}

/\*\*

\* @desc Executes when counter1's value is changed

\* @params [none] No input parameter.

\* @return [none] No directly returned data.

\*/

private void counter1\_ValueChanged(object sender, EventArgs e)

{

// if counter1's value is decreased (-1)

// (prevvalue is first 1, counter1 is also 1,

// so if first change is decrease counter1 will be 0)

if (Counter1\_PrevValue > counter1.Value)

{

// if opposite counter is not at max yet

if (counter1.Value < counter1.Maximum)

{

int iCounter2;

// Retrieve current value of opposite counter

iCounter2 = int.Parse(label\_borrowedcounter.Text);

// Increase opposite counter's value

iCounter2++;

// Set opposite counter to the new value

label\_borrowedcounter.Text = iCounter2.ToString();

// Store current value of counter1 as prevvalue

Counter1\_PrevValue = int.Parse(counter1.Value.ToString());

}

}

// if counter1's value is increased (+1)

// (prevvalue is first 1, counter1 is also 1,

// so if first change is increase counter1 will be 2)

else if (Counter1\_PrevValue < counter1.Value)

{

// if opposite counter is not at min yet

if (counter1.Value > counter1.Minimum)

{

int iCounter2;

// Retrieve current value of opposite counter

iCounter2 = int.Parse(label\_borrowedcounter.Text);

// Decrease opposite counter's value

iCounter2--;

// Set opposite counter to the new value

label\_borrowedcounter.Text = iCounter2.ToString();

// Store current value of counter1 as prevvalue

Counter1\_PrevValue = int.Parse(counter1.Value.ToString());

}

}

}

}

}

## Example of custom utils class (utils.cs)

using System;

using System.Security.Cryptography;

using System.Text;

using System.Windows.Forms;

using System.Globalization;

using System.Net.Mail;

namespace Gym\_administration

{

static class Utils

{

public const UInt32 MB\_OK = 0x00000000;

public const UInt32 MB\_OKCANCEL = 0x00000001;

public const UInt32 MB\_ABORTRETRYIGNORE = 0x00000002;

public const UInt32 MB\_YESNOCANCEL = 0x00000003;

public const UInt32 MB\_YESNO = 0x00000004;

public const UInt32 MB\_RETRYCANCEL = 0x00000005;

public const UInt32 MB\_CANCELTRYCONTINUE = 0x00000006;

public const UInt32 MB\_CUST1 = 0x00000007;

public const UInt32 MB\_CUST2 = 0x00000008;

public const UInt32 MB\_CUST3 = 0x00000009;

public const UInt32 MB\_CUST4 = 0x00000010;

public const UInt32 MB\_HELP = 0x00004000;

/\*\*

\* @desc Returns true when the email received is valid

\* @params [string] sEmail

\* @return [bool] True when the e-mail is a valid e-mail address

\* otherwise it will return false

\*/

static public bool bValidateEmail(string sEmail)

{

try

{

MailAddress temp = new System.Net.Mail.MailAddress(sEmail);

}

catch (FormatException)

{

return false;

}

catch (ArgumentException)

{

return false;

}

return true;

}

/\*\*

\* @desc It creates and md5hash useful to generate hashes for the password

\* to be stored in the database

\* @params [string] input

\* @return [string] The md5 string hash

\*/

static public string CreateMD5Hash(string input)

{

// Use input string to calculate MD5 hash

MD5 md5 = System.Security.Cryptography.MD5.Create();

byte[] inputBytes = System.Text.Encoding.ASCII.GetBytes(input);

byte[] hashBytes = md5.ComputeHash(inputBytes);

// Convert the byte array to hexadecimal string

StringBuilder sb = new StringBuilder();

for (int i = 0; i < hashBytes.Length; i++)

{

sb.Append(hashBytes[i].ToString("X2"));

// To force the hex string to lower-case letters instead of

// upper-case, use he following line instead:

// sb.Append(hashBytes[i].ToString("x2"));

}

return sb.ToString().ToLower();

}

/\*\*

\* @desc It generates the a new random member number

\* @return [string] The new member number

\*/

static public string sGenerateNewMemberNumber()

{

// TODO: Check for existing number

//Create a new Random class

Random RandomClass = new Random();

string sRandomNumber1 = RandomClass.Next(100, 999).ToString();

string sRandomNumber2 = RandomClass.Next(100, 999).ToString();

string sRandomNumber3 = RandomClass.Next(100, 999).ToString();

return sRandomNumber1 + " " + sRandomNumber2 + " " + sRandomNumber3;

}

/\*\*

\* @desc It returns a valid formated date [YYYY-MM-DD] to be stored in the

\* database from a U.K format [DD/MM/YYYY]

\* @params [string] date in [DD/MM/YYYY] format

\* @return [string] date in [YYYY-MM-DD] format

\*/

static public string sGetMysqlDate(string date)

{

try

{

// fetch the en-GB culture

CultureInfo ukCulture = new CultureInfo("en-GB");

// pass the DateTimeFormat information to DateTime.Parse

DateTime dateTime = DateTime.Parse(date, ukCulture.DateTimeFormat);

return String.Format("{0:yyyy-MM-dd}", dateTime);

}

catch (FormatException)

{

return "0000-00-00";

}

}

/\*\*

\* @desc It validates a string to see if it's in a time format

\* @params [string] time in [HH:MM] format

\* @return [bool] only true if the time is correct

\* \*/

static public bool bValidateTime(string sTime)

{

try

{

// fetch the en-GB culture

CultureInfo ukCulture = new CultureInfo("en-GB");

// pass the DateTimeFormat information to DateTime.Parse

DateTime dateTime = DateTime.Parse(sTime, ukCulture.DateTimeFormat);

string sFormat = String.Format("{0:HH:mm}", dateTime);

return true;

}

catch (FormatException)

{

return false;

}

}

/\*\*

\* @desc It returns a valid formated date [DD/MM/YYYY] to be used in the c sharp forms

\* from a mysql formated date [YYYY-MM-DD]

\* @params [string] date in [YYYY-MM-DD] format

\* @return [string] date in [DD/MM/YYYY] format

\*/

static public string sGetCsharpDateFromMysqlDate(string date)

{

try

{

// fetch the en-US culture

CultureInfo usCulture = new CultureInfo("en-US");

// pass the DateTimeFormat information to DateTime.Parse

DateTime dateTime = DateTime.Parse(date, usCulture.DateTimeFormat);

return String.Format("{0:dd/MM/yyyy}", dateTime);

}

catch (FormatException)

{

return "00/00/0000";

}

}

/\*\*

\* @desc It checks if the form is already opened, it it is, it shows it on top

\* @params [Form] The form to check

\* @return [bool] True if it is opened

\*/

static public bool bIsAlreadyOpened(Form frmMyForm)

{

foreach (Form frmOpenForm in Application.OpenForms)

{

if (frmOpenForm.GetType() == frmMyForm.GetType())

{

frmOpenForm.TopMost = true;

frmOpenForm.Visible = true;

frmOpenForm.Activate();

return true;

}

}

return false;

}

}

# Testing Methods

Software testing can obtain the quality of a product or service; it allows an independent view of the software and helps the developer understand the risks of implementing the software. The basic concept of software testing is the process of validating and verifying software of a program or application. Tests can check that the product meets the business and technical requirements of a company as shown by the original design. Software testing can take place at any time in the development process; however the most comprehensive testing takes place once all the coding has been completed. The gym system has been tested using Black Box, White Box and Usability Testing; the results are displayed in the following pages.

## Black Box Testing

Black Box Testing can be simply seen as the opposite of White Box Testing, the software is treated as a black box without any knowledge of the internal implementation of the software. This means that the user can use the software without knowing the limitations which can really push the performance of the software. For the first test a novice user will be asked to carry out two simple tasks that are both possible to see if they find problems that are not apparent to a user that has built the software.

Task 1: Add a member then book that member on a class.

Task 2: Add a class and a set of equipment to that class.

User Results

These are the observations of the user completing the tasks that were sent.

Task1: The user found it easy to login in and find the correct form very quickly, the new member was added within 2 minutes and the user didn’t have any queries on how to use the system. The user then started the second part of the task by looking for the class list, they opened a few forms before finding the correct form but once they did the double clicked the class they wished to enrol the member on and then checked that classes attendance list to check that it had worked. The user was able to do this without asking for any assistance; however it would be more efficient for the client if they were presented with a simple training manual so that if there were queries they could be simply resolved. The user is computer literate but has not used software like this before.

Task 2: This task involves the management functions of the software, the user logged in again and looked through the staff functions before then clicking on the management functions. Once the user opened the menu the user swiftly picked the correct ‘add class’ option, they then filled in the information and clicked the ‘equipment booking’ button to add a set of racket. When the user clicked save there was an error displayed as they had entered an incorrect ‘end time’. Once they closed the error and rectified the mistake and clicked save again. The user then decided to view the class list to view the information that they had added.

The user highlighted that the error message was precise allowing them to change the information that the page was rejecting. The user also had their own suggestion of adding messages incorporated to each window, for example if they could roll over an icon next to a button or field and information appear on the screen.

This kind of testing can be invaluable when trying to determine whether the software is ready to hand over to the client. An inexperienced user can give a very different perspective on the software and their suggestions can help improve the usability of the software.

## White Box Testing

This method of testing is completed by someone with the knowledge of the internal coding of the software. Test cases are then designed based on the internal structure as shown in the test cases on the following pages. The extensive testing of the software can determine whether or not the software meets the client’s requirements.

White Box -Test Cases

### Login

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Test username | ‘admin@ad.min’ | Combine with the correct password (“admin”) the system will log in. | Pass | 25/04/10 |
| 2 | Test username | ‘kmsmith’ | Log in will fail and an error message will be displayed. | Pass | 25/04/10 |
| 3 | Test password | ‘admin’ | Combined with the correct user name (“admin@ad.min”) the system will log in. | Pass | 25/04/10 |
| 4 | Test password | ‘jlk456’ | Log in will fail and an error message will be displayed. | Pass | 25/04/10 |
| 5 | Cancel button | Press ‘Cancel’ | Login window will close. | Pass | 25/04/10 |
| 6 | Accept button | Press ‘Accept’ | With correct data the login window will close and the menu will be displayed. | Pass | 25/04/10 |
| 7 | Accept button | Press ‘Accept’ | With incorrect data the login window will stay on display and an error message will be displayed. | Pass | 25/04/10 |

### Main Menu

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Add New Member | Click Add New Member | A Blank Member form should be displayed to the user. |  | 24/04/10 |
| 2 | Edit Member | Click Edit Member | The member list should be displayed to the user so that they can search for a member they wish to edit. |  | 24/04/10 |
| 3 | Arrange Class | Click Arrange Class | The Class Arrange window will be displayed to that the used can fill in the blank booking form. |  | 24/04/10 |
| 4 | Edit Arranged Class | Click Edit Arranged Class | The Class Arrangements List will be displayed so that the users can the select a class they wish to edit. |  | 24/04/10 |
| 5 | Edit Equip Bookings | Click Edit Equip Bookings | The Equipment Booking List window is then displayed so that the users can pick a equipment booking that they wish to edit. |  | 24/04/10 |
| 6 | Edit Room List | Click Edit Room List | The room list is displayed so that the user can choose a room they wish to edit. |  | 24/04/10 |
| 7 | Management Functions | Click Management Functions | This expands the Management Functions Menu and shrinking the other menu that was displaying. |  | 24/04/10 |
| 8 | Add New Staff | Click Add New Staff | The Gym Staff Form window will open displaying a blank form that the user can fill out and save the information of a new member of staff. |  | 24/04/10 |
| 9 | Edit Staff | Click Edit Staff | The Staff List window will be displayed so that users can select members of staff that they wish to edit. |  | 24/04/10 |
| 10 | Membership Fees | Click Membership Fees | The Payments window will then open to show all the payments that have taken place, a new payment could then be added to the list. |  | 24/04/10 |
| 11 | Edit Arranged Class | Click Edit Arranged Class | The Class Arrangements List will then be displayed with the list of arranged classes, this classes can then be double clicked to open the class form where the information can be edited. |  | 24/04/10 |
| 12 | Add New Class | Click Add New Class | The Class Form window opens displaying a blank class information form. |  | 24/04/10 |
| 13 | Edit Class | Click Edit Class | The Class List window opens displaying the class list, the entries can then be double clicked and the information can be edited. |  | 24/04/10 |
| 14 | Add New Equipment | Click Add New Equipment | The Equipment Form window opens displaying a blank equipments form that can be filled in and saved to the equipment list. |  | 24/04/10 |
| 15 | Edit Equipment | Click Edit Equipment | The Equipment List window opens displaying the list of available equipments, each item can be doubled clicked the equipment form for that piece of equipment opens allowing users to edit this information and then save the changes. |  | 24/04/10 |
| 16 | Edit Room List | Click Edit Room List | The Room List window opens displaying the list of rooms than can then be double clicked to open the Room Form for that particular room containing the information about the room, the information can then be edited. The Room List also contains a ‘Add Room’ button. |  | 24/04/10 |
| 17 | Login Options | Click Login Options | This expands the Login Options Menu and shrinking the other menu that was displaying. |  | 24/04/10 |
| 18 | Change Password | Click Change Password | This opens the Password Update window that allows the user to change their password. |  | 24/04/10 |
| 19 | Staff Functions | Click Staff Functions | This expands the Staff Functions Menu and shrinking the other menu that was displaying. |  | 24/04/10 |

### Staff Functions

#### ‘Add New Member’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Member number | add several members | Check that a new member number is checked each time. | Pass | 25/04/10 |
| 2 | Member number | Try editing the number | The field will not be able to be edited. | Pass | 25/04/10 |
| 3 | Date of Birth | ‘19/06/1989’ | Date will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 4 | Date of Birth | ‘14/04/87’ | Data will be rejected as it is the incorrect format, an error will be displayed. | FAIL | 25/04/10 |
| 5 | Date of Birth | ‘14/j4/5k6’ | Only integers should be able to be typed into this field. | Pass | 25/04/10 |
| 6 | Date of Birth | ‘12/56/1987’ | Data will be rejected as it is the incorrect format, an error will be displayed. | Pass | 25/04/10 |
| 7 | Postcode | ‘CB8 1HH’ | Data will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 8 | Postcode | ‘GL20 8HL’ | Data will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 9 | Postcode | ‘145 GHJ’ | Field is formatted to not let integers be typed in as the first character. | Pass | 25/04/10 |
| 10 | Telephone | ‘01223 456789’ | Data will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 11 | Telephone | ‘01684 KL456’ | Data will not be save, an error will be displayed to the user when they click save. | FAIL | 25/04/10 |
| 12 | Email | ‘louiseplant@plant.com’ | Data will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 13 | Email | ‘Louiseplant’ | Unless the information in the field contains ‘@’ and ‘.com’ it will produce an error when the user clicks save. | Pass | 25/04/10 |
| 14 | Medical Notes | Edit information | Data will be accepted and the user information added to the database. | Pass | 25/04/10 |
| 15 | Medical Notes  (Restore Button) | Click to restore data | All information in the box will be replaced with the original default. | Pass | 25/04/10 |
| 16 | Save & Open | Correct data | If all data is correct then window will appear notifying the user that the member has been added successfully. The member list will then be displayed. | Pass | 25/04/10 |
| 17 | Save & Open | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |
| 18 | Save & Close | Correct data | If all data is correct then window will appear notifying the user that the member has been added successfully. The page will then close. | Pass | 25/04/10 |
| 19 | Save & Close | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |
| 20 | Save & Stay | Correct data | If all data is correct then window will appear notifying the user that the member information has been saved successfully. It will then stay on that Members Information page. | Pass | 25/04/10 |
| 21 | Save & Stay | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |

#### ‘Edit Member’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Membership Type | ‘Student | Display all the members that have the Membership Type ‘Student’. | FAIL | 25/04/10 |
| 2 | First Name | ‘Louise’ | Displays the members with the first name ‘Louise’ i.e Member Number 37. | Pass | 25/04/10 |
| 3 | First Name | ‘John’ | No members are displayed as there are no users with the first name ‘John’. | Pass | 25/04/10 |
| 4 | Last Name | ‘Plant’ | Displays all the members the last name ‘Plant’ i.e Member Number 37. | Pass | 25/04/10 |
| 5 | Last Name | ‘Smith’ | No members are displayed as there are no users with the last name ‘Smith’. | Pass | 25/04/10 |
| 6 | Date of Birth | ‘12/07/1990’ | Displays all the members the date of birth ‘12/07/1990’. i.e Member Number 17. | Pass | 25/04/10 |
| 7 | Date of Birth | ‘01/05/1989’ | No members are displayed as there are no users with the date of birth ‘01/05/1989’. | Pass | 25/04/10 |
| 8 | Email | ‘louiseplant@plant.com’ | Displays the members with the email ‘louiseplant@plant.com’ i.e Member Number 37. | Pass | 25/04/10 |
| 9 | Email | ‘john@yahoo.co.uk’ | No members are displayed as there are no users with the email ‘john@yahoo.co.uk’. | Pass | 25/04/10 |
| 10 | Search | Click Search | When the Search is clicked it will run the criteria though the database and return the matching results. | Pass | 25/04/10 |
| 11 | Add New Member | Click Add New Member | The Member Form window will open displaying a blank Member Form. | Pass | 25/04/10 |
| 12 | Member Results | Double Click Result | Member Information Form will open displaying the current member information that can then be edited and the changes saved. | Pass | 25/04/10 |
| 13 | Copy All Selected Cells To Clipboard | Click Copy All Selected Cells To Clipboard | The selected will be saved to the clipboard and can then be pasted in other programs. i.e Microsoft Excel. | Pass | 25/04/10 |

#### ‘Arrange Class’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Date | ‘14/04/2010’ | Data will be accepted and the class information added to the database. | Pass | 25/04/10 |
| 2 | Date | Blank | The validation will produce an error instructor the user of the error within the form. | Pass | 25/04/10 |
| 3 | Start Time | ‘03.00’ | Data will be accepted and the class information added to the database. | Pass | 25/04/10 |
| 4 | Start Time | Blank | The validation will produce an error instructor the user of the error within the form. | Pass | 25/04/10 |
| 5 | End Time | ‘04.00’ | Data will be accepted and the class information added to the database. | Pass | 25/04/10 |
| 6 | End Time | Blank | The validation will produce an error instructor the user of the error within the form. | Pass | 25/04/10 |
| 7 | Save & Open | Click Save & Open | If the information is correct then it will be save to the Class List, the Class List will then be displayed on the screen. | Pass | 25/04/10 |
| 8 | Save & Open | Click Save & Open | If the information is incorrect then the validation will produce errors and display a window to instruct the user what to do. | Pass | 25/04/10 |
| 9 | Save & Stay | Click Save & Stay | If the information is correct then it will be save to the Class List, and the Class Information form will stay displayed on the screen. | Pass | 25/04/10 |
| 10 | Save & Stay | Click Save & Stay | If the information is incorrect then the validation will produce errors and display a window to instruct the user what to do. | Pass | 25/04/10 |
| 11 | Save & Close | Click Save & Close | If the information is correct then it will be save to the Class List, the Class List form will then be closed. | Pass | 25/04/10 |
| 12 | Save & Close | Click Save & Close | If the information is incorrect then the validation will produce errors and display a window to instruct the user what to do. | Pass | 25/04/10 |
| 13 | Equipment Booking | Click Equipment Booking | The Equipment List window will open showing the current equipment bookings. | Pass | 25/04/10 |
| 14 | View Attendants | Click View Attendants | The Members List Form will open showing the current list of Members attending that class. | Pass | 25/04/10 |
| 15 | Enroll Members | Click Enroll Members | The Members List window will then open, Members can then be chosen to add to the class list. | Pass | 25/04/10 |

#### ‘Edit Arranged Class’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Search | ‘Yoga’ | A list of all the Yoga classes will be returned to the window. Any of the results can then be double clicked to edit the class information. | Pass | 25/04/10 |
| 2 | Search | ‘Track’ | No results will be returned as there is no class by the name of ‘Track’. | Pass | 25/04/10 |

#### ‘Edit Equip Bookings’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Edit Booking | Double Click a Booking | A window will be displayed so that equipment can be returned. | Pass | 25/04/10 |
| 2 | Return Equipment | Return a football (Booking Number 16) | Once the booking has been selected and the return window has opened then the user can select how many items they wish to return and the equipment list will be updated. | Pass | 25/04/10 |

#### ‘Edit Room List’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Search | ‘3’ | The information for room number ‘3’ is returned to the window. | Pass | 25/04/10 |
| 2 | Add Room | Click Add Room | A Blank Room form will be displayed so that a user can fill it in and add the new room will be added to the Room List. | Pass | 25/04/10 |
| 3 | Edit Room Information | Double Click Room 3 | The Room Form containing that room’s information will be displayed; it can then be edited and saved. The list will then also be updated. | Pass | 25/04/10 |
| 4 | Remove Room | Click Remove | In the Room Form that opens when double clicking a room on the room list there is a button at the bottom of the window that will remove the room from the list. | Pass | 25/04/10 |

### Managements

#### ‘Add New Staff’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Date of Birth | ‘24/06/1987’ | Date will be accepted and the user information added to the database. | Pass | 26/04/10 |
| 2 | Date of Birth | ‘31/04/87’ | Data will be rejected as it is the incorrect format, an error will be displayed. | FAIL | 26/04/10 |
| 3 | Date of Birth | ‘18j4/546’ | Only integers should be able to be typed into this field. | Pass | 26/04/10 |
| 4 | Date of Birth | ‘12/99/1786’ | Data will be rejected as it is the incorrect format, an error will be displayed. | Pass | 26/04/10 |
| 5 | Postcode | ‘CB8 1HH’ | Data will be accepted and the user information added to the database. | Pass | 26/04/10 |
| 6 | Postcode | ‘GL20 8HL’ | Data will be accepted and the user information added to the database. | Pass | 26/04/10 |
| 7 | Postcode | ‘145 GHJ’ | Field is formatted to not let integers be typed in as the first character. | Pass | 26/04/10 |
| 8 | Telephone | ‘01223 456789’ | Data will be accepted and the user information added to the database. | Pass | 26/04/10 |
| 9 | Telephone | ‘01684 KL456’ | Data will not be save, an error will be displayed to the user when they click save. | FAIL | 26/04/10 |
| 10 | Email | ‘tomshoe@plant.com’ | Data will be accepted and the user information added to the database. | Pass | 26/04/10 |
| 11 | Email | ‘tomshoe’ | Unless the information in the field contains ‘@’ and ‘.com’ it will produce an error when the user clicks save. | Pass | 26/04/10 |
| 12 | Contract Start | Automatic data | When the Gym Staff Form is opened the current date is automatically generated and placed in the Contract Start field. | Pass | 26/04/10 |
| 13 | Contract Start | ‘10/jk/4’ | Field validation will not allow for anything integers to typed into the field | Pass | 26/04/10 |
| 14 | Contract Finish | ‘20/05/2011’ | The data should be accepted and saved to the Staff list within the database. | Pass | 26/04/10 |
| 15 | Contract Finish | ‘14/06/10kl’ | Field validation will not allow for anything integers to typed into the field | Pass | 26/04/10 |
| 16 | Employment Type | Drop Down List | The list contains the correct options and allows the user to select the appropriate choice. | Pass | 26/04/10 |
| 17 | Position | Drop Down List | The list contains the correct options and allows the user to select the appropriate choice. | Pass | 26/04/10 |
| 18 | Remove | Click Remove | This will delete the current Member of staff. | Pass | 26/04/10 |
| 19 | Cancel | Click Cancel | Will close the window saving no changes to the databse. | Pass | 26/04/10 |
| 20 | Save & Open | Correct data | If all data is correct then window will appear notifying the user that the member has been added successfully. The member list will then be displayed. | Pass | 25/04/10 |
| 21 | Save & Open | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |
| 22 | Save & Close | Correct data | If all data is correct then window will appear notifying the user that the member has been added successfully. The page will then close. | Pass | 25/04/10 |
| 23 | Save & Close | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |
| 24 | Save & Stay | Correct data | If all data is correct then window will appear notifying the user that the member information has been saved successfully. It will then stay on that Members Information page. | Pass | 25/04/10 |
| 25 | Save & Stay | Incorrect data | A window will appear highlighting errors within the form that need to be corrected. The Member information member will stay displayed so that the errors can be resolved. | Pass | 25/04/10 |

#### ‘Edit Staff’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | First Name | ‘George’ | Displays the members with the first name ‘George’ i.e. Staff Number 6. | FAIL | 26/04/10 |
| 2 | First Name | ‘Peter’ | No members are displayed as there are no users with the first name ‘Peter’. | FAIL | 26/04/10 |
| 3 | Last Name | ‘Clooney’ | Displays the members with the first name ‘Clooney’ i.e. Staff Number 6. | FAIL | 26/04/10 |
| 4 | Last Name | ‘Beaver’ | No members are displayed as there are no users with the last name ‘Beaver’. | FAIL | 26/04/10 |
| 5 | Date of Birth | ‘01/01/1999’ | Displays all the members the date of birth ‘01/01/1999’. i.e Member Number 6. | FAIL | 26/04/10 |
| 6 | Date of Birth | ‘01/05/1989’ | No members are displayed as there are no users with the date of birth ‘01/05/1989’. | FAIL | 26/04/10 |
| 7 | Email | ‘janimani@a.com’ | Displays the members with the email ‘janimani@a.com’ i.e Member Number 1. | FAIL | 26/04/10 |
| 8 | Email | ‘jane@yahoo.co.uk’ | No members are displayed as there are no users with the email ‘jane@yahoo.co.uk’. | FAIL | 26/04/10 |
| 9 | Search | Click Search | When the Search is clicked it will run the criteria though the database and return the matching results. | FAIL | 26/04/10 |
| 10 | Add Staff Member | Click Add New Member | The Staff Form window will open displaying a blank Staff Form that can be filled in a saved. | Pass | 26/04/10 |
| 11 | Close | Click Close | The current window will close. | Pass | 26/04/10 |

#### ‘Membership Fees’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Add Payments | Click Add Payment | The Member List will open. | Pass | 26/04/10 |
| 2 | Add Payment | Choose Member | Double clicking on a member will open the Add Payment Form, this form can then be filled out to record the payment details which will then be added to the payment list. | Pass | 26/04/10 |

#### ‘Edit Arranged Class’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Search | ‘Yoga’ | Three results will be displayed showing the Yoga classes. | Pass | 26/04/10 |
| 2 | Edit Class | Double Click a Class | The Class Arrange window will open, all the information can then be edited and saved. | Pass | 26/04/10 |
| 3 | Equipment Bookings | Click Equipment Bookings | This will open the Equipment List allowing users to add equipment to the class. | Pass | 26/04/10 |
| 4 | Add Equipment | Click on a item of Equipment | A window will open asking if you wish to borrow the piece of equipment or borrow it. | Pass | 26/04/10 |
| 5 | View Attendants | Click View Attendants | This will open the Members list showing all the members attending that class. | Pass | 26/04/10 |
| 6 | Enroll Members | Click Enroll Members | This will open the Member List Form that contains the list of all Member’s, the members can then be searched for and added to the class. | Pass | 26/04/10 |
| 7 | Enroll Member | Enroll Member on a Class | Once the user double clicks on the member a window will pop up asking whether the user wants to add this Member to the class. | Pass | 26/04/10 |

#### ‘Add New Class’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Add Class | ‘Fitness’, ‘1hour.’ Group | The class will be added to the class list. | Pass | 26/04/10 |
| 2 | Save | Click Save | This information will be saved and the class will appear on the Class List. | Pass | 26/04/10 |

#### ‘Edit Class’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Search | ‘Fitness’ | The Fitness classes are then displayed in the window. | Pass | 26/04/10 |
| 2 | View Class Information | Double click Fitness | The Class Information window will then appear displaying that class information. | Pass | 26/04/10 |
| 3 | Add Class | ‘Dance’ | A blank Class Form will open so that the user can fill it in. | Pass | 26/04/10 |
| 4 | Save | Click Save | The information will then be saved and the class added to the Class List | Pass | 26/04/10 |
| 5 | Cancel | Click Cancel | The window will close. | Pass | 26/04/10 |

#### ‘Add New Equipment’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Radio Buttons | ‘Set’ | The Set Name box and other drop down lists will activate to be able to log the information for the set. | Pass | 26/04/10 |
| 2 | Radio Buttons | ‘Item’ | The Item Name box and Description will then become active allowing the user to save the correct information. | Pass | 26/04/10 |
| 3 | Save | ‘Racket’ | Information will be saved to the database and appear on the Equipment List. | Pass | 26/04/10 |
| 4 | Close | Click Close | Will close the current window. | Pass | 26/04/10 |

#### ‘Edit Equipment’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Radio Buttons | ‘Set’ | The list of Sets will show. | Pass | 26/04/10 |
| 2 | Radio Buttons | ‘Item’ | The list of Items will show. | Pass | 26/04/10 |
| 3 | Add Equipment | Click Add Equipment | The Equipment Form will open allowing the user to add a new item or a new set. | Pass | 26/04/10 |
| 4 | Edit Equipment | Double Click Set ‘Rackets’ | The Equipment Form containing that set information. | Pass | 26/04/10 |
| 5 | Remove | Click Remove | This will delete the current Item/Set. | Pass | 26/04/10 |
| 6 | Save | Change description ‘red’ | Information will be saved to the database and appear on the Equipment List. | Pass | 26/04/10 |
| 7 | Close | Click Close | Will close the current window. | Pass | 26/04/10 |

#### ‘Edit Room List’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Search | ‘RT123’ | The results will display to screen. | Pass | 26/04/10 |
| 2 | Search | ‘Bry013’ | No results will be returned as there is no room with the name Bry013. | Pass | 26/04/10 |
| 4 | Add Room | Click Room Form | A blank room form will open allowing the user to enter the information for a new room. | Pass | 26/04/10 |
| 5 | Save | ‘TennisCourt’ | The room will be added to the Room List. | Pass | 26/04/10 |
| 6 | Remove | Remove RT123 | Double click the room entry then click remove and a window will the pop up asking you to confirm the operation before removing the room from the list and the database. | Pass | 26/04/10 |
| 7 | Cancel | Click Cancel | The current window will close. | Pass | 26/04/10 |

### Login Option

#### ‘Change Password’

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Old Password | blank | An error message will appear instructor the user to enter the correct information. | Pass | 26/04/10 |
| 2 | New Password | blank | An error message will appear instructor the user to enter the correct information. | Pass | 26/04/10 |
| 3 | Repeat Password | blank | An error message will appear instructor the user to enter the correct information. | Pass | 26/04/10 |
| 4 | New Password | ‘beetle’ | If the passwords don’t match an error will display on screen. | Pass | 26/04/10 |
| Repeat Password | ‘red’ |

## Usability Testing

Usability Test Cases

### Visibility of System Status

The system should always keep user informed about what is going on, through appropriate feedback within reasonable time.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 1.1 | Does every display begin with a title or header that describes screen contents? | X O O | 01/May/2010 |
| 1.2 | Is there a consistent icon design scheme and stylistic treatment across the system? | X O O | 01/May/2010 |
| 1.3 | Is a single, selected icon clearly visible when surrounded by unselected icons? | X O O | 01/May/2010 |
| 1.4 | Do menu instructions, prompts, and error messages appear in the same place(s) on each menu? | X O O | 01/May/2010 |
| 1.5 | In multipage data entry screens, is each page labeled to show its relation to others? | X O O | 01/May/2010 |
| 1.6 | If overtype and insert mode are both available, is there a visible indication of which one the user is in? | O O X | 01/May/2010 |
| 1.7 | If pop-up windows are used to display error messages, do they allow the user to see the field in error? | X O O | 01/May/2010 |
| 1.8 | Is there some form of system feedback for every operator action? | X O O | 01/May/2010 |
| 1.9 | After the user completes an action (or group of actions), does the feedback indicate that the next group of actions can be started? | X O O | 01/May/2010 |
| 1.10 | Is there visual feedback in menus or dialog boxes about which choices are selectable? | X O O | 01/May/2010 |
| 1.11 | Is there visual feedback in menus or dialog boxes about which choice the cursor is on now? | X O O | 01/May/2010 |
| 1.12 | If multiple options can be selected in a menu or dialog box, is there visual feedback about which options are already selected? | X O O | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 1.13 | Is there visual feedback when objects are selected or moved? | X O O | 01/May/2010 |
| 1.14 | Is the current status of an icon clearly indicated? | X O O | 01/May/2010 |
| 1.15 | Is there feedback when function keys are pressed? | O O X | 01/May/2010 |
| 1.16 | If there are observable delays (greater than fifteen seconds) in the system’s response time, is the user kept informed of the system's progress? | O O X | 01/May/2010 |
| 1.17 | Are response times appropriate to the task? | X O O | 01/May/2010 |
| 1.18 | Typing, cursor motion, mouse selection: 50-1 50 milliseconds | X O O | 01/May/2010 |
| 1.19 | Simple, frequent tasks: less than 1 second | X O O | 01/May/2010 |
| 1.20 | Common tasks: 2-4 seconds | X O O | 01/May/2010 |
| 1.21 | Complex tasks: 8-12 seconds | X O O | 01/May/2010 |
| 1.22 | Are response times appropriate to the user's cognitive processing? | X O O | 01/May/2010 |
| 1.23 | Continuity of thinking is required and information must be remembered throughout several responses: less than two seconds. | X O O | 01/May/2010 |
| 1.24 | High levels of concentration aren't necessary and remembering information is not required: two to fifteen seconds. | X O O | 01/May/2010 |
| 1.25 | Is the menu-naming terminology consistent with the user's task domain? | X O O | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 1.26 | Does the system provide *visibility:* that is, by looking, can the user tell the state of the system and the alternatives for action? | X O O | 01/May/2010 |
| 1.27 | Do GUI menus make obvious which item has been selected? | X O O | 01/May/2010 |
| 1.28 | Do GUI menus make obvious whether deselection is possible? | O O X | 01/May/2010 |
| 1.29 | If users must navigate between multiple screens, does the system use context labels, menu maps, and place markers as navigational aids? | O O X | 01/May/2010 |

### Match Between System and the Real World

The system should speak the user’s language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 2.1 | Are icons concrete and familiar? | X O O | 01/May/2010 |
| 2.2 | Are menu choices ordered in the most logical way, given the user, the item names, and the task variables? | X O O | 01/May/2010 |
| 2.3 | If there is a natural sequence to menu choices, has it been used? | X O O | 01/May/2010 |
| 2.4 | Do related and interdependent fields appear on the same screen? | X O O | 01/May/2010 |
| 2.5 | If shape is used as a visual cue, does it match cultural conventions? | X O O | 01/May/2010 |
| 2.6 | Do the selected colors correspond to common expectations about color codes? | X O O | 01/May/2010 |
| 2.7 | When prompts imply a necessary action, are the words in the message consistent with that action? | X O O | 01/May/2010 |
| 2.8 | Do keystroke references in prompts match actual key names? | X O O | 01/May/2010 |
| 2.9 | On data entry screens, are tasks described in terminology familiar to users? | X O O | 01/May/2010 |
| 2.10 | Are field-level prompts provided for data entry screens? | X O O | 01/May/2010 |
| 2.11 | For question and answer interfaces, are questions stated in clear, simple language? | X O O | 01/May/2010 |
| 2.12 | Do menu choices fit logically into categories that have readily understood meanings? | X O O | 01/May/2010 |
| 2.13 | Are menu titles parallel grammatically? | X O O | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 2.14 | Does the command language employ user jargon and avoid computer jargon? | X O O | 01/May/2010 |
| 2.15 | Are command names specific rather than general? | X O O | 01/May/2010 |
| 2.16 | Does the command language allow both full names and abbreviations? | X O O | 01/May/2010 |
| 2.17 | Are input data codes meaningful? | X O O | 01/May/2010 |
| 2.18 | Have uncommon letter sequences been avoided whenever possible? | X O O | 01/May/2010 |
| 2.19 | Does the system automatically enter leading or trailing spaces to align decimal points? | O X O | 01/May/2010 |
| 2.20 | Does the system automatically enter a dollar sign and decimal for monetary entries? | O O X | 01/May/2010 |
| 2.21 | Does the system automatically enter commas in numeric values greater than 9999? | O X O | 01/May/2010 |
| 2.22 | Do GUI menus offer activation: that is, make obvious how to say *“now do it"?* | X O O | 01/May/2010 |
| 2.23 | Has the system been designed so that keys with similar names do not perform opposite (and potentially dangerous) actions? | O O X | 01/May/2010 |
| 2.24 | Are function keys labeled clearly and distinctively, even if this means breaking consistency rules? | X O O | 01/May/2010 |

### User Control and Freedom

Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 3.1 | If setting up windows is a low-frequency task, is it particularly easy to remember? | X O O | 01/May/2010 |
| 3.2 | In systems that use overlapping windows, is it easy for users to rearrange windows on the screen? | O O X | 01/May/2010 |
| 3.3 | In systems that use overlapping windows, is it easy for users to switch between windows? | O O X | 01/May/2010 |
| 3.4 | When a user's task is complete, does the system wait for a signal from the user before processing? | X O O | 01/May/2010 |
| 3.5 | Can users type-ahead in a system with many nested menus? | O X O | 01/May/2010 |
| 3.6 | Are users prompted to confirm commands that have drastic, destructive consequences? | O X O | 01/May/2010 |
| 3.7 | Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions? | O X O | 01/May/2010 |
| 3.8 | Can users cancel out of operations in progress? | X O O | 01/May/2010 |
| 3.9 | Are character edits allowed in commands? | O O X | 01/May/2010 |
| 3.10 | Can users reduce data entry time by copying and modifying existing data? | X O O | 01/May/2010 |
| 3.11 | Are character edits allowed in data entry fields? | X O O | 01/May/2010 |
| 3.12 | If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code? | O O X | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 3.13 | If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut? | O X O | 01/May/2010 |
| 3.14 | Are menus broad (many items on a menu) rather than deep (many menu levels)? | X O O | 01/May/2010 |
| 3.15 | If the system has multiple menu levels, is there a mechanism that allows users to go back to previous menus? | X O O | 01/May/2010 |
| 3.16 | If users can go back to a previous menu, can they change their earlier menu choice? | X O O | 01/May/2010 |
| 3.17 | Can users move forward and backward between fields or dialog box options? | X O O | 01/May/2010 |
| 3.18 | If the system has multipage data entry screens, can users move backward and forward among all the pages in the set? | O O X | 01/May/2010 |
| 3.19 | If the system uses a question and answer interface, can users go back to previous questions or skip forward to later questions? | O O X | 01/May/2010 |
| 3.20 | Do function keys that can cause serious consequences have an undo feature? | O O X | 01/May/2010 |
| 3.21 | Can users easily reverse their actions? | O X O | 01/May/2010 |
| 3.22 | If the system allows users to reverse their actions, is there a retracing mechanism to allow for multiple undos? | O O X | 01/May/2010 |
| 3.23 | Can users set their own system, session, file, and screen defaults? | O X O | 01/May/2010 |

### Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 4.1 | Have industry or company formatting standards been followed consistently in all screens within a system? | X O O | 01/May/2010 |
| 4.2 | Has a heavy use of all uppercase letters on a screen been avoided? | X O O | 01/May/2010 |
| 4.3 | Do abbreviations not include punctuation? | X O O | 01/May/2010 |
| 4.4 | Are integers right-justified and real numbers decimal-aligned? | X O O | 01/May/2010 |
| 4.5 | Are icons labeled? | X O O | 01/May/2010 |
| 4.6 | Are there no more than twelve to twenty icon types? | X O O | 01/May/2010 |
| 4.7 | Are there salient visual cues to identify the active window? | X O O | 01/May/2010 |
| 4.8 | Does each window have a title? | X O O | 01/May/2010 |
| 4.9 | Are vertical and horizontal scrolling possible in each window? | X O O | 01/May/2010 |
| 4.10 | Does the menu structure match the task structure? | X O O | 01/May/2010 |
| 4.11 | Have industry or company standards been established for menu design, and are they applied consistently on all menu screens in the system? | X O O | 01/May/2010 |
| 4.12 | Are menu choice lists presented vertically? | O O O | 01/May/2010 |
| 4.13 | If "exit" is a menu choice, does it always appear at the bottom of the list? | O O X | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 4.14 | Are menu titles either centered or left-justified? | O O X | 01/May/2010 |
| 4.15 | Are menu items left-justified, with the item number or mnemonic preceding the name? | O O X | 01/May/2010 |
| 4.16 | Do embedded field-level prompts appear to the right of the field label? | O O X | 01/May/2010 |
| 4.17 | Do on-line instructions appear in a consistent location across screens? | O O X | 01/May/2010 |
| 4.18 | Are field labels and fields distinguished typographically? | X O O | 01/May/2010 |
| 4.19 | Are field labels consistent from one data entry screen to another? | X O O | 01/May/2010 |
| 4.20 | Are fields and labels left-justified for alpha lists and right-justified for numeric lists? | X O O | 01/May/2010 |
| 4.21 | Do field labels appear to the left of single fields and above list fields? | X O O | 01/May/2010 |
| 4.22 | Are attention-getting techniques used with care? | X O O | 01/May/2010 |
| 4.23 | Intensity: two levels only | X O O | 01/May/2010 |
| 4.24 | Size: up to four sizes | X O O | 01/May/2010 |
| 4.25 | Font: up to three | X O O | 01/May/2010 |
| 4.26 | Blink: two to four hertz | O O X | 01/May/2010 |
| 4.27 | Color: up to four (additional colors for occasional use only) | X O O | 01/May/2010 |
| 4.28 | Sound: soft tones for regular positive feedback, harsh for rare critical conditions | O O X | 01/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 4.29 | Are attention-getting techniques used only for exceptional conditions or for time-dependent information? | X O O | 01/May/2010 |
| 4.30 | Are there no more than four to seven colors, and are they far apart along the visible spectrum? | X O O | 01/May/2010 |
| 4.31 | Is a legend provided if color codes are numerous or not obvious in meaning? | O O X | 01/May/2010 |
| 4.32 | Have pairings of high-chroma, spectrally extreme colors been avoided? | X O O | 01/May/2010 |
| 4.33 | Are saturated blues avoided for text or other small, thin line symbols? | X O O | 01/May/2010 |
| 4.34 | Is the most important information placed at the beginning of the prompt? | X O O | 01/May/2010 |
| 4.35 | Are user actions named consistently across all prompts in the system? | X O O | 01/May/2010 |
| 4.36 | Are system objects named consistently across all prompts in the system? | X O O | 01/May/2010 |
| 4.37 | Do field-level prompts provide more information than a restatement of the field name? | X O O | 01/May/2010 |
| 4.38 | For question and answer interfaces, are the valid inputs for a question listed? | X O O | 01/May/2010 |
| 4.39 | Are menu choice names consistent, both within each menu and across the system, in grammatical style and terminology? | X O O | 01/May/2010 |
| 4.40 | Does the structure of menu choice names match their corresponding menu titles? | X O O | 01/May/2010 |
| 4.41 | Are commands used the same way, and do they mean the same thing, in all parts of the system? | X O O | 01/May/2010 |
| 4.42 | Does the command language have a consistent, natural, and mnemonic syntax? | X O O | 01/May/2010 |
| 4.43 | Do abbreviations follow a simple primary rule and, if necessary, a simple secondary rule for abbreviations that otherwise would be duplicates? | X O O | 01/May/2010 |
| 4.44 | Is the secondary rule used only when necessary? | O O O | 01/May/2010 |
| 4.45 | Are abbreviated words all the same length? | O O O | 01/May/2010 |
| 4.46 | Is the structure of a data entry value consistent from screen to screen? | O O O | 01/May/2010 |
| 4.47 | Is the method for moving the cursor to the next or previous field consistent throughout the system? | O O O | 01/May/2010 |
| 4.48 | If the system has multipage data entry screens, do all pages have the same title? | O O O | 01/May/2010 |
| 4.49 | If the system has multipage data entry screens, does each page have a sequential page number? | O O O | 01/May/2010 |
| 4.50 | Does the system follow industry or company standards for function key assignments? | O O O | 01/May/2010 |
| 4.51 | Are high-value, high-chroma colors used to attract attention? | O O O | 01/May/2010 |

### Help Users Recognize, Diagnose, and Recover From Errors

Error messages should be expressed in plain language(NO CODES).

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 5.1 | Is sound used to signal an error? | X O O | 02/May/2010 |
| 5.2 | Are prompts stated constructively, without overt or implied criticism of the user? | X O O | 02/May/2010 |
| 5.3 | Do prompts imply that the user is in control? | X O O | 02/May/2010 |
| 5.4 | Are prompts brief and unambiguous. | X O O | 02/May/2010 |
| 5.5 | Are error messages worded so that the system, not the user, takes the blame? | X O O | 02/May/2010 |
| 5.6 | If humorous error messages are used, are they appropriate and inoffensive to the user population? | O O X | 02/May/2010 |
| 5.7 | Are error messages grammatically correct? | X O O | 02/May/2010 |
| 5.8 | Do error messages avoid the use of exclamation points? | O X O | 02/May/2010 |
| 5.9 | Do error messages avoid the use of violent or hostile words? | X O O | 02/May/2010 |
| 5.10 | Do error messages avoid an anthropomorphic tone? | X O O | 02/May/2010 |
| 5.11 | Do all error messages in the system use consistent grammatical style, form, terminology, and abbreviations? | X O O | 02/May/2010 |
| 5.12 | Do messages place users in control of the system? | X O O | 02/May/2010 |
| 5.13 | Does the command language use normal action-object syntax? | O O X | 02/May/2010 |
| 5.14 | Does the command language avoid arbitrary, non-English use of punctuation, except for symbols that users already know? | O O X | 02/May/2010 |
| 5.15 | If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error? | O X O | 02/May/2010 |
| 5.16 | Do error messages inform the user of the error's severity? | X O O | 02/May/2010 |
| 5.17 | Do error messages suggest the cause of the problem? | X O O | 02/May/2010 |
| 5.18 | Do error messages provide appropriate semantic information? | X O O | 02/May/2010 |
| 5.19 | Do error messages provide appropriate syntactic information? | X O O | 02/May/2010 |
| 5.20 | Do error messages indicate what action the user needs to take to correct the error? | X O O | 02/May/2010 |
| 5.21 | If the system supports both novice and expert users, are multiple levels of error-message detail available? | O X O | 02/May/2010 |

Error Prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 6.1 | If the database includes groups of data, can users enter more than one group on a single screen? | X O O | 02/May/2010 |
| 6.2 | Have dots or underscores been used to indicate field length? | O X O | 02/May/2010 |
| 6.3 | Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu? | X O O | 02/May/2010 |
| 6.4 | Are menu choices logical, distinctive, and mutually exclusive? | X O O | 02/May/2010 |
| 6.5 | Are data inputs case-blind whenever possible? | X O O | 02/May/2010 |
| 6.6 | If the system displays multiple windows, is navigation between windows simple and visible? | O O X | 02/May/2010 |
| 6.7 | Are the function keys that can cause the most serious consequences in hard-to-reach positions? | O O X | 02/May/2010 |
| 6.8 | Are the function keys that can cause the most serious consequences located far away from low-consequence and high-use keys? | O O X | 02/May/2010 |
| 6.9 | Has the use of qualifier keys been minimized? | O O X | 02/May/2010 |
| 6.10 | If the system uses qualifier keys, are they used consistently throughout the system? | O O X | 02/May/2010 |
| 6.11 | Does the system prevent users from making errors whenever possible? | X O O | 02/May/2010 |
| 6.12 | Does the system warn users if they are about to make a potentially serious error? | O X O | 02/May/2010 |
| 6.13 | Does the system intelligently interpret variations in user commands? | O O X | 02/May/2010 |
| 6.14 | Do data entry screens and dialog boxes indicate the number of character spaces available in a field? | O X O | 02/May/2010 |
| 6.15 | Do fields in data entry screens and dialog boxes contain default values when appropriate? | X O O | 02/May/2010 |

### Recognition Rather Than Recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 7.1 | For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input? | X O O | 02/May/2010 |
| 7.2 | Does the data display start in the upper-left corner of the screen? | X O O | 02/May/2010 |
| 7.3 | Are multiword field labels placed horizontally (not stacked vertically)? | X O O | 02/May/2010 |
| 7.4 | Are all data a user needs on display at each step in a transaction sequence? | X O O | 02/May/2010 |
| 7.5 | Are prompts, cues, and messages placed where the eye is likely to be looking on the screen? | X O O | 02/May/2010 |
| 7.6 | Have prompts been formatted using white space, justification, and visual cues for easy scanning? | X O O | 02/May/2010 |
| 7.7 | Do text areas have "breathing space" around them? | X O O | 02/May/2010 |
| 7.8 | Is there an obvious visual distinction made between "choose one" menu and "choose many" menus? | X O O | 02/May/2010 |
| 7.9 | Have spatial relationships between soft function keys (on-screen cues) and keyboard function keys been preserved? | O O X | 02/May/2010 |
| 7.10 | Does the system gray out or delete labels of currently inactive soft function keys? | X O O | 02/May/2010 |
| 7.11 | Is white space used to create symmetry and lead the eye in the appropriate direction? | X O O | 02/May/2010 |
| 7.12 | Have items been grouped into logical zones, and have headings been used to distinguish between zones? | X O O | 02/May/2010  No headings though. |
| # | Review Checklist | Yes No N/A | Checked on |
| 7.13 | Are zones no more than twelve to fourteen characters wide and six to seven lines high? | X O O | 02/May/2010 |
| 7.14 | Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas? | X O O | 02/May/2010 |
| 7.15 | Are field labels close to fields, but separated by at least one space? | X O O | 02/May/2010 |
| 7.16 | Are long columnar fields broken up into groups of five, separated by a blank line? | O O X | 02/May/2010 |
| 7.17 | Are optional data entry fields clearly marked? | X O O | 02/May/2010 |
| 7.18 | Are symbols used to break long input strings into "chunks"? | O O X | 02/May/2010 |
| 7.19 | Is reverse video or color highlighting used to get the user's attention? | O X O | 02/May/2010 |
| 7.20 | Is reverse video used to indicate that an item has been selected? | X O O | 02/May/2010 |
| 7.21 | Are size, boldface, underlining, color, shading, or typography used to show relative quantity or importance of different screen items? | X O O | 02/May/2010 |
| 7.22 | Are borders used to identify meaningful groups? | O X O | 02/May/2010 |
| 7.23 | Has the same color been used to group related elements? | O O X | 02/May/2010 |
| 7.24 | Is color coding consistent throughout the system? | X O O | 02/May/2010 |
| 7.25 | Is color used in conjunction with some other redundant cue? | O O X | 02/May/2010 |
| 7.26 | Is there good color and brightness contrast between image and background colors? | X O O | 02/May/2010 |
| 7.27 | Have light, bright, saturated colors been used to emphasize data and have darker, duller, and desaturated colors been used to de-emphasize data? | O X O | 02/May/2010 |
| 7.28 | Is the first word of each menu choice the most important? | X O O | 02/May/2010 |
| 7.29 | Does the system provide *mapping:* that is, are the relationships between controls and actions apparent to the user? | X O O | 02/May/2010 |
| 7.30 | Are input data codes distinctive? | X O O | 02/May/2010 |
| 7.31 | Have frequently confused data pairs been eliminated whenever possible? | X O O | 02/May/2010 |
| 7.32 | Have large strings of numbers or letters been broken into chunks? | O O X | 02/May/2010 |
| 7.33 | Are inactive menu items grayed out or omitted? | X O O | 02/May/2010 |
| 7.34 | Are there menu selection defaults? | X O O | 02/May/2010 |
| 7.35 | If the system has many menu levels or complex menu levels, do users have access to an on-line spatial menu map? | O O X | 02/May/2010 |
| 7.36 | Do GUI menus offer affordance: that is, make obvious where selection is possible? | X O O | 02/May/2010 |
| 7.37 | Are there salient visual cues to identify the active window? | O O X | 02/May/2010 |
| 7.38 | Are function keys arranged in logical groups? | O O X | 02/May/2010 |
| 7.39 | Do data entry screens and dialog boxes indicate when fields are optional? | X O O | 02/May/2010 |
| 7.40 | On data entry screens and dialog boxes, are dependent fields displayed only when necessary? | O O X | 02/May/2010 |

### Fexibility and Minimalist Design

Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the “average” user (e.g., physical or cognitive ability, culture, language, etc.)

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 8.1 | If the system supports both novice and expert users, are multiple levels of error message detail available? | O O X | 02/May/2010 |
| 8.2 | Does the system allow novices to use a keyword grammar and experts to use a positional grammar? | O O X | 02/May/2010 |
| 8.3 | Can users define their own synonyms for commands? | O O X | 02/May/2010 |
| 8.4 | Does the system allow novice users to enter the simplest, most common form of each command, and allow expert users to add parameters? | O O X | 02/May/2010 |
| 8.5 | Do expert users have the option of entering multiple commands in a single string? | O O X | 02/May/2010 |
| 8.6 | Does the system provide function keys for high-frequency commands? | O X O | 02/May/2010 |
| 8.7 | For data entry screens with many fields or in which source documents may be incomplete, can users save a partially filled screen? | X O O | 02/May/2010 |
| 8.8 | Does the system automatically enter leading zeros? | O X O | 02/May/2010 |
| 8.9 | If menu lists are short (seven items or fewer), can users select an item by moving the cursor? | O O X | 02/May/2010 |
| 8.10 | If the system uses a type-ahead strategy, do the menu items have mnemonic codes? | O O X | 02/May/2010 |
| 8.11 | If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut? | O X O | 02/May/2010 |
| 8.12 | Does the system offer "find next" and "find previous" shortcuts for database searches? | O X O | 02/May/2010 |
| 8.13 | On data entry screens, do users have the option of either clicking directly on a field or using a keyboard shortcut? | X O O | 02/May/2010 |
| 8.14 | On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut? | X O O | 02/May/2010 |
| 8.15 | In dialog boxes, do users have the option of either clicking directly on a dialog box option or using a keyboard shortcut? | X O O | 02/May/2010 |
| 8.16 | Can expert users bypass nested dialog boxes with either type-ahead, user-defined macros, or keyboard shortcuts? | O X O | 02/May/2010 |

### Aesthetic and Minimalist Design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 9.1 | Is only (and all) information essential to decision making displayed on the screen? | X O O | 02/May/2010 |
| 9.2 | Are all icons in a set visually and conceptually distinct? | X O O | 02/May/2010 |
| 9.3 | Have large objects, bold lines, and simple areas been used to distinguish icons? | X O O | 02/May/2010 |
| 9.4 | Does each icon stand out from its background? | X O O | 02/May/2010 |
| 9.5 | If the system uses a standard GUI interface where menu sequence has already been specified, do menus adhere to the specification whenever possible? | O O X | 02/May/2010 |
| 9.6 | Are meaningful groups of items separated by white space? | X O O | 02/May/2010 |
| 9.7 | Does each data entry screen have a short, simple, clear, distinctive title? | X O O | 02/May/2010 |
| 9.8 | Are field labels brief, familiar, and descriptive? | X O O | 02/May/2010 |
| 9.9 | Are prompts expressed in the affirmative, and do they use the active voice? | X O O | 02/May/2010 |
| 9.10 | Is each lower-level menu choice associated with only one higher level menu? | O X O | 02/May/2010 |
| 9.11 | Are menu titles brief, yet long enough to communicate? | X O O | 02/May/2010 |
| 9.12 | Are there pop-up or pull-down menus within data entry fields that have many, but well-defined, entry options? | X O O | 02/May/2010 |

### Help and Documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 10.1 | If users are working from hard copy, are the parts of the hard copy that go on-line marked? | O O X | 02/May/2010 |
| 10.2 | Are on-line instructions visually distinct? | X O O | 02/May/2010 |
| 10.3 | Do the instructions follow the sequence of user actions? | X O O | 02/May/2010 |
| 10.4 | If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected? | X O O | 02/May/2010 |
| 10.5 | Are data entry screens and dialog boxes supported by navigation and completion instructions? | X O O | 02/May/2010 |
| 10.6 | If menu items are ambiguous, does the system provide additional explanatory information when an item is selected? | X O O | 02/May/2010 |
| 10.7 | Are there memory aids for commands, either through on-line quick reference or prompting? | O O X | 02/May/2010 |
| 10.8 | Is the help function visible; for example, a key labeled HELP or a special menu? | O O X | 02/May/2010 |
| 10.9 | Is the help system interface (navigation, presentation, and conversation) consistent with the navigation, presentation, and conversation interfaces of the application it supports? | O O X | 02/May/2010 |
| 10.10 | Navigation: Is information easy to find? | X O O | 02/May/2010 |

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 10.11 | Presentation: Is the visual layout well designed? | X O O | 02/May/2010 |
| 10.12 | Conversation: Is the information accurate, complete, and understandable? | X O O | 02/May/2010 |
| 10.13 | Is the information relevant? | X O O | 02/May/2010 |
| 10.14 | Goal-oriented (What can I do with this program?) | X O O | 02/May/2010 |
| 10.15 | Descriptive (What is this thing for?) | X O O | 02/May/2010 |
| 10.16 | Procedural (How do I do this task?) | X O O | 02/May/2010 |
| 10.17 | Interpretive (Why did that happen?) | X O O | 02/May/2010 |
| 10.18 | Navigational (Where am I?) | X O O | 02/May/2010 |
| 10.19 | Is there context-sensitive help? | O X O | 02/May/2010 |
| 10.20 | Can the user change the level of detail available? | O X O | 02/May/2010 |
| 10.21 | Can users easily switch between help and their work? | O O X | 02/May/2010 |
| 10.22 | Is it easy to access and return from the help system? | O O X | 02/May/2010 |
| 10.23 | Can users resume work where they left off after accessing help? | O O X | 02/May/2010 |

### Skills

The system should support, extend, supplement, or enhance the user’s skills, background knowledge, and expertise ----not replace them.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 11.1 | Can users choose between iconic and text display of information? | O X O | 03/May/2010 |
| 11.2 | Are window operations easy to learn and use? | X O O | 03/May/2010 |
| 11.3 | If users are experts, usage is frequent, or the system has a slow response time, are there fewer screens (more information per screen)? | O X O | 03/May/2010 |
| 11.4 | If users are novices, usage is infrequent, or the system has a fast response time, are there more screens (less information per screen)? | O X O | 03/May/2010 |
| 11.5 | Does the system automatically color-code items, with little or no user effort? | O X O | 03/May/2010 |
| 11.6 | If the system supports both novice and expert users, are multiple levels of detail available. | O O X | 03/May/2010 |
| 11.7 | Are users the initiators of actions rather than the responders? | X O O | 03/May/2010 |
| 11.8 | Does the system perform data translations for users? | X O O | 03/May/2010 |
| 11.9 | Do field values avoid mixing alpha and numeric characters whenever possible? | X O O | 03/May/2010 |
| 11.10 | If the system has deep (multilevel) menus, do users have the option of typing ahead? | O O X | 03/May/2010 |
| 11.12 | When the user enters a screen or dialog box, is the cursor already positioned in the field users are most likely to need? | X O O | 03/May/2010 |
| 11.13 | Can users move forward and backward within a field? | X O O | 03/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 11.14 | Is the method for moving the cursor to the next or previous field both simple and visible? | X O O | 03/May/2010 |
| 11.15 | Has auto-tabbing been avoided except when fields have fixed lengths or users are experienced? | O O X | 03/May/2010 |
| 11.16 | Do the selected input device(s) match user capabilities? | O O X | 03/May/2010 |
| 11.17 | Are cursor keys arranged in either an inverted T (best for experts) or a cross configuration (best for novices)? | X O O | 03/May/2010 |
| 11.18 | Are important keys (for example, ENTER , TAB) larger than other keys? | X O O | 03/May/2010 |
| 11.19 | Are there enough function keys to support functionality, but not so many that scanning and finding are difficult? | X O O | 03/May/2010 |
| 11.20 | Are function keys reserved for generic, high-frequency, important functions? | O O X | 03/May/2010 |
| 11.21 | Are function key assignments consistent across screens, subsystems, and related products? | O O X | 03/May/2010 |
| 11.22 | Does the system correctly anticipate and prompt for the user's probable next activity? | O O X | 03/May/2010 |

### Pleasurable and Respectful Interaction with the User

The user’s interactions with the system should enhance the quality of her or his work-life. The user should be treated with respect. The design should be aesthetically pleasing- with artistic as well as functional value.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 12.1 | Is each individual icon a harmonious member of a family of icons? | X O O | 03/May/2010 |
| 12.2 | Has excessive detail in icon design been avoided? | X O O | 03/May/2010 |
| 12.3 | Has color been used with discretion? | X O O | 03/May/2010 |
| 12.4 | Has the amount of required window housekeeping been kept to a minimum? | X O O | 03/May/2010 |
| 12.5 | If users are working from hard copy, does the screen layout match the paper form? | O O X | 03/May/2010 |
| 12.6 | Has color been used specifically to draw attention, communicate organization, indicate status changes, and establish relationships? | O X O | 03/May/2010 |
| 12.7 | Can users turn off automatic color coding if necessary? | O O X | 03/May/2010 |
| 12.8 | Are typing requirements minimal for question and answer interfaces? | X O O | 03/May/2010 |
| 12.9 | Do the selected input device(s) match environmental constraints? | X O O | 03/May/2010 |
| 12.13 | If the system uses multiple input devices, has hand and eye movement between input devices been minimized? | O O X | 03/May/2010 |
| 12.14 | If the system supports graphical tasks, has an alternative pointing device been provided? | O O X | 03/May/2010 |
| 12.15 | Is the numeric keypad located to the right of the alpha key area? | X O O | 03/May/2010 |
| # | Review Checklist | Yes No N/A | Checked on |
| 12.16 | Are the most frequently used function keys in the most accessible positions? | O O X | 03/May/2010 |
| 12.17 | Does the system complete unambiguous partial input on a data entry field? | O X O | 03/May/2010 |

### Privacy

The system should help the user to protect personal or private information- belonging to the user or the his/her clients.

|  |  |  |  |
| --- | --- | --- | --- |
| # | Review Checklist | Yes No N/A | Checked on |
| 13.1 | Are protected areas completely inaccessible? | X O O | 03/May/2010 |
| 13.2 | Can protected or confidential areas be accessed with certain passwords. | X O O | 03/May/2010 |
| 13.3 | Is this feature effective and successful. | X O O | 03/May/2010 |

*References*

(Xerox Corporation*, 1995;* Weiss, E. 1993.; Nielsen, J. Mack, R. 1994.)

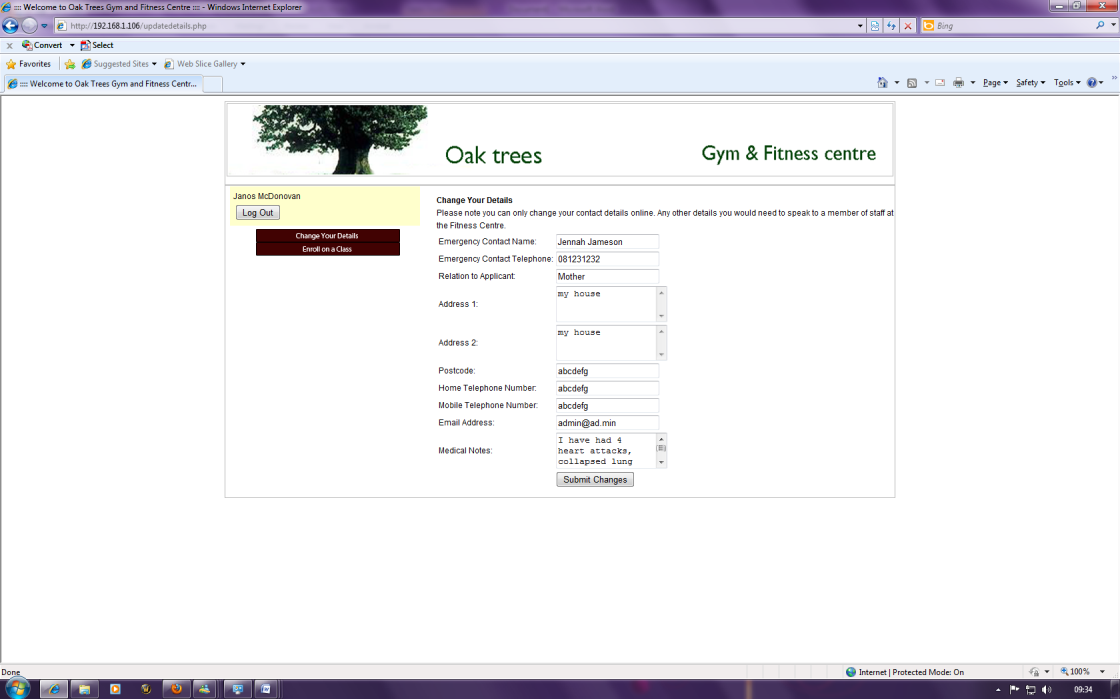
## Website Testing

Small Issues

IE: When changing details, the confirmation box comes up twice and the page almost drops part of the CSS.



The change details section is now below the menu, when it should be along side it.



So it appears there is some loss of formatting somewhere. This does not happen in the other browsers however. It only seems to be happening in IE at the moment.

### Website Test Plan – Firefox

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Content renders correctly | Loading web page into browser | Content to appear rendered correctly | Pass | 23/04/2010 |
| 2 | Links working correctly | Loading web page into browser | Links to go to the correct page | Pass | 23/04/2010 |
| 3 | Database connection established | Loading the web page and attempting to connect to db | Connects to database correctly | Pass | 23/04/2010 |
| 4 | Database information displayed correctly | Loading web page, establishing db connection | Displays information from database correctly | Pass | 23/04/2010 |
| 5 | Changes to database being permitted when required | Loading web page and change data in db | Allows changes to database when its required | Pass | 23/04/2010 |
| 6 | Logging in shows correct message if incorrect | Loading web page and attempting to log in | Should display page saying that log in details are incorrect | Pass | 23/04/2010 |
| 7 | JavaScript validating numbers in number field | Attempting to add letters to numeric field | Should error when non numeric characters are entered | Fail | 23/04/2010 |
| 8 | JavaScript validating postcode in postcode field | Attempting to enter something other than a postcode | Should error when something that is not a postcode is entered | Fail | 23/04/2010 |
| 9 | Class Booking system allows member to enroll on class | Attempting to enroll on a class | Should allow member to enroll on a class when they request it unless full | Pass | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Attempting to log out of the system | Should show a logged out page | Pass | 23/04/2010 |

### Website Test Plan - Chrome

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Content renders correctly | Loading web page into browser | Content to appear rendered correctly | Pass | 23/04/2010 |
| 2 | Links working correctly | Loading web page into browser | Links to go to the correct page | Pass | 23/04/2010 |
| 3 | Database connection established | Loading the web page and attempting to connect to db | Connects to database correctly | Pass | 23/04/2010 |
| 4 | Database information displayed correctly | Loading web page, establishing db connection | Displays information from database correctly | Pass | 23/04/2010 |
| 5 | Changes to database being permitted when required | Loading web page and change data in db | Allows changes to database when its required | Pass | 23/04/2010 |
| 6 | Logging in shows correct message if incorrect | Loading web page and attempting to log in | Should display page saying that log in details are incorrect | Pass | 23/04/2010 |
| 7 | JavaScript validating numbers in number field | Attempting to add letters to numeric field | Should error when non numeric characters are entered | Fail | 23/04/2010 |
| 8 | JavaScript validating postcode in postcode field | Attempting to enter something other than a postcode | Should error when something that is not a postcode is entered | Fail | 23/04/2010 |
| 9 | Class Booking system allows member to enroll on class | Attempting to enroll on a class | Should allow member to enroll on a class when they request it unless full | Pass | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Attempting to log out of the system | Should show a logged out page | Pass | 23/04/2010 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Test Data** | **Expected Result** | **Pass/Fail** | **Date of Test** |
| 1 | Content renders correctly | Loading web page into browser | Content to appear rendered correctly | Pass | 23/04/2010 |
| 2 | Links working correctly | Loading web page into browser | Links to go to the correct page | Pass | 23/04/2010 |
| 3 | Database connection established | Loading the web page and attempting to connect to db | Connects to database correctly | Pass | 23/04/2010 |
| 4 | Database information displayed correctly | Loading web page, establishing db connection | Displays information from database correctly | Pass | 23/04/2010 |
| 5 | Changes to database being permitted when required | Loading web page and change data in db | Allows changes to database when its required | Pass | 23/04/2010 |
| 6 | Logging in shows correct message if incorrect | Loading web page and attempting to log in | Should display page saying that log in details are incorrect | Pass | 23/04/2010 |
| 7 | JavaScript validating numbers in number field | Attempting to add letters to numeric field | Should error when non numeric characters are entered | Fail | 23/04/2010 |
| 8 | JavaScript validating postcode in postcode field | Attempting to enter something other than a postcode | Should error when something that is not a postcode is entered | Fail | 23/04/2010 |
| 9 | Class Booking system allows member to enroll on class | Attempting to enroll on a class | Should allow member to enroll on a class when they request it unless full | Pass | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Attempting to log out of the system | Should show a logged out page | Pass | 23/04/2010 |

### Website Test Plan - Internet Explorer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Actual Result** | **Result (P/F)** | **Comment** | **Date of Test** |
| 1 | To check to see if the content renders correctly when the page is loaded | Pages load fine | Pass | Pages loaded fine, css working correctly | 23/04/2010 |
| 2 | Links working correctly and going to the correct page | Links go to correct places | Pass | Some of the pages need some content, but at this stage as long as the links work its good | 23/04/2010 |
| 3 | Allowing a connection to the database | Logs in a user with no problems | Pass | The fact that a user is being logged in is all I really needed to do to test this | 23/04/2010 |
| 4 | Displaying information from the database to the screen | Loaded the change details page, was populated fine | Pass | All fields loaded correctly, some differences in font but that was due to the different text boxes | 23/04/2010 |
| 5 | Allowing changes to be made to database data when required | Changes are allowed to be made to the db | Pass | The changes are being permitted through the sql queries in the php | 23/04/2010 |
| 6 | Logging in shows the correct message when an incorrect value is entered | Attempting to log in brings up incorrect message | Pass | Loads the page saying that the username and password is incorrect - which is right! | 23/04/2010 |
| 7 | JavaScript validating correctly if a non numeric character is entered in numeric field | Allowed changes to go through unvalidated | Fail | This is something that could be rather simple to implement, but at this stage we don't have the time. If it was going live however, | 23/04/2010 |
| 8 | JavaScript validating correctly if a value that isn't a postcode is entered in a postcode field | Allowed changes to go through unvalidated | Fail | Then this would be implemented to prevent changes done to the database that would affect the fields in question. | 23/04/2010 |
| 9 | Allowing a member to enroll onto a class using the class booking system | Allowed the user to book onto a class | Pass | Spelling error on the pop up box that asks for confirmation of the class booking | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Returned the user to the home page | Pass | Ideally to let the user know they had logged out would be beter. | 23/04/2010 |

### Website Test Log – Firefox

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Actual Result** | **Result (P/F)** | **Comment** | **Date of Test** |
| 1 | To check to see if the content renders correctly when the page is loaded | Pages load fine | Pass | Pages loaded fine, css working correctly | 23/04/2010 |
| 2 | Links working correctly and going to the correct page | Links go to correct places | Pass | Some of the pages need some content, but at this stage as long as the links work its good | 23/04/2010 |
| 3 | Allowing a connection to the database | Logs in a user with no problems | Pass | The fact that a user is being logged in is all I really needed to do to test this | 23/04/2010 |
| 4 | Displaying information from the database to the screen | Loaded the change details page, was populated fine | Pass | All fields loaded correctly, some differences in font but that was due to the different text boxes | 23/04/2010 |
| 5 | Allowing changes to be made to database data when required | Changes are allowed to be made to the db | Pass | The changes are being permitted through the sql queries in the php | 23/04/2010 |
| 6 | Logging in shows the correct message when an incorrect value is entered | Attempting to log in brings up incorrect message | Pass | Loads the page saying that the username and password is incorrect - which is right! | 23/04/2010 |
| 7 | JavaScript validating correctly if a non numeric character is entered in numeric field | Allowed changes to go through unvalidated | Fail | This is something that could be rather simple to implement, but at this stage we don't have the time. If it was going live however, | 23/04/2010 |
| 8 | JavaScript validating correctly if a value that isn't a postcode is entered in a postcode field | Allowed changes to go through unvalidated | Fail | Then this would be implemented to prevent changes done to the database that would affect the fields in question. | 23/04/2010 |
| 9 | Allowing a member to enroll onto a class using the class booking system | Allowed the user to book onto a class | Pass | Spelling error on the pop up box that asks for confirmation of the class booking | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Returned the user to the home page | Pass | Ideally showing a page that would let the user know they had logged out would be better | 23/04/2010 |

### Website Test Log – Chrome

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Actual Result** | **Result (P/F)** | **Comment** | **Date of Test** |
| 1 | To check to see if the content renders correctly when the page is loaded | Pages are displayed as per design | Pass | Everything loads correctly when the page is loaded into the browser | 23/04/2010 |
| 2 | Links working correctly and going to the correct page | Links go to correct destination | Pass | All links take user to the correct page when they are clicked on | 23/04/2010 |
| 3 | Allowing a connection to the database | User logs in without a problem | Pass | If the user couldn't log in then there would be an issue with the database | 23/04/2010 |
| 4 | Displaying information from the database to the screen | Loads the page with changes fine | Pass | Page is loaded with changes done in another browser, so they are getting saved to db | 23/04/2010 |
| 5 | Allowing changes to be made to database data when required | Allows the changes through fine | Pass | Changes are written back to the database like it should be | 23/04/2010 |
| 6 | Logging in shows the correct message when an incorrect value is entered | Shows incorrect login page | Pass | User cannot log in if they enter an incorrect username and password - works as intended | 23/04/2010 |
| 7 | JavaScript validating correctly if a non numeric character is entered in numeric field | Doesn't validate how it is supposed to, allows anything through | Fail | At this stage, the fact that a small piece of JavaScript does not work is not really a cause for concern. If this was being launched then | 23/04/2010 |
| 8 | JavaScript validating correctly if a value that isn't a postcode is entered in a postcode field | Doesn't validate how it is supposed to, allows anything through | Fail | Ideally this would need to be fixed and tested again to ensure that no illegal entries could be added to the database | 23/04/2010 |
| 9 | Allowing a member to enroll onto a class using the class booking system | Allows the user to enroll on class | Pass | Had to cancel booking to enroll on it again, as logged in using the same login details | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Returns to home page | Pass | Should show that they have logged out, but it works, that's what is important! | 23/04/2010 |

### Website Test Log - Internet Explorer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Purpose of Test** | **Actual Result** | **Result (P/F)** | **Comment** | **Date of Test** |
| 1 | To check to see if the content renders correctly when the page is loaded | Page loads with no errors | Pass | Page renders fine, no errors or anything when it loads | 23/04/2010 |
| 2 | Links working correctly and going to the correct page | Links go to correct pages | Pass | The links work correctly, taking the user to the correct page | 23/04/2010 |
| 3 | Allowing a connection to the database | Logs the user in with no problems | Pass | Seeing as the user can log in, there are no problems with the database connection | 23/04/2010 |
| 4 | Displaying information from the database to the screen | All details from previous browser test loaded | Pass | As details from the previous browser test are loading, it is a sign that this is still working across different platforms | 23/04/2010 |
| 5 | Allowing changes to be made to database data when required | Details are submitted, page breaks afterwards | Pass | Changes are submitted but page layout goes funny afterwards... hmm. Dialog box appears twice and then the page breaks | 23/04/2010 |
| 6 | Logging in shows the correct message when an incorrect value is entered | Shows error if wrong details | Pass | Loads the page saying that the username and password is incorrect - which is right! | 23/04/2010 |
| 7 | JavaScript validating correctly if a non numeric character is entered in numeric field | Not bringing up an error if an incorrect value is entered | Fail | This is something that could be rather simple to implement, but at this stage we don't have the time. If it was going live however, | 23/04/2010 |
| 8 | JavaScript validating correctly if a value that isn't a postcode is entered in a postcode field | Not bringing up an error if an incorrect value is entered | Fail | Then this would be implemented to prevent changes done to the database that would affect the fields in question. | 23/04/2010 |
| 9 | Allowing a member to enroll onto a class using the class booking system | Allows user to enroll on a class | Pass | Enrolls fine and doesn't break the page after the dialog box has been and gone | 23/04/2010 |
| 10 | Allowing the user to log out properly and closes the database connection | Logs the user out | Pass | User is successfully logged out of the system when clicking logout. | 23/04/2010 |

# Overview

While testing the website, I considered testing it on the web server I have set up, or testing it through another machine to get the feel of it from a client perspective. After disabling the firewall of the web server, I was able to make contact with the server from my client machine and test each page as if it were from a users point of view. Choosing to use three web browsers could be one of the most accurate methods of testing a website, simply because not everyone uses the same browser. While I have only tested on the three main ones, the test results infer that if it was accessed on something other than Internet Explorer, Google Chrome or Mozilla Firefox, they should have very few problems.

The testing of the website proved to be successful in that the majority of the tests gave a pass result. The only tests that failed on each of the browser tests was the JavaScript validation, which would prevent a user entering a non numeric character into a field that required characters, such as telephone number and a series of characters into the postcode field that didn't correspond to something that resembled a postcode.

Whilst the testing was mostly successful, a few small issues were highlighted. As shown above the testing tables, when trying to change details using Internet Explorer, the dialog box confirming the changes have been made appears twice, and after you have clicked ok to get rid of it, the page layout changes. This does not happen on the class booking page, despite the fact a similar dialog box is used. This is only happening with Internet Explorer, and only on this page.

One thing that does stand out is the current lack of content on some of the pages. Whilst this web based addition to our main C# application will never get published, it would be nice if it was something that resembled a finished product. However due to the time constraints, this is not going to be possible this late in the project. The pages themselves resemble the design closely, and I feel that in its development, the design has been followed thoroughly to make it as true to what was required as possible.

# Appendix



## Bibliography - References

### References from documentation

Cisco, 800 Series Routers - Products & Services (2010a) [Online] (Updated 6 May 2010)

Available at: <http://www.cisco.com/en/US/products/hw/routers/ps380/index.html> [Accessed 6 May 2010].

Cisco, Catalyst Express 500 Series Switches - Products & Services (2010b) [Online]

(Updated 6 May 2010) Available at: <http://www.cisco.com/en/US/products/ps6545/index.html> [Accessed 6 May 2010].

Cisco, Cisco Packet Tracer (2010c) [Online] (Updated 6 May 2010) Available at: <http://www.cisco.com/web/learning/netacad/course_catalog/PacketTracer.html> [Accessed 6 May 2010].

CiscoSystems, Why Use Cisco Network Systems?  [Cisco Catalyst 6500 Series Switches] - (2010) [Online] (Updated 6 May 2010) Available at: <http://www.ciscosystems.com/en/US/prod/collateral/switches/ps5718/ps708/product_solution_overview0900aecd80630c78.html> [Accessed 6 May 2010].

Dell, Vostro 230s Slim Tower Desktop | Dell UK. (2010) [Online] (Updated 6 May 2010)

Available at: <http://www1.euro.dell.com/uk/en/business/Desktops/vostro-230st/pd.aspx?refid=vostro-230st&s=bsd&cs=ukbsdt1> [Accessed 6 May 2010].

Nielsen, J. Mack, R. 1994.*Usability Inspection Methods. ISBN: 1-55542*-622-0

Weiss, E. 1993. *Making Computers People Literate.* ISBN: 0-471-01877-5

Xerox Corporation*, 1995. Usability Analysis & Design,* Xerox Corporation*.*

### References from code

#### frm\_member.cs, frm\_staff.cs, frm\_equipment\_bookings\_list.cs, frm\_class\_instance\_arrange

Alani, S. 2006. *How to check string is number or not in C#*.[online] (Updated 17 Oct 2009) Available at: <http://social.msdn.microsoft.com/Forums/en-US/winforms/thread/84990ad2-5046-472b-b103-f862bfcd5dbc> [Accessed 6 May 2010].

#### frm\_equipment.cs

Evilertoaster, 2010. *ComboBox Default Value? | debugging*.[online] (Updated 6 May 2010) Available at: <http://www.debugging.com/bug/12621> [Accessed 6 May 2010].

Mayo, J. 2009. *C# Station: C# Tutorial Lesson 14 - Introduction to Delegates and Events*.[online] (Updated 12 Jan 2009) Available at: <http://www.csharp-station.com/Tutorials/Lesson14.aspx> [Accessed 6 May 2010].

MSDN, 2010. *DateTime Methods (System)*.[online] (Updated 6 May 2010) Available at: <http://msdn.microsoft.com/en-us/library/system.datetime_methods%28v=VS.90%29.aspx> [Accessed 6 May 2010].

*MySQL :: MySQL 5.1 Reference Manual :: 11.6 Date and Time Functions*.[online] (Updated 6 May 2010) Available at: <http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html#function_date> [Accessed 6 May 2010].

Slama, J., 2008. *String Format for DateTime [C#]*.[online] (Updated 6 May 2010) Available at: <http://www.csharp-examples.net/string-format-datetime/> [Accessed 6 May 2010].

#### frm\_member.cs

CSharpFreak, 2006. *How can i create a "right-click menu" i C#*.[online] (Updated 15 Sept 2006) Available at: <http://social.msdn.microsoft.com/Forums/en-US/winforms/thread/790cd8be-0ba8-4f10-95a8-c88f1023d6e7> [Accessed 6 May 2010].

C# Online.NET, 2010.  *Working with Images—Using the Bitmap Class (CSharp-Online.NET)*.[online] (Updated 6 May 2010) Available at: <http://en.csharp-online.net/Working_with_Images%E2%80%94Using_the_Bitmap_Class> [Accessed 6 May 2010].

DevAsp, 2006. *How to load Image in C# and set properties of the Picture Box - .Net Articles & Samples*.[online] (Updated 24 Apr 2010) Available at: <http://www.devasp.net/net/articles/display/385.html> [Accessed 6 May 2010].

Java2s.com, 2010. *Mouse left click, right click, middle click, hover : Mouse Event « Event « C# / C Sharp*.[online] (Updated 6 May 2010) Available at: <http://www.java2s.com/Code/CSharp/Event/Mouseleftclickrightclickmiddleclickhover.htm> [Accessed 6 May 2010].

Meshplex, 2008. *C Sharp/String Literals - Meshplex*.[online] (Updated 4 Oct 2008) Available at: <http://www.meshplex.org/wiki/C_Sharp/String_Literals> [Accessed 6 May 2010].

Rae, M. 2005. *C# Tool tip text - C# / C Sharp answers*.[online] (Updated 17 Nov 2005) Available at: <http://bytes.com/topic/c-sharp/answers/272413-c-tool-tip-text> [Accessed 6 May 2010].

Vaupel, R., 2009.*MySQL :: PROBLEM: INSERT Blob-Data*.[online] (Updated 3 Oct 2009) Available at: <http://forums.mysql.com/read.php?47,284456,284456> [Accessed 6 May 2010].

#### Class.cs

Microsoft, Capitalization Conventions. (2010) [Online] (Updated 6 May 2010) Available at: <http://msdn.microsoft.com/en-us/library/ms229043%28v=VS.90%29.aspx> [Accessed 6 May 2010].

#### frm\_member\_list.cs

Aranbs, 2009. *line break in message box - C# / C Sharp answers*.[online] (Updated 14 Mar 2009) Available at: <http://bytes.com/topic/c-sharp/answers/866510-line-break-message-box> [Accessed 6 May 2010].

MSDN, 2010a.*Clipboard Class (System.Windows.Forms)*.[online] (Updated 6 May 2010) Available at: <http://msdn.microsoft.com/en-us/library/system.windows.forms.clipboard.aspx> [Accessed 6 May 2010].

MSDN, 2010b. *Clipboard.ContainsData Method (System.Windows.Forms)*.[online] (Updated 6 May 2010) Available at: <http://msdn.microsoft.com/en-us/library/system.windows.forms.clipboard.containsdata.aspx> [Accessed 6 May 2010].

MSDN, 2010c. *DataGridView.SelectedCells Property (System.Windows.Forms)*.[online] (Updated 6 May 2010) Available at: <http://msdn.microsoft.com/en-us/library/system.windows.forms.datagridview.selectedcells.aspx> [Accessed 6 May 2010].

#### frm\_message\_box.cs

*Drumm, D., 2005. Create Custom Dialog like MessageBox - C# / C Sharp answers*.[online] (Updated 15 Nov 2005) Available at: <http://bytes.com/topic/c-sharp/answers/227206-create-custom-dialog-like-messagebox> [Accessed 6 May 2010].

Hussain, S., 2007. *Creating a Custom Message Box - CodeGuru*.[online] (Updated 21 May 2007) Available at: <http://www.codeguru.com/csharp/.net/net_general/eventsanddelegates/article.php/c13193> [Accessed 6 May 2010].

#### Outlookbar.cs

Clifton, M. 2003. *An Outlook Bar Implementation - CodeProject*.[online] (Updated 12 Apr 2003) Available at: <http://www.codeproject.com/KB/selection/csoutlookbar.aspx> [Accessed 6 May 2010].

Ivar, 2002. *Outlook Bar Control*.[online] (Updated 18 Apr 2002) Available at: <http://www.c-sharpcorner.com/UploadFile/ivar/OutlookBarIL11212005014057AM/OutlookBarIL.aspx> [Accessed 6 May 2010].

Kuijpers, H., 2006. *OutlookBar: A Simplified Outlook Style Sidebar Control - CodeProject*.[online] (Updated 10 Nov 2006) Available at: <http://www.codeproject.com/KB/menus/OutlookBar.aspx> [Accessed 6 May 2010].

## Licenses

### Free Stock Icons

Free Icons used from:

<http://www.iconarchive.com/category/application/aesthetica-2-icons-by-dryicons.html>

Under free licence:

<http://dryicons.com/terms/>

### Women at Fitness Club Image

The class icon images were derived from this picture:

[http://www.bigstockphoto.com/image-2678004/stock-photo-group-of-woman-at-fitness-club](https://owa.anglia.ac.uk/exchweb/bin/redir.asp?URL=http://www.bigstockphoto.com/image-2678004/stock-photo-group-of-woman-at-fitness-club)

Permission for using the image for the project was given as:

Hi, Janos

Since this would be for internal use only and the image file itself  
would not be accessible by itself to anyone this type of usage would be  
perfectly acceptable under our standard usage agreement and would not  
require a special license.  
  
Thank you.  
  
----  
Joseph  
Customer Support  
Bigstock  
155 West 19th Street, 2nd Floor  
New York, NY 10011  
1-800-631-9809  
[http://www.bigstock.com](https://owa.anglia.ac.uk/exchweb/bin/redir.asp?URL=http://www.bigstock.com)

## Class Documentation

Separate Printout Attached to the End of this Document.

## Initial interface designs

Separate Printout Attached to the End of this Document.

## Course Logbook

Attached to the End of this Document