KNM Consultants

GYM DATABASE SYSTEM

VERSION 1.0

THE WAREHOUSE GYM

DATABASE REPORT DOCUMENT

MICROSOFT ACCESS

THIS GYM DATABASE SYSTEM IS A MICROSOFT ACCESS RELATIONAL DATABASE MANAGEMENT SYSTEM WHICH ENABLES THE MANAGEMENT OF ALL CORE GYM FUNCTIONS FROM TRACKING ALL MEMBERSHIP FEES DUE TO THE GYM, TO ADDING NEW MEMBERS, TO PRODUCING REPORTS FOR PAYROLL.

DATABASE MANUAL

THIS DOCUMENT COMPRISES OF THE DESIGN GUIDANCE REPORT AND THE TECHNICAL REPORT WHICH WILL OUTLINE THE STEPS TAKEN AS PART OF THE DESIGN PROCESS AS WELL AS A STEP-BY-STEP INSTRUCTIONAL MANUAL OF HOW THIS DATABASE WAS CREATED AND DESIGNED TO BE USED WITHIN THE WAREHOUSE GYM ORGANISATION..

MODULE CODE: 52230
APPLICATION DEVELOPMENT
DATE OF SUBMISSION: 22/11/2019

Kevin Blaney G00333093
Nigel Waldron G00352112
Michael O'Grady G00351238

APPLICATION DEVELOPMENT

GMIT
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICHEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY
CASTLEBAR
LETTERFRACK
MOUNTBELLEW

Assignment Cover Sheet

Student Name:	Kevin Blaney
Student Number:	G00333097
Programme:	Business Information Systems
Year:	4
Module:	App Development — Module Code: 52330
Lecturer	Dr Trevor Clohessy
Assignment Title:	Database Project
Number of Pages:	
Due Date:	22 Nov 2019
Date Submitted:	22 Nov 19
Additional Information:	

Plagiarism Disclaimer:

The School of Business recognises that plagiarism is a serious academic offence and reserves the right to deal with breach of the offence appropriately.

I have read and understand the 'School of Business Style Guide for Assignments' and agree to the guidelines set out therein in relation to plagiarism and referencing. I confirm that the entire content of this assignment has been completed by me, and is not the work of another. I also confirm that I have properly referenced and acknowledged all sources used in preparation of this assignment. I understand that if I plagiarise, or if I knowingly assist others in plagiarising that I may receive a mark of zero and be brought before the disciplinary committee.

I declare that this is entirely my work based on my own personal study and/or research and I further declare that I have not engaged the services of another to either assist in or complete this assignment.

Signed:	Kevin Blaney
Date:	21/11/2019

Please note: Students MUST retain a hard/soft copy of all assignments.

For Lecturer Use Only:	
Mark Awarded:	
Marked Out of:	
Comments:	

GMIT
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY
CASTLEBAR
LETTERFRACK
MOUNTBELLEW

Assignment Cover Sheet

Student Name:	Nigel Waldron		
Student Number:	G00352112		
Programme:	Business Information Systems		
Year:	Year 4		
Module:	Application Development	Module Code:	COMP08053
Lecturer	Trevor Clohessy		
Assignment Title:	Application Development Project		
Number of Pages:			
Due Date:	22-Nov-2019		
Date Submitted:	22-Nov-2019		
Additional Information:			

Plagiarism Disclaimer:

The School of Business recognises that plagiarism is a serious academic offence and reserves the right to deal with breach of the offence appropriately.

I have read and understand the 'School of Business Style Guide for Assignments' and agree to the guidelines set out therein in relation to plagiarism and referencing. I confirm that the entire content of this assignment has been completed by me, and is not the work of another. I also confirm that I have properly referenced and acknowledged all sources used in preparation of this assignment. I understand that if I plagiarise, or if I knowingly assist others in plagiarising that I may receive a mark of zero and be brought before the disciplinary committee.

I declare that this is entirely my work based on my own personal study and/or research and I further declare that I have not engaged the services of another to either assist in or complete this assignment.

Signed:	Nigel Waldron
Date:	22-Nov-2019

Please note: Students MUST retain a hard/soft copy of all assignments.

For Lecturer Use Only:	
Mark Awarded:	
Marked Out of:	
Comments:	

GMIT
GALWAY-MAYO INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO

GALWAY
CASTLEBAR
LETTERFRACK
MOUNTBELLEW

Assignment Cover Sheet

Student Name:	MICHAEL O GRADY		
Student Number:	G00351238		
Programme:	Business Information Systems BSc hons L8		
Year:	2019		
Module:	Application Development	Module Code:	52230
Lecturer	DR. Trevor Clohessy		
Assignment Title:	The Warehouse Gym Database		
Number of Pages:	30		
Due Date:	22/11/2019		
Date Submitted:	22/11/2019		
Additional Information:			

Plagiarism Disclaimer:

The School of Business recognises that plagiarism is a serious academic offence and reserves the right to deal with breach of the offence appropriately.

I have read and understand the 'School of Business Style Guide for Assignments' and agree to the guidelines set out therein in relation to plagiarism and referencing. I confirm that the entire content of this assignment has been completed by me, and is not the work of another. I also confirm that I have properly referenced and acknowledged all sources used in preparation of this assignment. I understand that if I plagiarise, or if I knowingly assist others in plagiarising that I may receive a mark of zero and be brought before the disciplinary committee.

I declare that this is entirely my work based on my own personal study and/or research and I further declare that I have not engaged the services of another to either assist in or complete this assignment.

Signed:	Michael O Grady
Date:	21/11/2019

Please note: Students MUST retain a hard/soft copy of all assignments.

For Lecturer Use Only:	
Mark Awarded:	
Marked Out of:	
Comments:	

Contents

Design Guidance Report 2
Background 2
Rationale 2
List of Assumptions 3
Overview of Challenges Encountered and how they were Overcame 4
Overview of how the Warehouse Gym Database is designed in line with the 5 Characteristics of Good Databases 5
Data Integrity 5
Data Independence: 6
Physical Data Independence 6
Logical Data Independence: 6
Avoiding Data Redundancy 7
Data Security: 8
Data Maintenance 9
Microsoft OneNote Project Log 9
Bibliography: 10
Technical Report 11
A step by step instruction manual of the creation process 11
Steps taken during the Design process 16
Testing procedures that were carried out to test the database 34
..... 35

Log in Credentials			
Admin Users (Admin form)		**Staff User (Navigation form)**	
Username	Password	Username	Password
michael	pass	adam	pass
nigel	pass		
kevin	pass		

Design Guidance Report

Background

The organisation we have selected is the Warehouse Gym. This Gym has two locations in Mervue and Oranmore which makes it a popular gym chain based here in Galway. The Warehouse Gym have a large number of members across two sites offering a number of memberships and various classes. They are opened from 6:00am till midnight Monday to Friday, and 8:00am till 10:00pm - Sat & Sun. They have personal trainers who operate as staff members who offer assistance to members along with conducting various classes. The Warehouse Gym also offers a new work experience program on their website for personal trainers and also offer members fitness programs to follow. They are also partnered with major health companies based in Ireland such as life fitness, Stairmaster® and Flexinutrition® and have established a solid customer base across the city. The Warehouse Gym needs an effective database system to log members and to provide an effective way to store and hold data involved in the day-to-day activities of gym management. They would also like the ability to derive statistical data based on the information collected with a view to aid future decision making in terms of determining future staffing needs and the possibility of predicting future sales revenue.

Rationale

The rationale is to provide a fully functioning database for operation by the end users which are, a) the staff members (personal trainers) and b) gym management. The operation of the database is centred around staff members logging information through form interfaces for the purposes of tracking members, memberships, staff and classes and associated activities needed to run a real-life gym scenario and useful to management.

- The structure of the Warehouse Gym business allowed for various queries and reports to be completed to view and analyse data related to the operation of the gym.
- The organisation is within the leisure industry which typically consists of little to no IT systems for member management and as such, presents opportunities for growth regarding to rolling out our database to other companies.
- Our rationale behind the warehouse gym was that the database would provide all the elements needed to use our knowledge on creating databases. We began by looking up examples of gym databases online and researching the fundamentals of what a basic gym database would consist of. We came to the conclusion that a database has a couple of main entities such as a members table, a staff table and somewhere in the middle of these would be a classes table that both parties would be involved in. In our eyes this provided us with a good starting point in creating a database where the relationship between all entities would make sense and could be manipulated with queries, forms, entity relationship diagrams and basic design principles to create a functional and well-presented database.
- We also envisaged the possibility of added functionality.

List of Assumptions

When creating the database, we determined a number of assumptions in order for the database to make sense. The majority of these assumptions are within the relationship table where the entities are connected to one another via referential integrity.

In regards to the Gym table:

- There are two gyms; one in Oranmore and one in Mervue (Galway city)
- The Gym table has a one-to-many relationship with the Staff table — we can assume that one gym can have many staff members.
- There is a one- to-many relationship with the members table from the gym table — we can assume that one gym has many members.
- There is a one-to-one with the supplier table to the gym table — we can assume that one supplier supplies one gym at a time here.

In regards to the Members and Admissions table:

- There is a one-to-many relationship with the admissions table — we can assume that one member can have many admissions into the gym.

In regards to the PaymentMethods table:

- There is a one-to-many relationship between payments and the members table — we can assume that one payment method (e.g. paying by visa or cash) has many members using this type of payment.

In regards to Membership:

- There is a one-to-many relationship from the membership table to the members table — we can assume that one membership type (e.g. student membership) has many members.

In regards to the Classes and Equipment tables:

- There is a one-to-many relationship from classes to membership— we can assume that one class such as a yoga class can have many members attending it.
- There is a one-to-one relationship from classes to the equipment table — we can assume that one class can have the necessary equipment needed. (for example a weight training class can have dumbbells as the equipment needed.
- There is a one -to-many relationship from classes to staff — we can assume that one class can have many staff members present. (for example an aerobics class can have 2 personal trainers teaching a class.

In regards to the Staff, Payroll and Clocking System tables:

- There is a one-to-one relationship between staff and the payroll table — we can assume that one staff member has one payroll account to receive their salary.
- There is a one-to-many relationship from staff to clocking system — we can assume that one staff member can clock in and out multiple times in the clocking system.

Overview of Challenges Encountered and how they were Overcame

List of challenges encountered:

- Relationship issues — data integrity, primary and foreign keys, we encountered issues connecting the tables together in terms of deciding on primary and foreign keys to operate correctly.

- Creating forms — inserting sub queries based on other datasheets, pulling data from the right tables; functionality and connectivity.

- Connecting ID numbers in multiple tables to record data — such as a members and membership table issue.

- “Creditcard” table — initially, we could not create credit card details for each member. We resolved this by creating a “payment method” table instead which allowed us to choose a method of payment such as visa, PayPal, cash and so forth, rather than deciding to record credit card details for each new member.

- We encountered difficulties connecting multiple members in our gym database to a single membership. MS Access would automatically create a new membership when creating a member in the “add members” form. We designed the database to have a membership type as a single membership id, for example “student pay as you go” has the membership type of “84”, however when a new member was created and given this type of membership they would automatically be given a new membership id. We resolved this by deleting all members and changing a lot of the attributes around and moving fields over from the membership table to the members table such as “amount owed” and “amount paid”. We created new membership types and each type was given a membership id for example “3monthstudent” had the “membershipid” of 87, the fee was also given for each membership type, for example €100. After this we created all new members and as we created them we assigned them one of the “membershipID” numbers as a foreign key that were created in the membership table. When given a membershipID in members, the new members were than understood by MS Access as being a part of that membership type and the members began appearing on the dropdown in membership table for the type of membership they selected.

- A lot of testing was conducted during production to evaluate the connections in the database between records, forms, queries and objects such as buttons and macros which proved successful after multiple efforts and further regression testing. We also referred back to our final normal form schema design during the normalisation process, making amendments where necessary to facilitate further changes as per the waterfall development model adopted in the development of our database.

Overview of how the Warehouse Gym Database is designed in line with the 5 Characteristics of Good Databases

Data Integrity

Data integrity refers to accuracy and consistency of data in your database. When a database is being created, attention needs to be given to data integrity and how you would go about maintaining it. Good databases will enforce data integrity whenever possible. Data integrity can be used to prevent unwanted circumstances occurring. An example could be a database user trying to enter a phone number into a date field. If the database enforces data integrity, then it will not allow the user to make this entry as it is a mistake.

Keeping data integrity maintained means making sure that data remains intact and unchanged throughout the entire life cycle of the data. This process can include capturing the data, storage of the data, updates to the data, transfers of the data, backups of the data and so on. Any time that data processed then there is a risk that the data could get corrupted either maliciously or accidentally.

There are many examples of situations which could lead to data integrity being at risk. Some of these include:

1. A user trying to enter a date outside an acceptable range.
2. A user trying to enter a phone number in the wrong format
3. A bug in an application may attempt to delete the incorrect record.
4. When a developer is transferring data between two databases, they may accidentally try to insert data into the incorrect table
5. When a developer is moving data between two databases, the network may go down.
6. A user may attempt to delete a record in a particular table, but another table in the database is referencing that record and relies on the relationship
7. A user may try to update a primary key value when there is already a foreign key in a related table which points to that particular value.
8. A developer may forget that they are working on a production system and start entering test data directly into the database.
9. A hacker may manage to steal all user passwords from the database.
10. A hacker may manage to hack into the network and delete the database and all its contents.
11. A fire could happen in the building, burning the computer which stored the database.
12. The backup of the database has failed for the last few months.

So it is obvious to see that there are many scenarios where data integrity could be at risk. The vast majority of these risks can be dealt with inside the database itself using data types and constraints for each column or encryption. Other risks can be addressed using features of the database management system such as regular backups and frequently testing that these backups actually restore the way that you expect. Other methods of addressing these risks include an offsite backup location, a properly functioning IT network, proper employee training, good security policies and so forth. Restricting the field size and character type also was implemented in this database which prevents SQL injection and protects the integrity of the database.

When creating the database tables, we carefully assigned each new field name to the correct corresponding data type. For example all primary keys are set to “autonumber” and all foreign keys are set to “number”.
Within the Gym Database we implemented this within the respective fields, i.e. date fields were set in the format of dd/mm/yy and the data type and also long time was set when required to log times for effective calculations and aggregates further on in fields such as total hours worked, this type was set to long time to enable the sub-calculation of the times. Fields such as amount paid were set to currency fields.
When creating the relationships within the Entity relationship Diagram, Referential Integrity was enforced between the primary and foreign keys of the tables to keep the data secure and in the same format.

Data Independence:

Data Independence is a fundamental part of a database that allows you to change a database schema at one level of a database system without the need to change the schema of the system at the next higher level. Its purpose is to separate the data from all other programs that make use of it. This separated or independent data can be used for computing and presentations. In many systems, data independence is an essential function carried out in a database. Its main purpose is to improve the quality of the data.

In a database there are two types of data independence: physical data independence and logical data independence

Physical Data Independence

A database has three levels physical, conceptual and external. The physical data independence is concerned with separated the conceptual level with the physical level, this allows a logical description of the database without specifying any physical structures. Physical data independence is easy to achieve an allows for easy change to any physical data storage without any effects to the schema.
The following are examples of changes that will not be affected under physical data independence.

- Changing to a new storage device like a hard drive.
- Updating the file organisation technique in the database.
- Switching to a new data structure.
- Modifying indexes.
- Changing the location of the database for a C drive to a D drive.

A physical data independence is mainly concerned with the storage of the data, is easily retrievable, easy to achieve, and is concerned with the internal schema.

Logical Data Independence:

Logical data independence is concerned with changing the conceptual scheme but not affecting either the external view or external programs. The changes made are picked up by the navigation between the external and conceptual levels. In comparison to physical data independence, logical data independence is challenging to achieve
The following are examples of changes under logical data independence that will not affect the external layer.

- Adding, updating, deleting a new attribute, entity or relationship is possible without rewriting the already existing application programs.

- The ability to merge two records into one record.
- The ability to break an existing record into two or more records.

Logical data independence is mainly concerned with changing the data definition or the structure. The data is mainly dependent on the logical structure of data making it difficult to retrieve the data. When new fields are added or deleted from the database, changes to the app program are necessary. Logical data independence is concerned with the conceptual schema such as adding, deleting and updating a new attribute.

In regards to our Physical data independence, the database will ideally be held between our two gyms, we will store the backend on the cloud whereas the front end will be available to both gyms on the main pc. We will roll this out on physical implementation of the project.
In regards to our Logical data independence, this is carried out through referential integrity and our normalization processes.

Avoiding Data Redundancy

Data redundancy is an issue that causes storing of redundant or excess amount of data in a database. It means unnecessary duplication of data. Data can be duplication in a number of ways. An example could be storing customer data in two different computers. Another example could be having employee mobile number stored in the employee personal data table, employee department data table and employee salary data table. Data redundancy will cause inconsistency in your database. If you need to change the mobile number of an employee, then you have to change it in three locations. If you happen to miss any of those locations, then you are not sure about that employee's mobile number.
It defeats the purpose of a relational database design to have unnecessary duplication. The data relationships present in a relational database should allow you to maintain a single data field, at one location, and then make the database's relational model responsible to port any changes, to that data field, across the database. If redundant data is present, then it is a waste of valuable space and it will introduce database maintenance issues.

For data redundancy to be eliminated from our database special care must be taken to organise our data into our relevant tables. Normalisation is the method used to organise data in such a way that it is prevented from causing redundancy. Normalisation involves establishing and maintaining the integrity of our data tables in addition to eliminating inconsistent data dependencies.

In order to establish and maintain integrity it is necessary for parent-child table relationships to be maintained. Eliminating inconsistent, data dependencies involves making sure that our data is housed in an appropriate access table in which the data has some relation to or dependence on the table.
Normalisation involves adhering to established rules by the database community to ensure our data is organised efficiently. The rules are referred to as normal forms. Normalisation means that you may be required to include extra tables in your access database. Normal form goes from 1 to 2 to 3 in most cases. The rules of normal form are cumulative in that the rules of 2nd normal form are inclusive of the rules of 1st. The rules of 3rd normal form are then inclusive of both 1st and 2nd normal forms.

1st normal form: Avoid storing similar data in multiple table fields.

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key.

2nd normal form: Records should be dependent, only, upon a table's primary key(s)

- Create separate tables for sets of values that apply to multiple records.
- Relate these tables with a foreign key.

3rd normal form: Record fields should be part of the record's key

- Eliminate fields that do not depend on the key.

To implement normalization in our database we reduced and cut all repeating fields that were unnecessary and redundant. We implemented this from the very beginning during the design process.

(Consumer-rankings.com, 2019)

Data Security:

Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Data security also protects data from corruption. Examples of data security technologies include backups, data masking and data erasure. A key data security technology measure is encryption, where digital data, software/hardware, and hard drives are encrypted and therefore rendered unreadable to unauthorized users and hackers.

For data security to be implemented effectively a number of steps and procedures must be followed:

- Secure authentication to the database is used.
- Only authorized users have access to the database.
- Users are granted the minimal permissions necessary for their job function in the database. Permissions are managed through roles or groups, and not by direct grants to user IDs where possible. Different categories for users of the database such as manager and clerical officer and so on.
- Strong passwords in the database are enforced when technically possible, and database passwords are encrypted when stored in the database or transmitted over the network.
- Applications require individual database login/password and roles/grants when possible.
- Non-DBA accounts do not allow the granting of roles or permissions in any environment with restricted data.
- Database accounts are locked after at most six failed logins.

To secure our database we created a login form that only the three administrators and one staff member can log into. The staff login form does not have an admin view of the database. We restricted the number of characters in the login form which controls access to the database, we done this by limiting the field size to15 characters.

Data Maintenance

Database Maintenance is a term we use to describe a set of tasks that are all run with the intention to improve your database. There are routines meant to help performance, free up disk space, check for data errors, check for hardware faults, update internal statistics, and many other obscure (but important) things. A poorly maintained or unmaintained database can adversely affect application performance and can even lead to downtime or data loss. The only way to mitigate the risk of such a problem is to conduct effective maintaining of your database. Regular maintenance is essential for the smooth, successful operation of your databases. This maintenance includes making backups to protect against data loss, performing consistency and integrity checks to ensure your data and indexes aren't corrupted, and regularly rebuilding indexes to help you compact and defragment your data. When performing Data Maintenance, you should run Table Analysis testing. This would involve checking the following:

- Missing tables.
- Missing or extra columns.
- Wrong column order, data type, length, or defaults.
- Wrong numeric precision.
- Missing primary keys.
- Missing or disabled/distrusted foreign keys.

We carefully went through our database and began analysing each field in the tables to ensure that referential integrity was applied and no data was missing or corrupt. We verified data types and objects in each table also. We used the compact and repair tool to help prevent and correct database file problems.

Microsoft OneNote Project Log

During the production of the database, KNM Consultants kept a log on the weekly activities on Microsoft OneNote the link to which is below.

https://galwaymayoinstitute-my.sharepoint.com/:u:/r/personal/g00351238_gmit_ie/Documents/Application%20Development%20Assignment%20(80%25)%20-%20SHARED%20FOLDER/Link%20to%20OneNote.url?csf=1&e=hw6ER5

https://galwaymayoinstitute-my.sharepoint.com/:o:/r/personal/g00351238_gmit_ie/_layouts/15/WopiFrame.aspx?sourcedoc=%7B7a8dff49-2736-41bb-a372-f67ee0be1957%7D&action=default&originalPath=aHR0cHM6Ly9nYWx3YXItYXIvaW5zdGl0dXRlLW15LnNoYXJlcG9pbnQuY29tLzpvOi9nL3BlcnNvbmFsL2cwMDM1MTIzOF9nbWl0X2llL0VrbI9qWG8ySjd0Qm8zTDJmdUMtR1ZjQkZFZmNnS01XYTNiY09uNW51cVh1Rnc_cnRpbWU9Y2w3akZpVnUxMGc

Bibliography:

Consumer-rankings.com. (2019). *Data Redundancy – Its Effects And How To Avoid Them | Consumer-Rankings.com*. [online] Available at: https://www.consumer-rankings.com/blog/data-redundancy-its-effects-and-how-to-avoid-them/ [Accessed 21 Nov. 2019].

Guru99.com. (2019). *Data Independence in DBMS: Physical & Logical with Examples*. [online] Available at: https://www.guru99.com/dbms-data-independence.html [Accessed 18 Nov. 2019].

databasedev.co.uk, 2015. Data Redundancy Defined. [Online]
Available at: http://www.databasedev.co.uk/data-redundancy.html
[Accessed 18 11 2019].

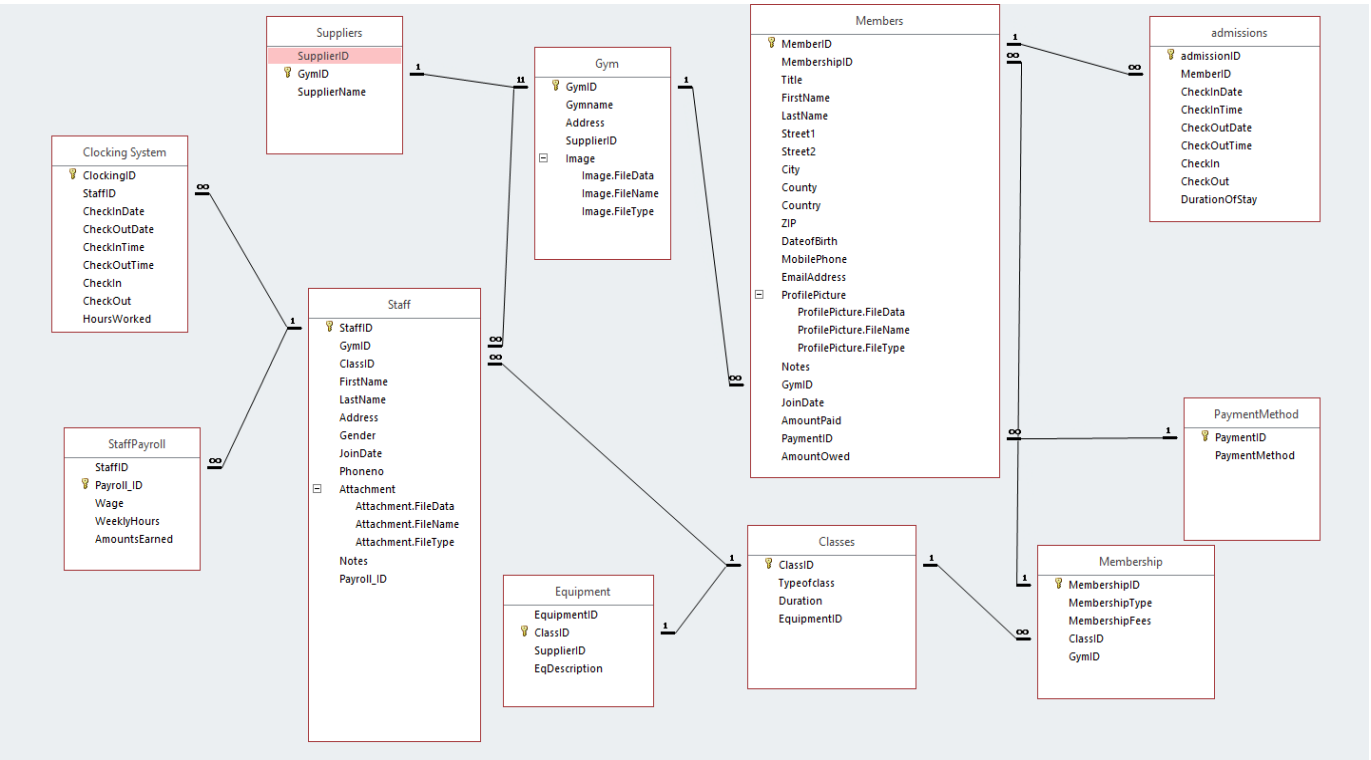
Technical Report

A step by step instruction manual of the creation process

After completing the normalisation process, we began to design the database schema by creating the tables.

The tables included are the gym, suppliers, members, membership, payment method, admissions, classes, equipment, staff, staffpayroll, clocking system.

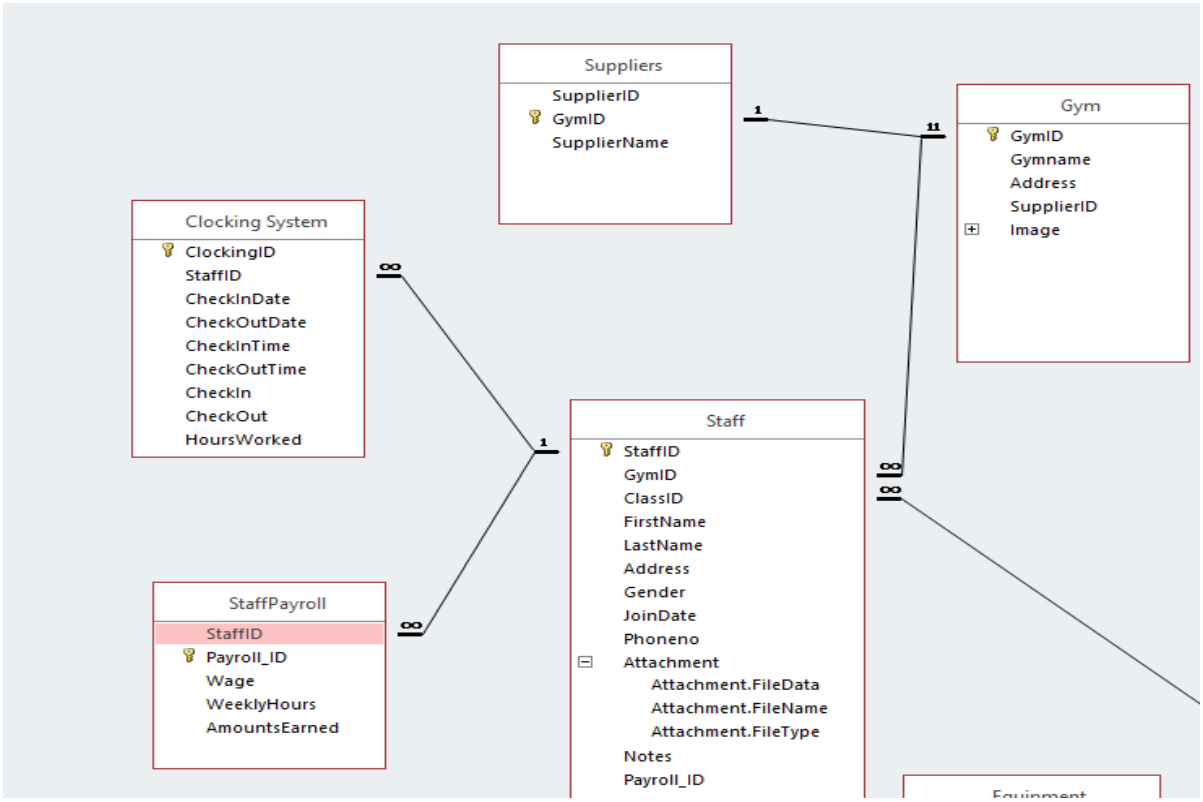
- Each table has the related ID to the table followed by the necessary fields.



<u>The Gym table:</u>

We created the table, and filled it with fields as follows: GymId(primary key), Gymname, Address, SupplierID(foreign key) and Image. The gym table is used to connect with the suppliers, staff and members table and played a big role in connecting the tables together. This table was seen as the centre point of the database.

Gym and Supplier table's:



Following through from the guidance document, the gym table contains the data from both gyms — one in Mervue and one in Oranmore with each gym connected to a different supplier; Mervue gym and Mervue supplier, and Oranmore gym and Oranmore supplier. The supplier table consisted of SupplierID, GymID(primary key) and the SupplierName. The tables are connected by the gymID being the primary key in both, thus they are connected by a one to one relationship, ie. One gym, one supplier.

Gym table and Staff table:

The gym is connected to the staff table by GymID as a foreign key in the staff table, thus creating a one to many relationship, meaning that one gym has many staff. The staff table consists of the following fields: StaffID(primary key), ClockingID(foreign key),GymID(foreign key), ClassID(foreign key), firstname, lastname, address, gender, joindate, phoneno, attachment(image) and notes.The staff table is there to display the personal details of our staff.

Staff table and clocking system table:

Staff table is connected to the clocking system by staffID as a foreign key in the clocking system, creating a one to many relationship from staff to clocking system. Clocking system consists of: clocking system(primary key) StaffID (foreign key) checkindate, checkoutdate, checkintime, checkout time,(placeholders) checkin ,checkout (macros) and hoursworked(a calcuated field).

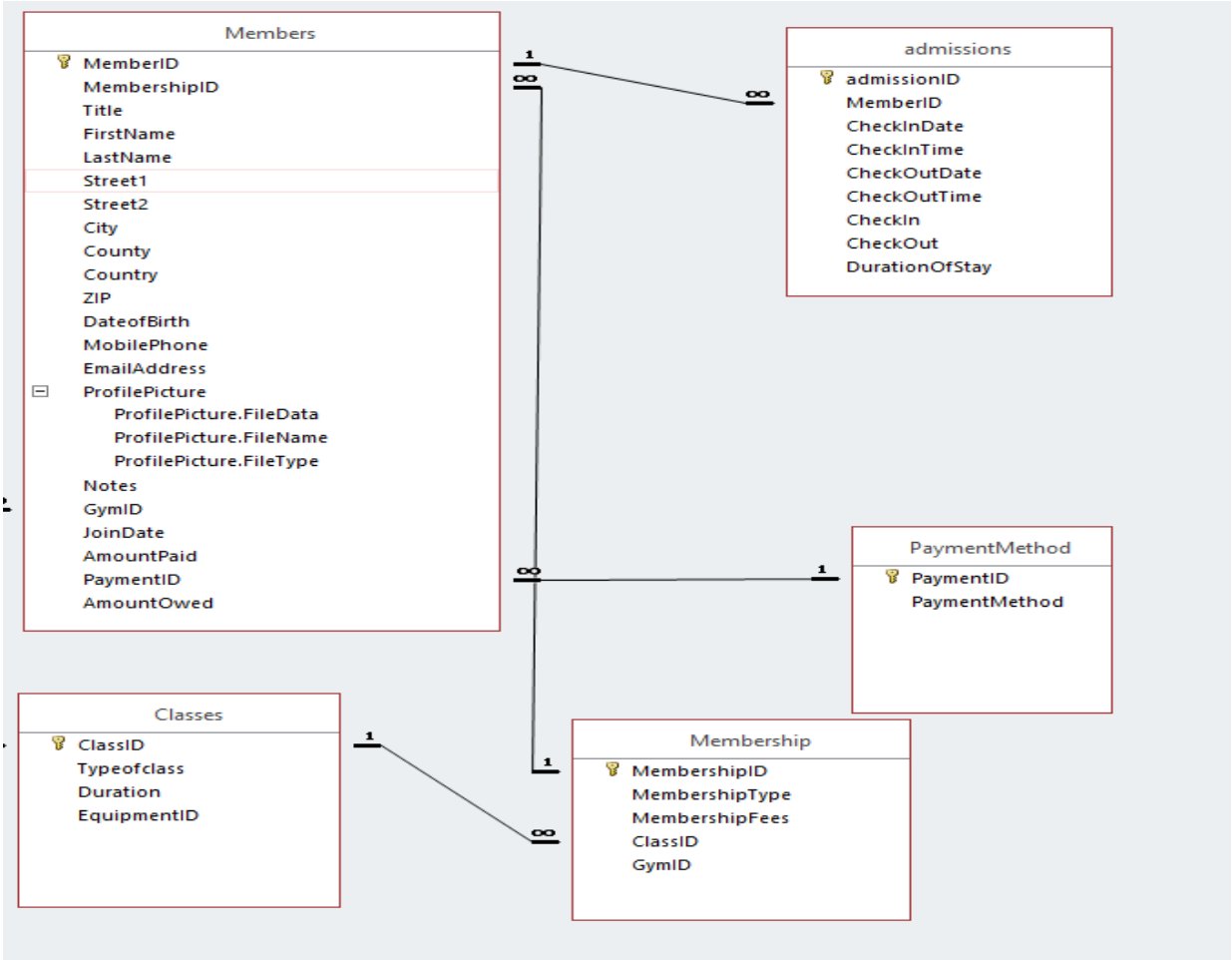
The check in date and times were created to record the date and time of check ins and outs, these were inserted into their respective text boxes by macros based in the check in and check out option set which inserted the current date/time as appropriate.

Adding this functionality increased the level of complexity of the database enabling the collection of data and allowing business intelligence and predictive analysis to be conducted by management on the data set. Management could now make decisions based on measuring and analysing previous customer and staff data in relation to hours worked time spend in the gym by members and hours worked by staff.

Staff table and payroll table:
The staff table and the staffpayroll are connected by the staffID being a foreign key in the staff payroll table, this means one staff has multiple payslips throughout the year(weekly payroll). The payroll table calculates each employess wages earned, it does this by a calculation in the “amountearned” field where wage is multipied by weekly hours to give the amount earned for the week. When the sub data sheet is selected you can see that payroll of each staff member.

As discussed previously, the gym table is the main starting point for the tree of tables. When looking at the relationship the gym branches off two ways, one towards the staff and one towards the members. The gym is situated in the middle of the ERD for this reason.

The gym table and the members table:
The members table was created to hold all information related to the members, much like the staff table. The following fields are included in members: MemberID, MembershipID, title, firstname, last name, street 1 & 2, city, county, country, ZIP, date of birth, mobilephone, emailaddress, profile picture(an attachment), notes, GymID(foreign key), joindate, amountpaid, paymentID(foreign key) and amountowed. The gym table is connected to members by gymID as a foreign key in members- creating a one to many relationship ie. One gym has many members. The sub data sheet in the gym table shows all members for each gym. (as displayed on the graphic on the next page).



Members table and admissions table:

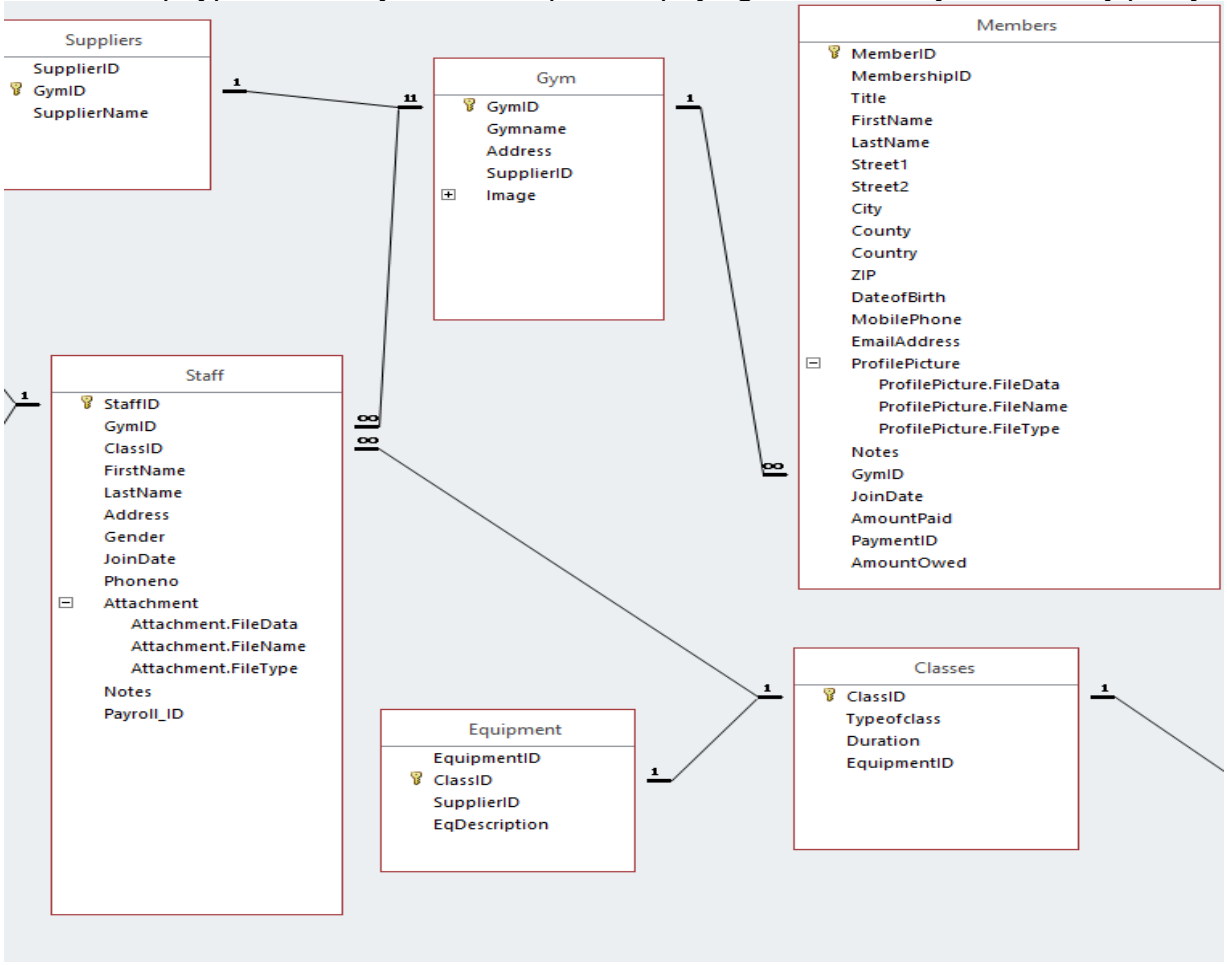
These two tables are connected similarly to the staff and clocking system tables. They are connected by membersID being the foreign key in the admissions table thus creating a one to many relationship, as in one member has many admissions. The admissions table has the following: admissionID(primarykey), MemberID(foreign key), CheckinDate, CheckinTime, CheckoutDate, CHeckoutTime, Checkin, CheckOut, DurationofStay. The admissions table is there to record member check ins and check outs from the managers and staff members point of view. It was created to be used as a sub form much like the clocking system for staff, so that when we scroll through our members in the “landing page” form we can see all the members that have clocked in and out. This allows us to evaluate our gym database to see take note on the busy hours or who logs in and out the most frequent. When you select the sub data sheet in members beside a field, you can see all admissions the members have had.

Members table and payment method table:

The members table is connected to the paymentmethod table by method id being the primary key in payment methods and a foreign key in members, thus creating a one to many from payments to members. Payment methods is a simple table, it consists of the primary key paymentID and also a field called payment method. The field allows members to choose a method of payment such as visadebit, cash or voucher based on the paymentID. Once the paymentID is placed in the members table, the method of payment is chosen. When the sub data sheet icon is selected you can see all the members that have selected each payment method.

Members table and Membership table:

The members table is connected to the membership table by membershipID being a primary key in the membership table and a foreign key in the members table. The membership table consists of: MembershipID(primary key), membershipType, membershipfees,classID(foreign key) and GymID. The membership table gave our members a range of memberships to choose from and the price of each. When clicking on the sub data sheet icon you can see all the members inside each membership type. This was designed for forms where creating a new member was shown the membership types and they have the option of paying all the money now or only partly.



Classes with staff and members table's:

The classes table was the connecting table between the staff and the memberships tables. The class table consists of: ClassID(primary key), typeofclass, duration, and equipmentID. The ClassID was put as a foreign key in both the staff and membership tables. The membership chosen by the member will offer a certain class with it, for example the "Yoga pay as you go" in the memberships table has "1" as the classID. In the class table "1" is "Yoga" in the type of class field. The staff are also assigned ClassID as a foreign key in the staff table as staff members are instructors at these classes. One class can have many staff members. One class can have many memberships, ie. Members present. If you select the sub data sheet in classes, you can see the instructor(staff) assigned to the given class.

Classes and equipment table's:

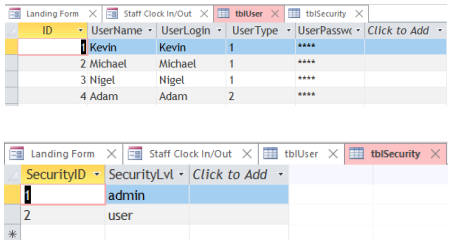
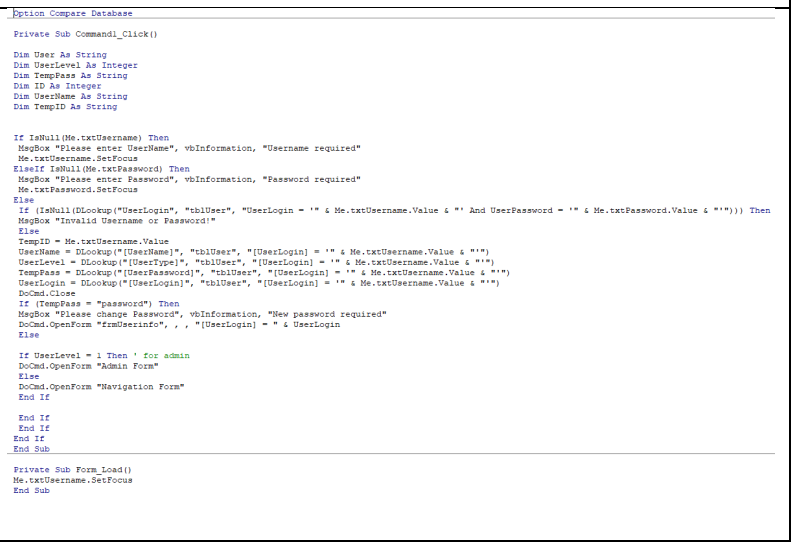
The classes table is also connected to the equipment table by a one to one relationship. The equipment table consists of EquipmentID, ClassID(primary key), SupplierID, and EqDescription. In the equipment table the equipmentID relates to a piece of equipment.In the class table the classID relates to a specific class given by staff. For example the classID of "2" in the class table is "weightraining" in the type of class field, and this is also given EquipmentID of "2". In the Equipment table, "2" under equipmentID is given the eq description of "dumbells" and the classID of "2". In short, the equipment is assigned to the related class through the ID numbers.

Steps taken during the Design process

Forms

Login Form (#1)	


Figure 1 — Login Form
Fig. 1 Login Form
	• Created as a popup form set to display on database entry. Settings first controlled by the creation of the following two tables below and some VBA code.
	  ○ tblSecurity
	  ○ tblUser
	• Created using VBA code
	• Sends the User to either of the below forms depending on if the user is a Admin or a User.
	  ○ Admin Form
	  ○ Navigation Form

 *Figure 2, 3 & 4 — Controls for Login Form. (tables and VBA code)*	

Employee Navigation Form (#2)

Figure 5 — Employee Navigation From

- Simple form based on buttons designed as the main navigation form for User's while also accessible by Admin's.
- Objective is to links forms through buttons via Macros making it easy for the user to input data without needing to navigate around through the tables.
- Designed to be the main control panel for Users (Staff).

Admin Form (#3)

Figure 6 — Admin Form

- "Landing page" a simple type of dashboard for Admin users based on a blank form containing our design elements related to the Warehouse Gym theme.
- Displays the latest graphical results of the database relating to the queries shown as charts. Queries were created separately.
- Buttons linked to pages of interest using Macros.

Staff Clock In/Out Form (#4)

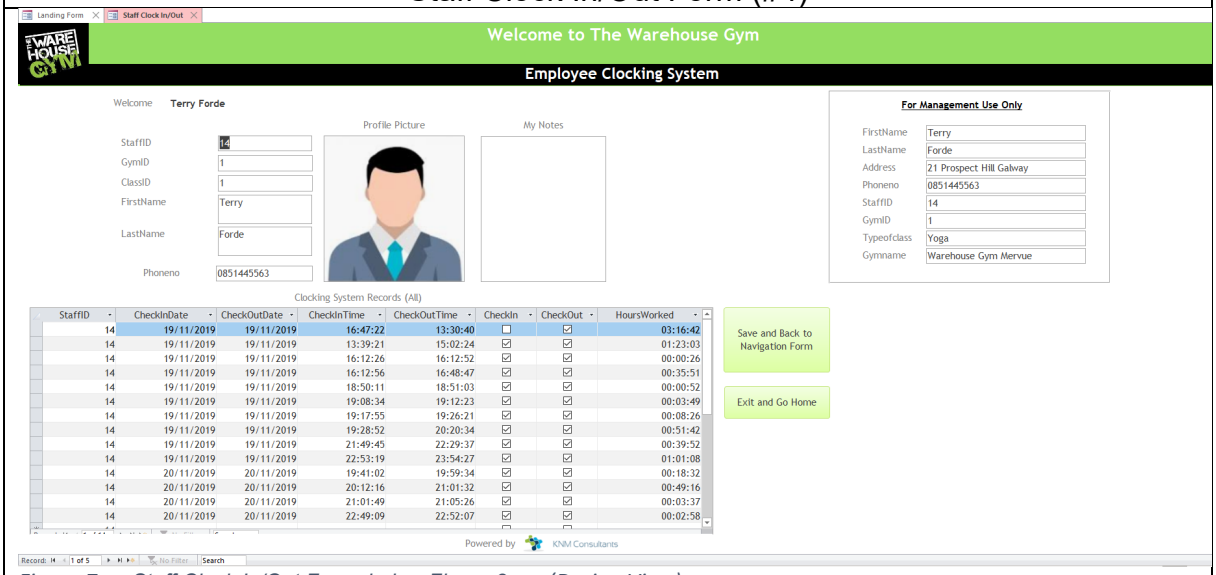
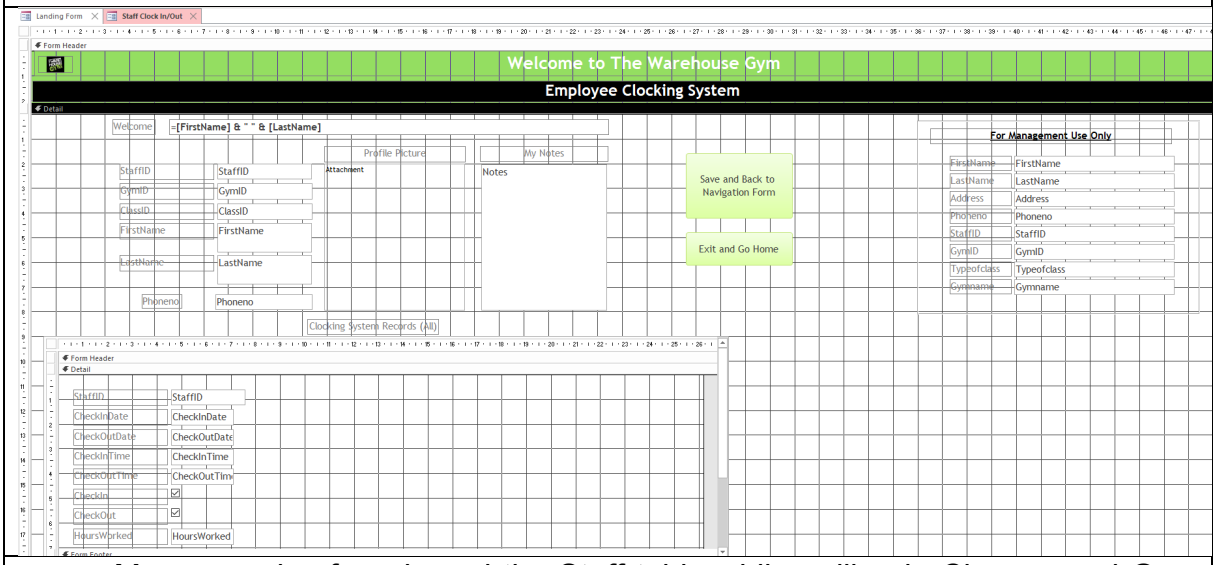


Figure 7 — Staff Clock In/Out Form, below Figure 8 — (Design View)



- More complex form based the Staff table while pulling in Classes and Gym through SQL – Required to pull all the data from fields located throughout the database.
- Expression Builder used to dynamically pull in the Staff member name from the Staff table.
- Subform displayed below based on the Clocking System subform which enables access to the table required to input the relevant clock in information and access the input controls.

Landing Form (#5)

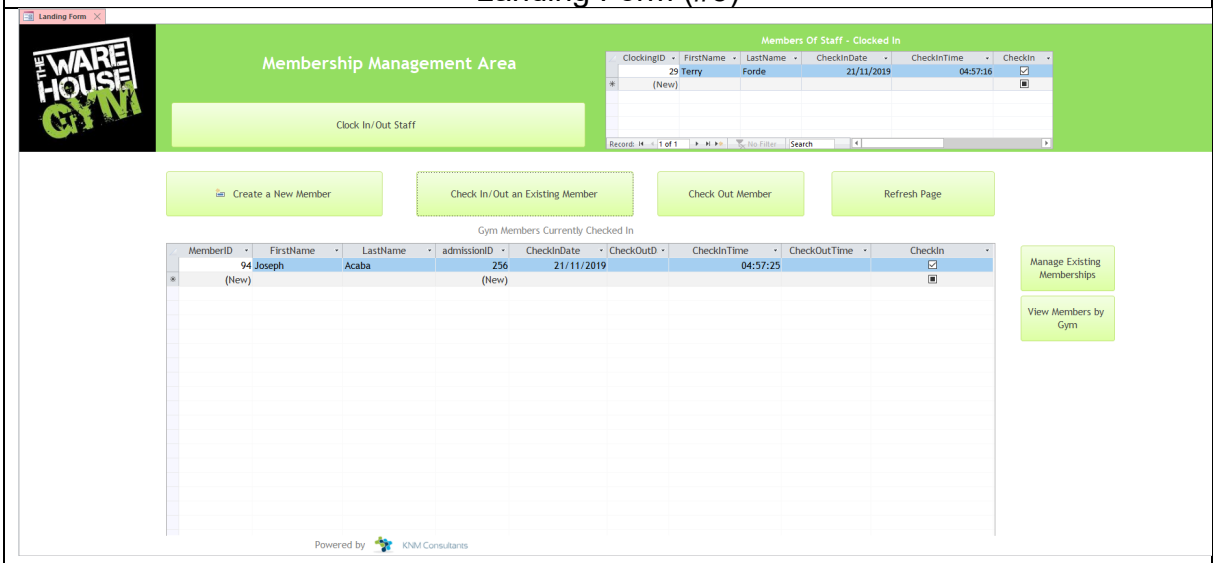
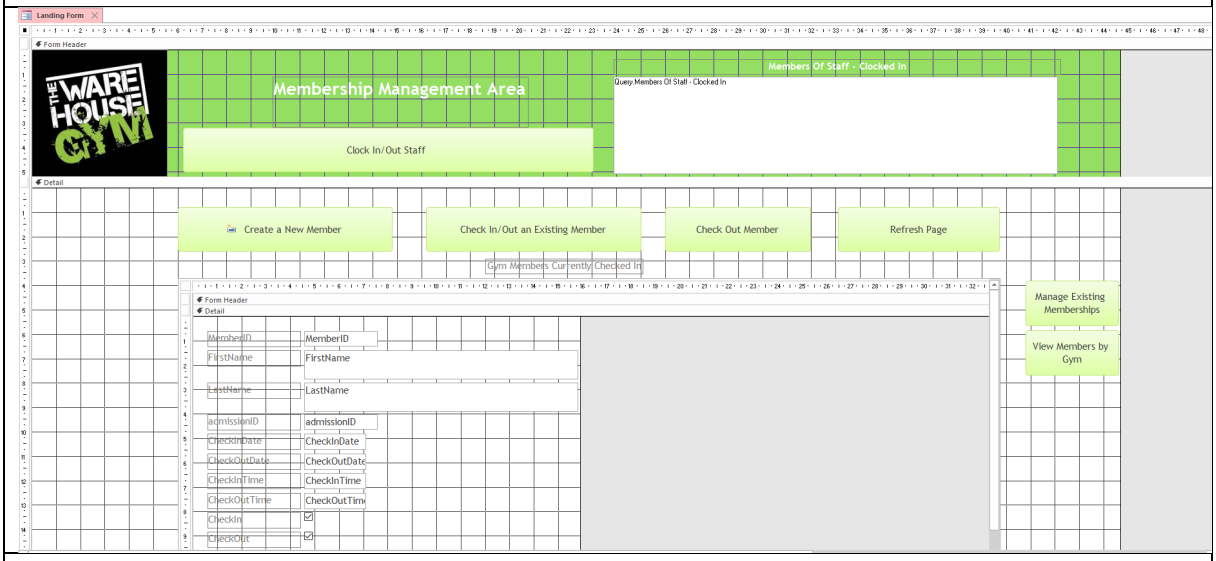


Figure 9 – Member Management Area (Form) below Figure 10 (design view)



- Membership Management Area form is designed to be the complete control panel for the member of staff.
- Pulls in query to display the staff members checked in.
- Contains subform based on an SQL query on the Members currently checked in and also a separate query for the Staff members currently checked in.
- Buttons containing macros to refresh page or navigate to other connected forms relevant to the day to day needs of a staff member.

Add Member Form (#6)

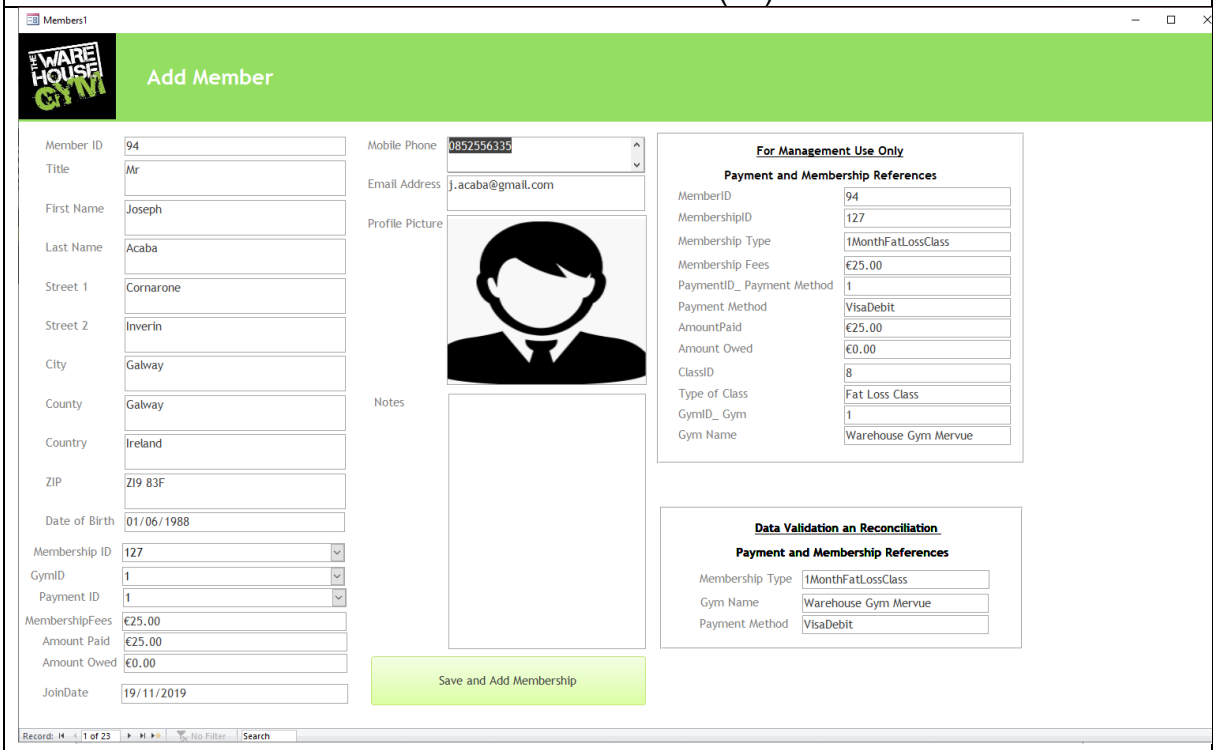
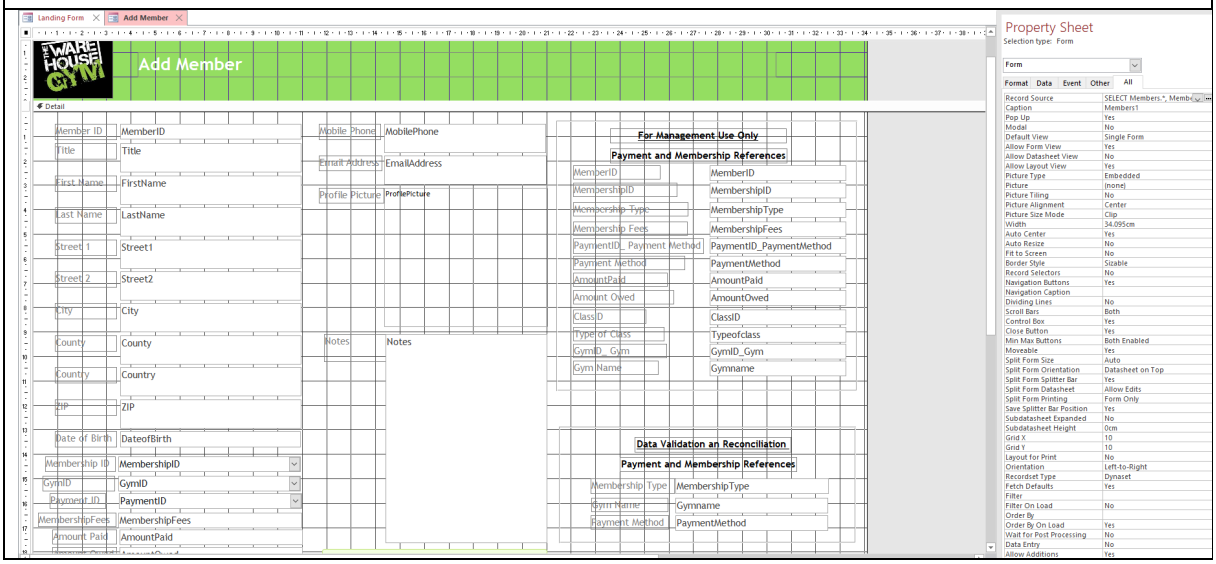
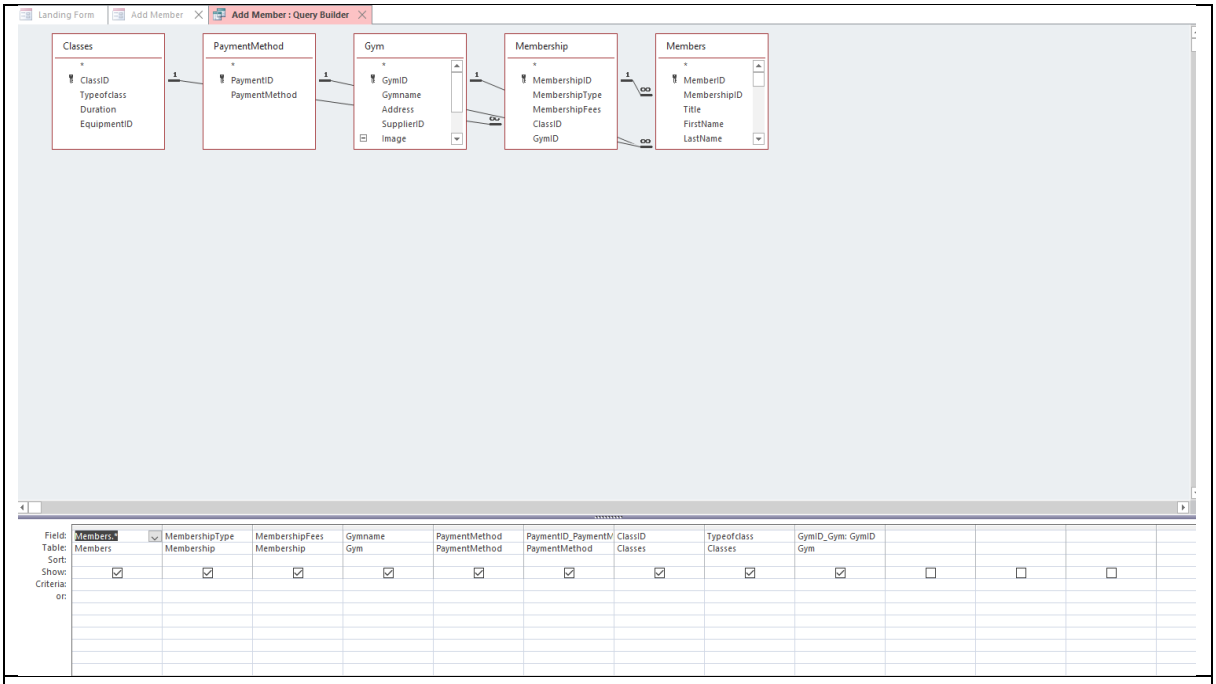


Figure 11 Add Member Form, Figure 12 (design view below) and Figure 13 SQL Query





For Management Use Only

Payment and Membership Reference

MemberID	94
MembershipID	127
Membership Type	1MonthFatLossClass
Membership Fees	€25.00
PaymentID_ Payment Method	1
Payment Method	VisaDebit
AmountPaid	€25.00
Amount Owed	€0.00
ClassID	8
Type of Class	Fat Loss Class
GymID_ Gym	1
Gym Name	Warehouse Gym Mervue

Figure 14 – Management Use Only Panel on Some Forms

- Form based on the members table with SQL bringing in associated fields for the purposes of data validation and reconciliation both for Admin's and Managers as well as having a members information panel open to display to all Staff.
- Management use only areas in forms where text boxes remain disabled and locked and used for the purposes explained above, also prevents staff confusion and possible incorrect data entry while remaining useful as reference data.

Check In/Out Member Form (#7)

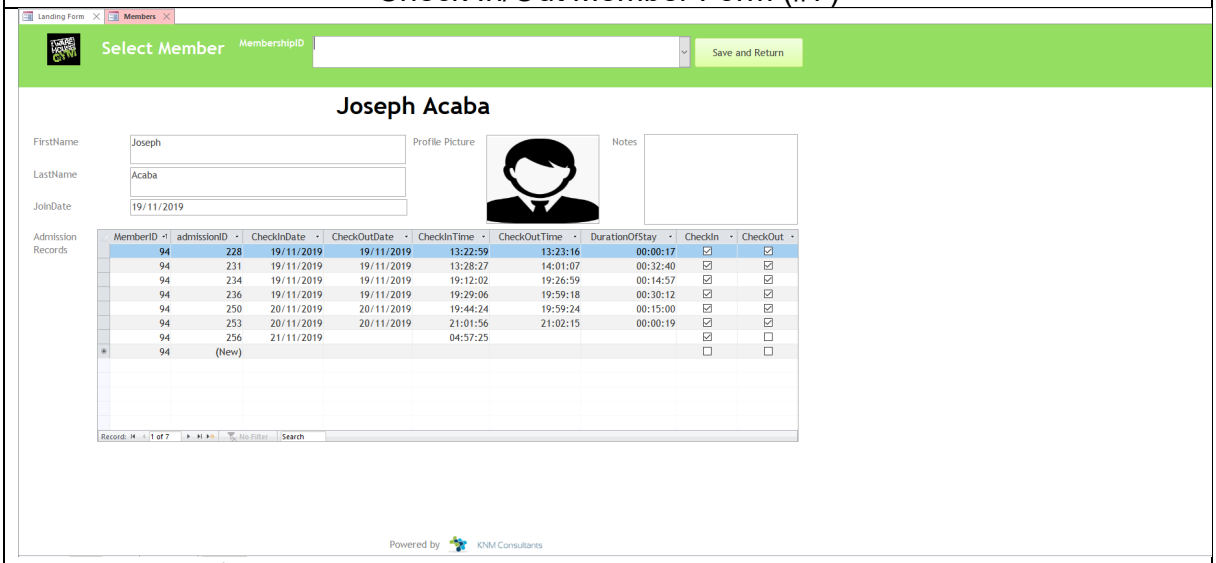
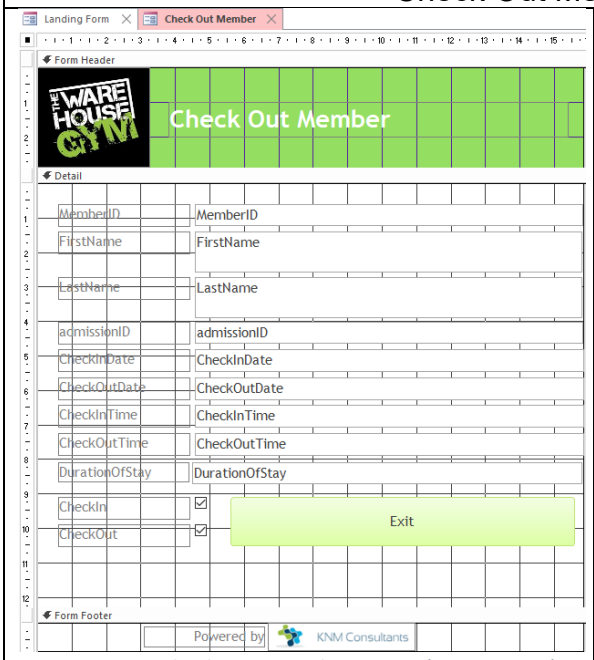


Figure 15 – Check In/Out Member Form

- Simple form based on the Members table and Admissions table. The subform allows the check box's to be 'checked' to True, enabling the query in the subsequent forms below which are based on the Check in = True, Date() which means members checked in and only those of which are checked in today.

Check Out Member Form (#8)



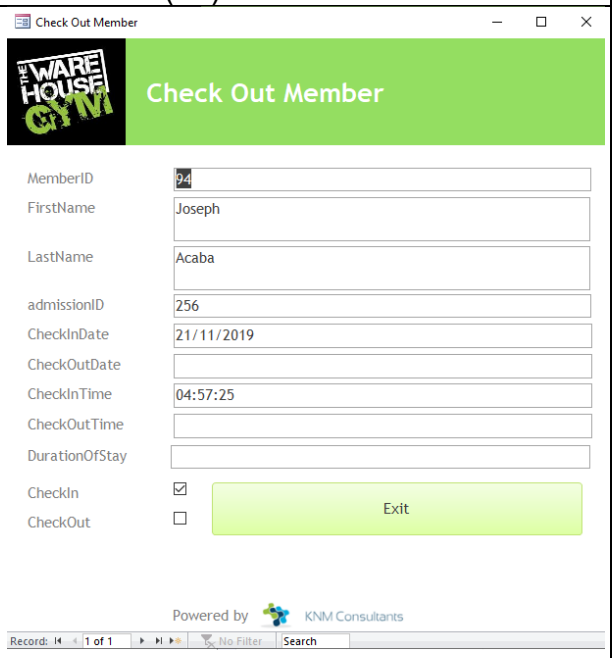
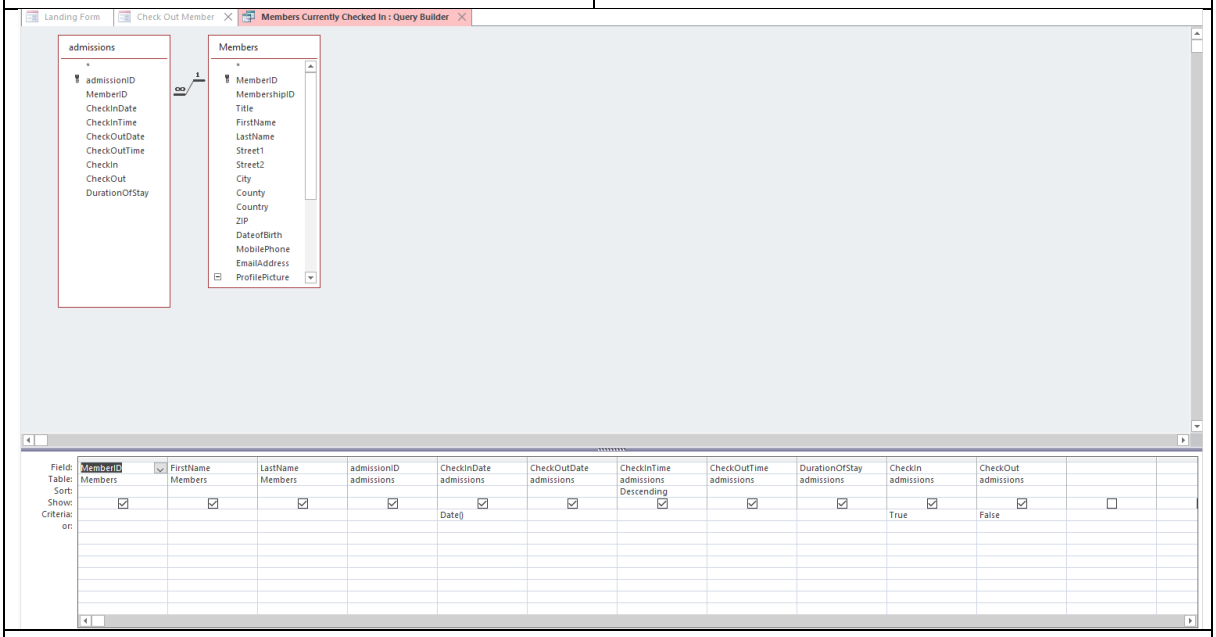
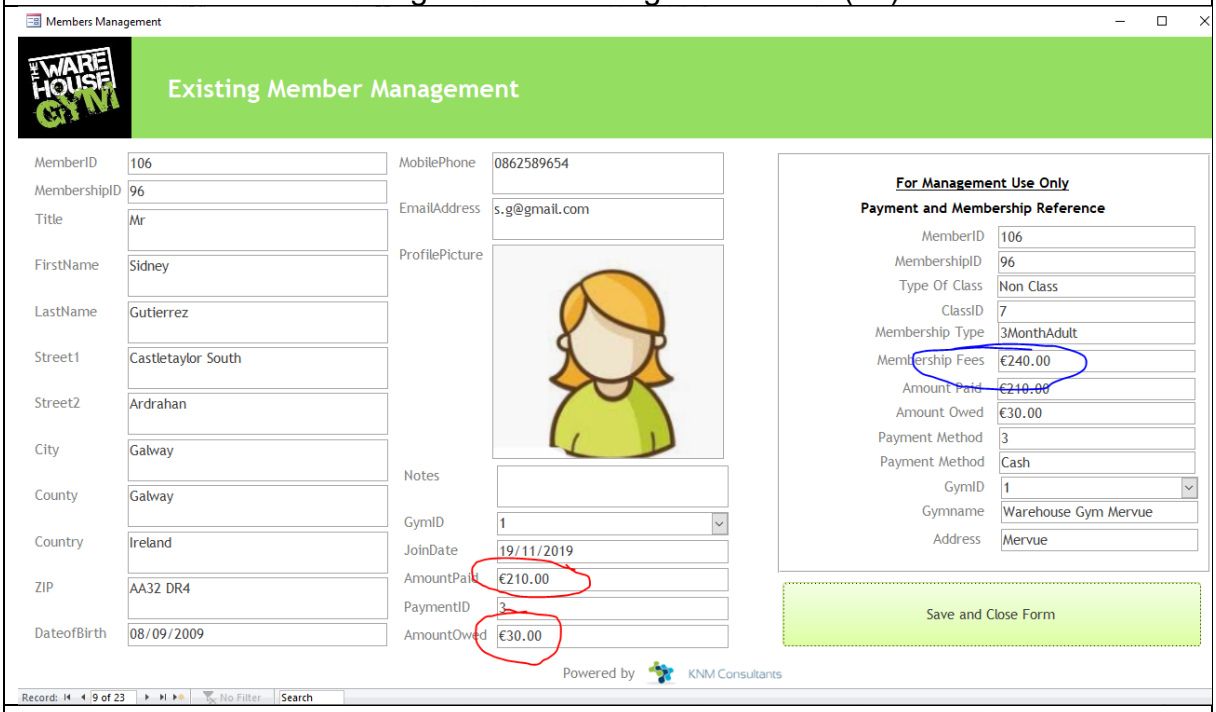


Figure 15-7 – Check Out Member Form, (Design view) & SQL Query View (below)



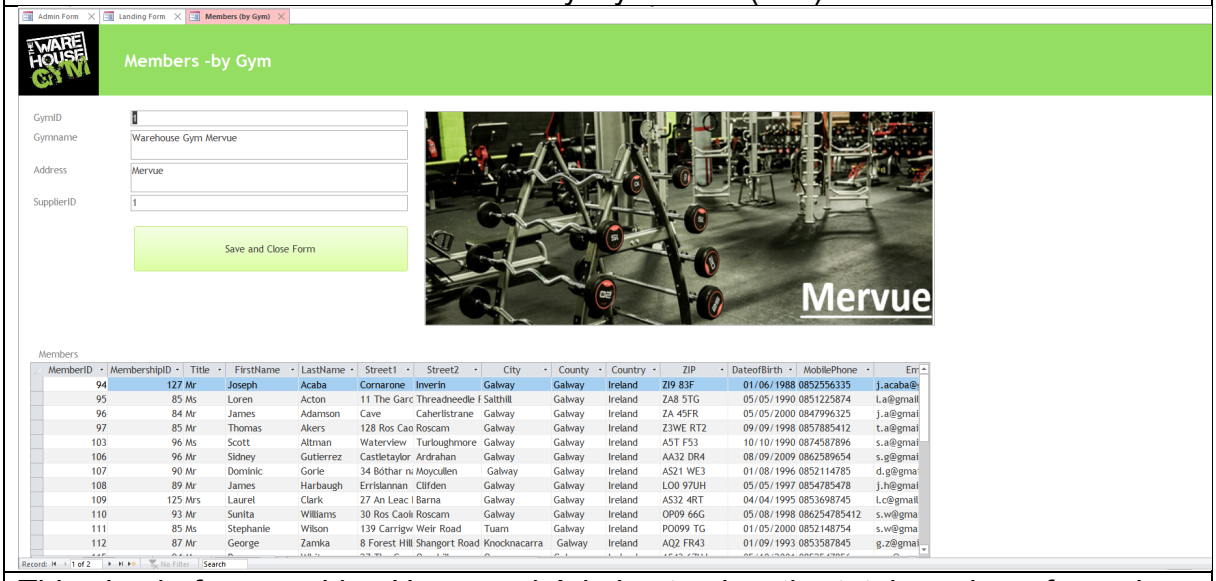
- Checkout member form explaining the SQL query contained within the subform which enables the functionality without a user needing to access tables or manually input data.

Existing Member Management Form (#9)



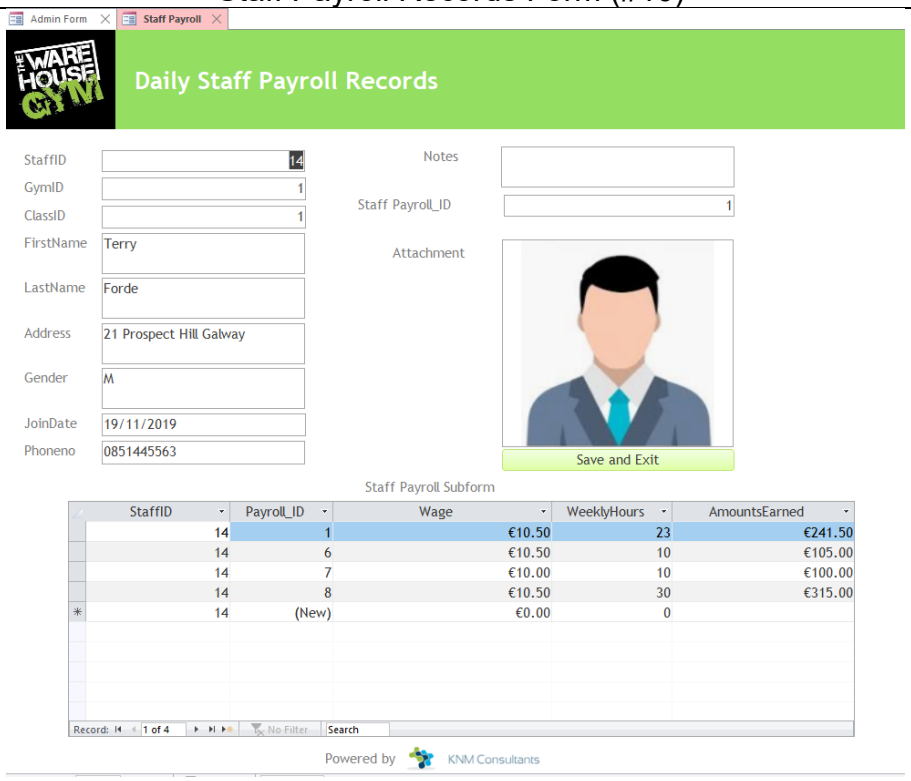
- Existing Member Management Form enables the Staff or Admin user to modify existing Member profiles in cases where personal details need to be amended or if the member wishes to pay the remainder of the Membership Fees. In this case the Staff member manually computes the difference between the red encircled figures above relating to the amount the member wishes to 'pay off'. The Staff member inputs the correct calculations and the record should be saved and updated.

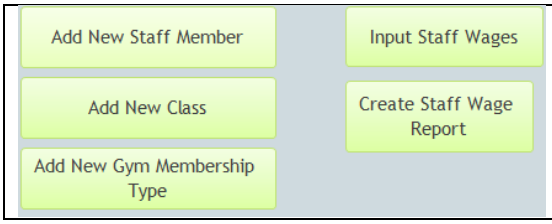
Members -by Gym Form (#10)



This simple form enables Users and Admins to view the total number of members per gym from a form based on the Gym table and datasheet sub form based on members.

Staff Payroll Records Form (#10)





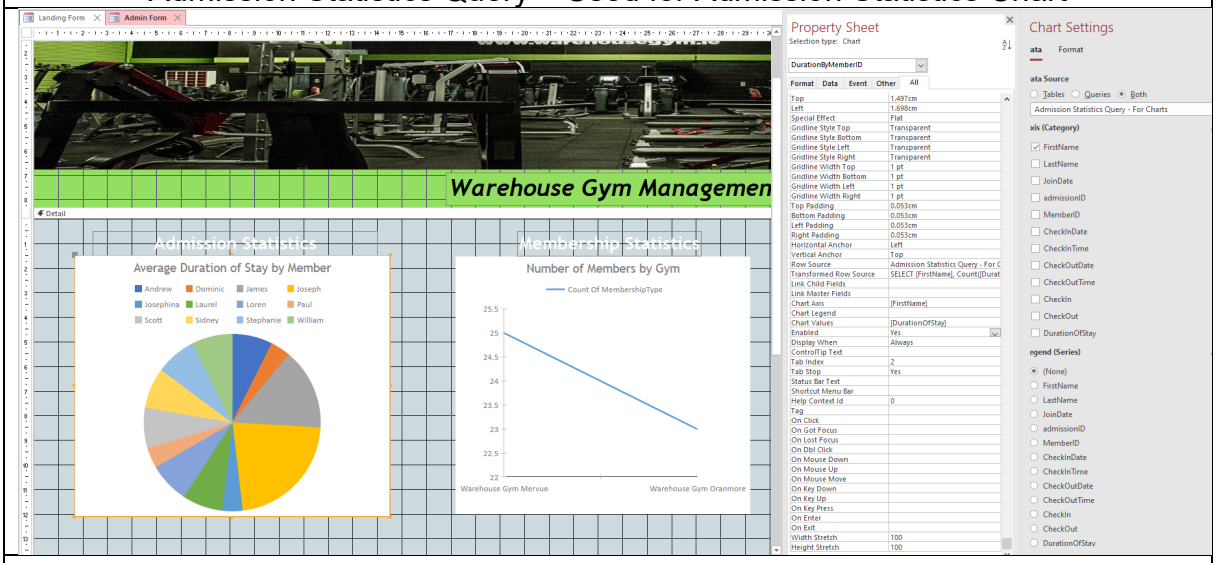
Additional Forms (4) contained within the Admin panel designed for use by Management only, allows creation of new records and report generation.

Queries

Dozens of queries were created across the forms to allow for the input and output of data across the database. Some examples of the more complex queries are below.

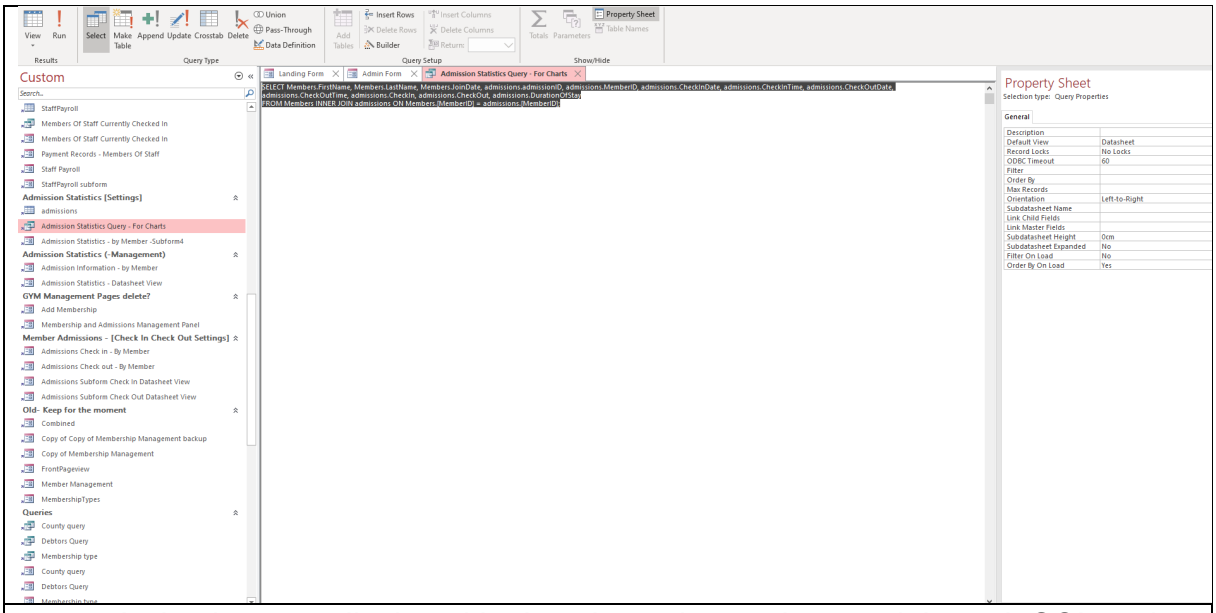
4 such queries were required to be created in order to output the following Reports: Gym Members Query, Members Admissions Query, Members County Query, Membership Type Query. These were created by a simple query wizard.

Admission Statistics Query – Used for Admission Statistics Chart



Landing Form | Admin Form | Admission Statistics Query - For Charts

FirstName	LastName	JoinDate	admissionID	MemberID	CheckInDate	CheckInTime	CheckIn	CheckOutDate	CheckOutTime	CheckOut
Joseph	Acaba	19/11/2019	228	94	19/11/2019	13:22:59	☑	19/11/2019	13:23:16	☑
Joseph	Acaba	19/11/2019	231	94	19/11/2019	13:28:27	☑	19/11/2019	14:01:07	☑
Joseph	Acaba	19/11/2019	234	94	19/11/2019	19:12:02	☑	19/11/2019	19:26:59	☑
Joseph	Acaba	19/11/2019	236	94	19/11/2019	19:29:06	☑	19/11/2019	19:59:18	☑
Joseph	Acaba	19/11/2019	250	94	20/11/2019	19:44:24	☑	20/11/2019	19:59:24	☑
Joseph	Acaba	19/11/2019	253	94	20/11/2019	21:01:56	☑	20/11/2019	21:02:15	☑
Joseph	Acaba	19/11/2019	256	94	21/11/2019	04:57:25	☑			☐
Loren	Acton	19/11/2019	241	95	19/11/2019	22:50:39	☑	19/11/2019	23:53:45	☑
Loren	Acton	19/11/2019	249	95	20/11/2019	19:41:51	☑	20/11/2019	19:42:02	☑
James	Adamson	19/11/2019	232	96	19/11/2019	13:31:25	☑	19/11/2019	14:10:10	☑
James	Adamson	19/11/2019	235	96	19/11/2019	19:18:44	☑	19/11/2019	19:26:48	☑
James	Adamson	19/11/2019	237	96	19/11/2019	19:59:27	☑	19/11/2019	20:20:23	☑
Andrew	Allen	19/11/2019	243	101	19/11/2019	22:52:46	☑	20/11/2019	22:49:43	☑
Andrew	Allen	19/11/2019	254	101	20/11/2019	22:49:23	☑	20/11/2019	22:49:43	☑
Josephina	Allen	18/11/2019	227	102	19/11/2019	12:41:12	☑	19/11/2019	12:42:20	☑
Scott	Altman	19/11/2019	242	103	19/11/2019	22:50:48	☑	19/11/2019	23:54:14	☑
Scott	Altman	19/11/2019	251	103	20/11/2019	19:46:55	☑	20/11/2019	19:59:22	☑
William	Anders	19/11/2019	239	104	19/11/2019	22:25:33	☑	19/11/2019	22:29:22	☑
William	Anders	19/11/2019	240	104	19/11/2019	22:31:39	☑	19/11/2019	23:25:04	☑
Sidney	Gutierrez	19/11/2019	233	106	19/11/2019	15:01:07	☑	19/11/2019	17:55:19	☑
Sidney	Gutierrez	19/11/2019	238	106	19/11/2019	20:20:45	☑	19/11/2019	22:29:31	☑
Dominic	Gorie	19/11/2019	244	107	20/11/2019	22:50:09	☑	20/11/2019	22:49:58	☑
James	Harbaugh	19/11/2019	245	108	20/11/2019	22:50:32	☑	20/11/2019	22:50:33	☑
Laurel	Clark	19/11/2019	246	109	19/11/2019	23:23:15	☑	19/11/2019	23:24:57	☑
Laurel	Clark	19/11/2019	252	109	20/11/2019	20:12:08	☑	20/11/2019	21:02:19	☑
Stephanie	Wilson	19/11/2019	247	111	20/11/2019	22:50:51	☑	20/11/2019	22:50:52	☑
Stephanie	Wilson	19/11/2019	255	111	20/11/2019	22:55:11	☑	20/11/2019	22:56:27	☑
Paul	Weitz	19/11/2019	248	113	19/11/2019	23:30:03	☑	19/11/2019	23:30:17	☑
			(New)				▣			▣



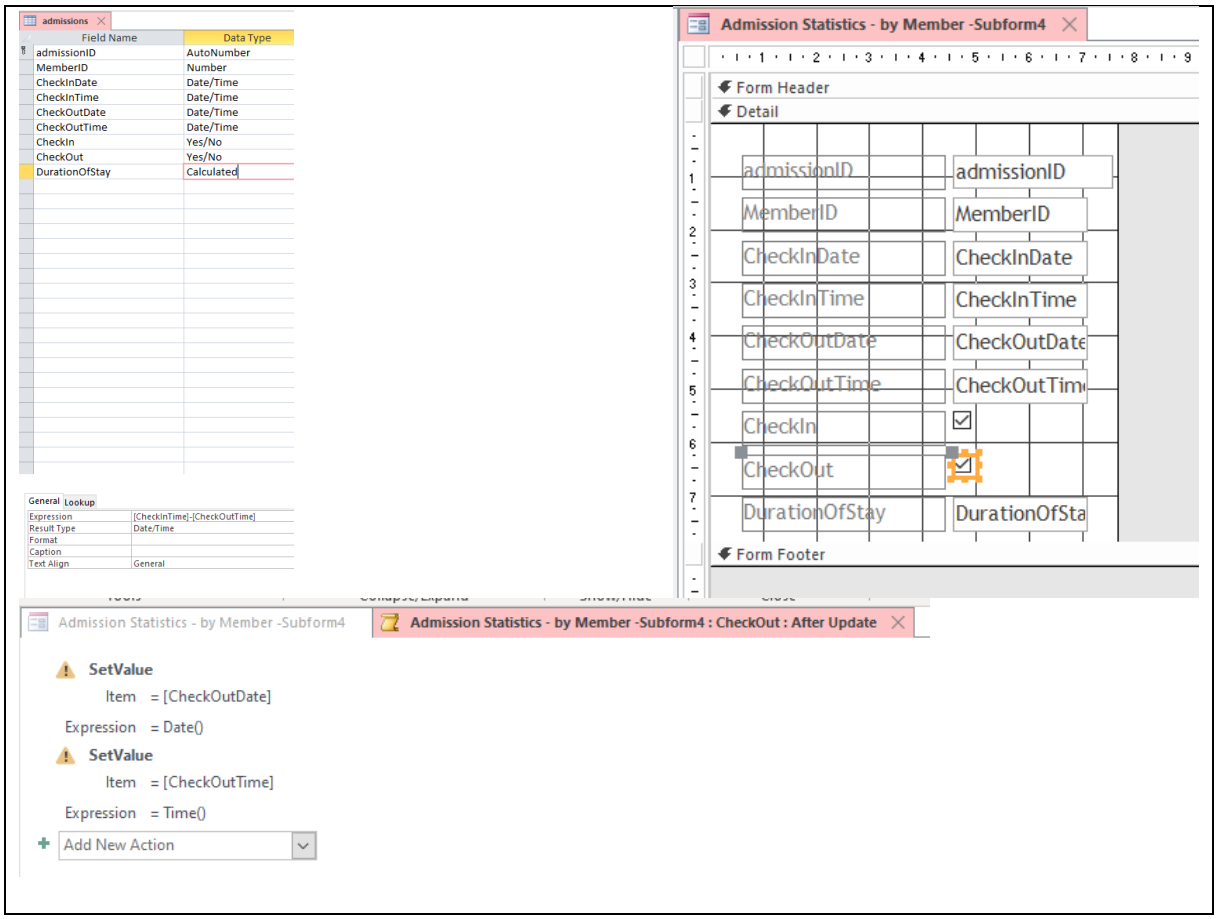
- A simple query displayed in datasheet view and created through SQL and saved allows this data to be dynamically queried each time the Admin page is loaded.
- Each time the page is loaded it updates with the latest information entered in the database and output into a chart based on duration of stay by member first name in this instance.

Advanced Query, Subform and Form combination to create the Check In/Out functionality

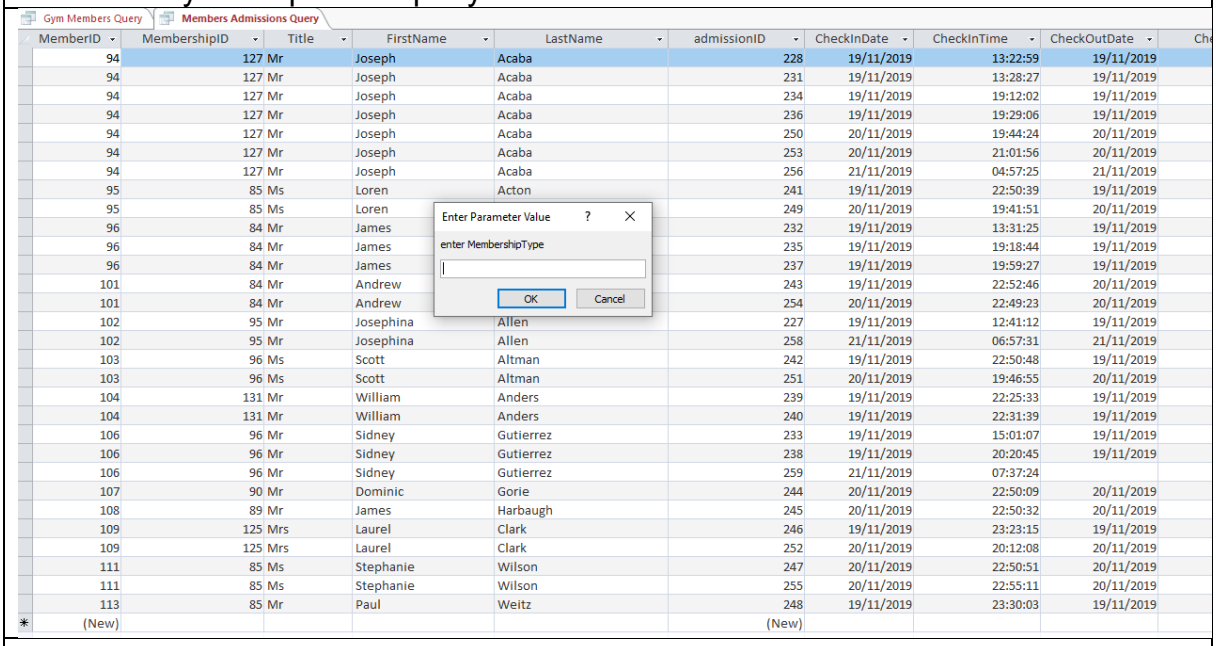
Landing Form | Admin Form | admissions

admissionID	MemberID	CheckInDate	CheckInTime	CheckIn	CheckOutDate	CheckOutTime	CheckOut	DurationOfStay	Click to Add
227	102	19/11/2019	12:41:12	☑	19/11/2019	12:42:20	☑	00:01:08	
228	94	19/11/2019	13:22:59	☑	19/11/2019	13:23:16	☑	00:00:17	
231	94	19/11/2019	13:28:27	☑	19/11/2019	14:01:07	☑	00:32:40	
232	96	19/11/2019	13:31:25	☑	19/11/2019	14:10:10	☑	00:38:45	
233	106	19/11/2019	15:01:07	☑	19/11/2019	17:55:19	☑	02:54:12	
234	94	19/11/2019	19:12:02	☑	19/11/2019	19:26:59	☑	00:14:57	
235	96	19/11/2019	19:18:44	☑	19/11/2019	19:26:48	☑	00:08:04	
236	94	19/11/2019	19:29:06	☑	19/11/2019	19:59:18	☑	00:30:12	
237	96	19/11/2019	19:59:27	☑	19/11/2019	20:20:23	☑	00:20:56	
238	106	19/11/2019	20:20:45	☑	19/11/2019	22:29:31	☑	02:08:46	
239	104	19/11/2019	22:25:33	☑	19/11/2019	22:29:22	☑	00:03:49	
240	104	19/11/2019	22:31:39	☑	19/11/2019	23:25:04	☑	00:53:25	
241	95	19/11/2019	22:50:39	☑	19/11/2019	23:53:45	☑	01:03:06	
242	103	19/11/2019	22:50:48	☑	19/11/2019	23:54:14	☑	01:03:26	
243	101	19/11/2019	22:52:46	☑	20/11/2019	22:49:43	☑	00:03:03	
244	107	20/11/2019	22:50:09	☑	20/11/2019	22:49:58	☑	00:00:11	
245	108	20/11/2019	22:50:32	☑	20/11/2019	22:50:33	☑	00:00:01	
246	109	19/11/2019	23:23:15	☑	19/11/2019	23:24:57	☑	00:01:42	
247	111	20/11/2019	22:50:51	☑	20/11/2019	22:50:52	☑	00:00:01	
248	113	19/11/2019	23:30:03	☑	19/11/2019	23:30:17	☑	00:00:14	
249	95	20/11/2019	19:41:51	☑	20/11/2019	19:42:02	☑	00:00:11	
250	94	20/11/2019	19:44:24	☑	20/11/2019	19:59:24	☑	00:15:00	
251	103	20/11/2019	19:46:55	☑	20/11/2019	19:59:22	☑	00:12:27	
252	109	20/11/2019	20:12:08	☑	20/11/2019	21:02:19	☑	00:50:11	
253	94	20/11/2019	21:01:56	☑	20/11/2019	21:02:15	☑	00:00:19	
254	101	20/11/2019	22:49:23	☑	20/11/2019	22:49:43	☑	00:00:20	
255	111	20/11/2019	22:55:11	☑	20/11/2019	22:56:27	☑	00:01:16	
256	94	21/11/2019	04:57:25	☑			☐		
(New)	0			☐			☐		

This function above also applies to the Staff Log In/Out forms and similarly, the Staff wages input form.



Functionality Example of a query



MemberID	MembershipID	Title	FirstName	LastName	admissionID	CheckInDate	CheckInTime	CheckOutDate
94	127	Mr	Joseph	Acaba	228	19/11/2019	13:22:59	19/11/2019
94	127	Mr	Joseph	Acaba	231	19/11/2019	13:28:27	19/11/2019
94	127	Mr	Joseph	Acaba	234	19/11/2019	19:12:02	19/11/2019
94	127	Mr	Joseph	Acaba	236	19/11/2019	19:29:06	19/11/2019
94	127	Mr	Joseph	Acaba	250	20/11/2019	19:44:24	20/11/2019
94	127	Mr	Joseph	Acaba	253	20/11/2019	21:01:56	20/11/2019
94	127	Mr	Joseph	Acaba	256	21/11/2019	04:57:25	21/11/2019
95	85	Ms	Loren	Acton	241	19/11/2019	22:50:39	19/11/2019
95	85	Ms	Loren	Acton	249	20/11/2019	19:41:51	20/11/2019
96	84	Mr	James	Allen	232	19/11/2019	13:31:25	19/11/2019
96	84	Mr	James	Allen	235	19/11/2019	19:18:44	19/11/2019
96	84	Mr	James	Allen	237	19/11/2019	19:59:27	19/11/2019
101	84	Mr	Andrew	Allen	243	19/11/2019	22:52:46	20/11/2019
101	84	Mr	Andrew	Allen	254	20/11/2019	22:49:23	20/11/2019
102	95	Mr	Josephina	Allen	227	19/11/2019	12:41:12	19/11/2019
102	95	Mr	Josephina	Allen	258	21/11/2019	06:57:31	21/11/2019
103	96	Ms	Scott	Altman	242	19/11/2019	22:50:48	19/11/2019
103	96	Ms	Scott	Altman	251	20/11/2019	19:46:55	20/11/2019
104	131	Mr	William	Anders	239	19/11/2019	22:25:33	19/11/2019
104	131	Mr	William	Anders	240	19/11/2019	22:31:39	19/11/2019
106	96	Mr	Sidney	Gutierrez	233	19/11/2019	15:01:07	19/11/2019
106	96	Mr	Sidney	Gutierrez	238	19/11/2019	20:20:45	19/11/2019
106	96	Mr	Sidney	Gutierrez	259	21/11/2019	07:37:24	
107	90	Mr	Dominic	Gorie	244	20/11/2019	22:50:09	20/11/2019
108	89	Mr	James	Harbaugh	245	20/11/2019	22:50:32	20/11/2019
109	125	Mrs	Laurel	Clark	246	19/11/2019	23:23:15	19/11/2019
109	125	Mrs	Laurel	Clark	252	20/11/2019	20:12:08	20/11/2019
111	85	Ms	Stephanie	Wilson	247	20/11/2019	22:50:51	20/11/2019
111	85	Ms	Stephanie	Wilson	255	20/11/2019	22:55:11	20/11/2019
113	85	Mr	Paul	Weitz	248	19/11/2019	23:30:03	19/11/2019
(New)					(New)			

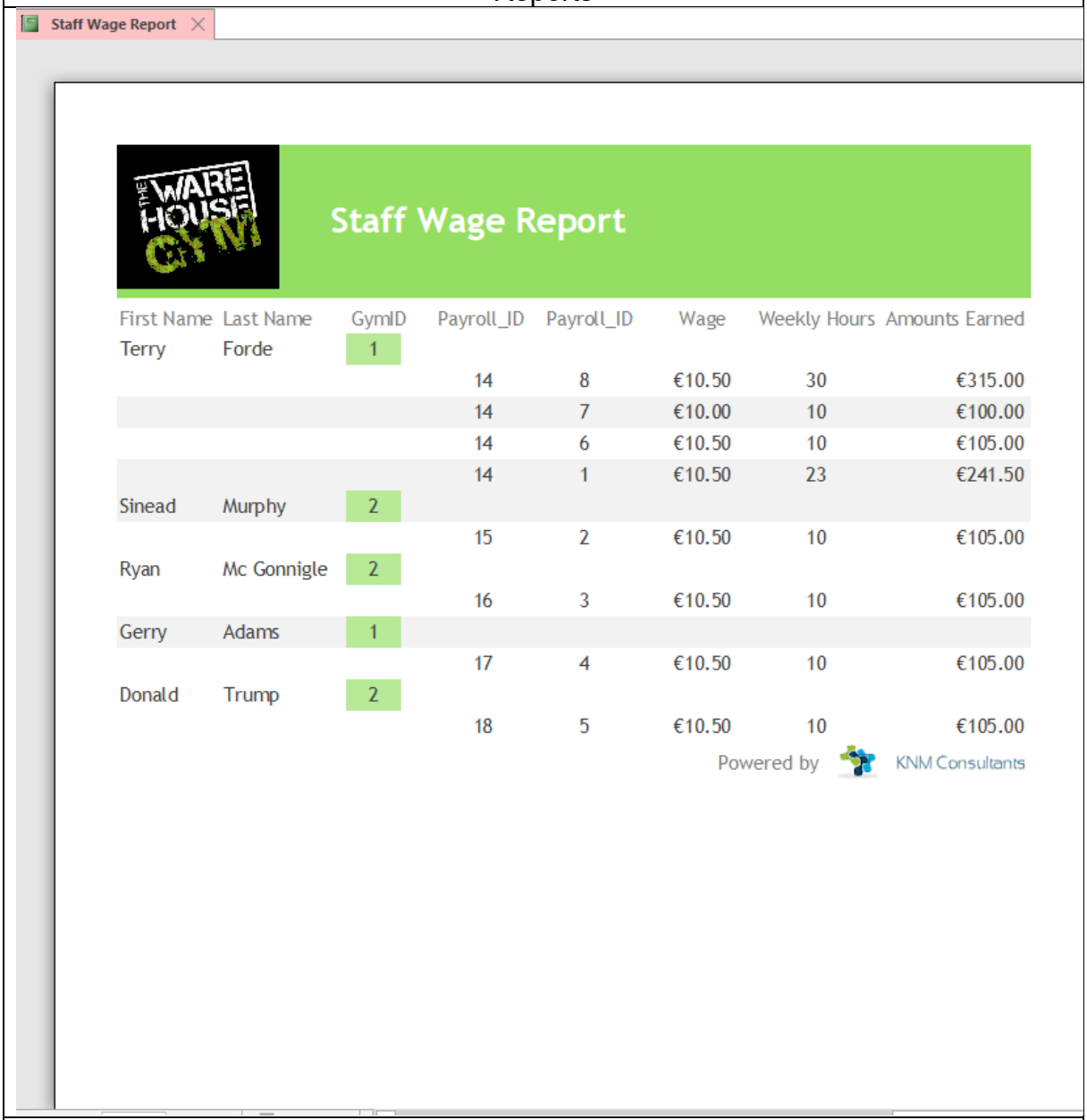
Macros

A number of macros were created to bring the functionality alive. As with the above example, the macro needed to be contained within the form for the purposes of recording the data in the subform. This could not be possible within the table without user access so we decided to contain this within another form. This is done many times throughout the database. We decided there was no need for any major macros as the staff typically would not need more advanced functions or need to delete the data.

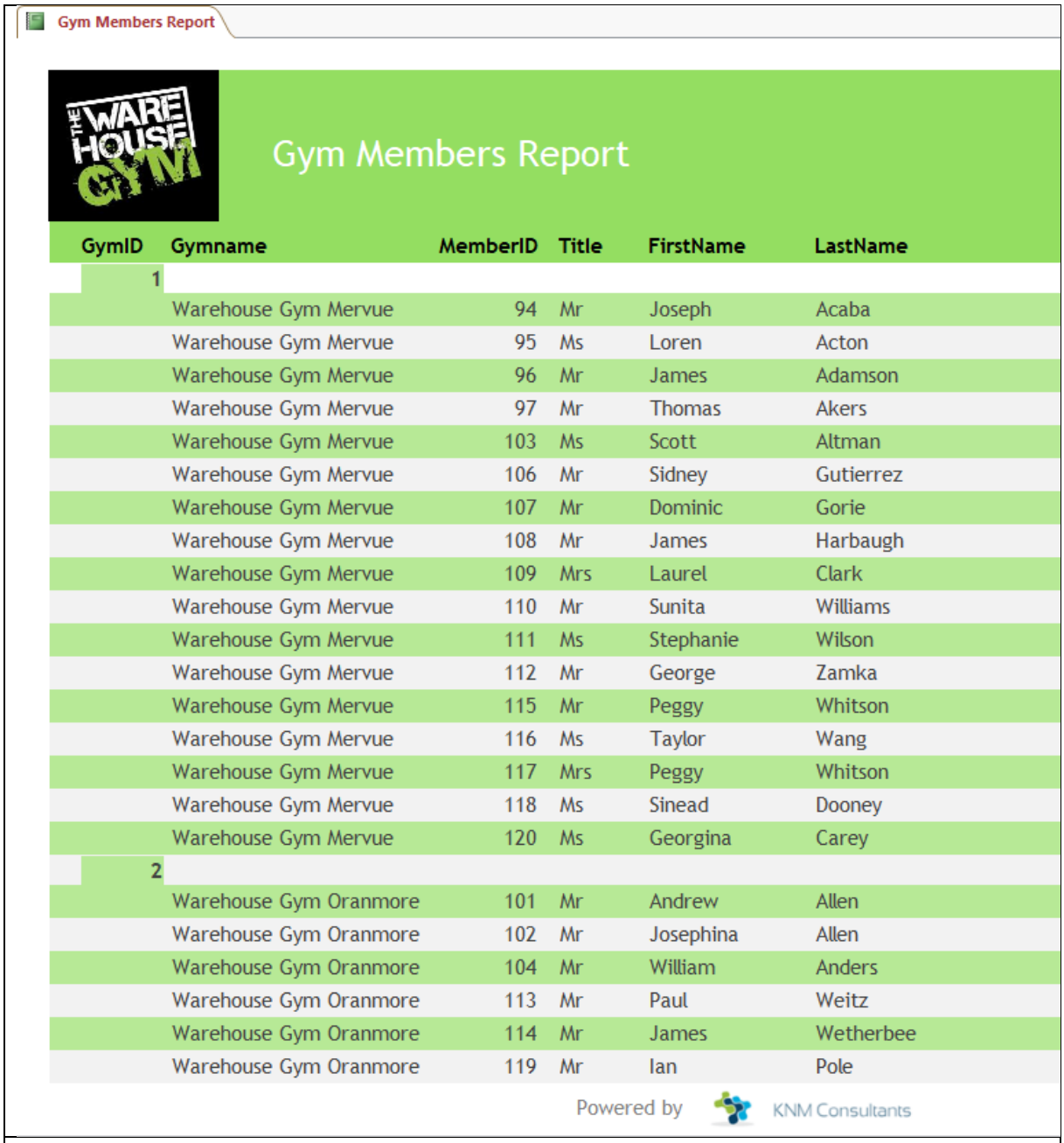
Reports

We included a number of Reports to enhance the functionality of the database. An example of some are below. Reports are easily created by the report wizard with some additional SQL for more advanced types. Aggregation is also easy to do if so required, this is done within the tables or subqueries themselves.

Reports



This report was created to show management how much each staff member earned during the payment period.



Gym Members Report

Gym Members Report

GymID	Gymname	MemberID	Title	FirstName	LastName
1					
	Warehouse Gym Mervue	94	Mr	Joseph	Acaba
	Warehouse Gym Mervue	95	Ms	Loren	Acton
	Warehouse Gym Mervue	96	Mr	James	Adamson
	Warehouse Gym Mervue	97	Mr	Thomas	Akers
	Warehouse Gym Mervue	103	Ms	Scott	Altman
	Warehouse Gym Mervue	106	Mr	Sidney	Gutierrez
	Warehouse Gym Mervue	107	Mr	Dominic	Gorie
	Warehouse Gym Mervue	108	Mr	James	Harbaugh
	Warehouse Gym Mervue	109	Mrs	Laurel	Clark
	Warehouse Gym Mervue	110	Mr	Sunita	Williams
	Warehouse Gym Mervue	111	Ms	Stephanie	Wilson
	Warehouse Gym Mervue	112	Mr	George	Zamka
	Warehouse Gym Mervue	115	Mr	Peggy	Whitson
	Warehouse Gym Mervue	116	Ms	Taylor	Wang
	Warehouse Gym Mervue	117	Mrs	Peggy	Whitson
	Warehouse Gym Mervue	118	Ms	Sinead	Dooney
	Warehouse Gym Mervue	120	Ms	Georgina	Carey
2					
	Warehouse Gym Oranmore	101	Mr	Andrew	Allen
	Warehouse Gym Oranmore	102	Mr	Josephina	Allen
	Warehouse Gym Oranmore	104	Mr	William	Anders
	Warehouse Gym Oranmore	113	Mr	Paul	Weitz
	Warehouse Gym Oranmore	114	Mr	James	Wetherbee
	Warehouse Gym Oranmore	119	Mr	Ian	Pole

Powered by KNM Consultants

This report was created to show how many members are in each gym, both Mervue and Oranmore.It was based on the Gym members query.


Members admissions report

Members admissions report

MemberID	MembershipID	Title	FirstName	LastName	admissionID	CheckInDate	CheckInTime	CheckOutDate	CheckOutTi
94	127	Mr	Joseph	Acaba					
					250	20/11/2019	19:44:24	20/11/2019	19:59:2
					256	21/11/2019	04:57:25	21/11/2019	06:58:4
					228	19/11/2019	13:22:59	19/11/2019	13:23:1
					236	19/11/2019	19:29:06	19/11/2019	19:59:1
					253	20/11/2019	21:01:56	20/11/2019	21:02:1
					231	19/11/2019	13:28:27	19/11/2019	14:01:0
					234	19/11/2019	19:12:02	19/11/2019	19:26:5
95	85	Ms	Loren	Acton					
					241	19/11/2019	22:50:39	19/11/2019	23:53:4
					249	20/11/2019	19:41:51	20/11/2019	19:42:0
96	84	Mr	James	Adamson					
					235	19/11/2019	19:18:44	19/11/2019	19:26:4
					237	19/11/2019	19:59:27	19/11/2019	20:20:2
					232	19/11/2019	13:31:25	19/11/2019	14:10:1
101	84	Mr	Andrew	Allen					
					243	19/11/2019	22:52:46	20/11/2019	22:49:4
					254	20/11/2019	22:49:23	20/11/2019	22:49:4
102	95	Mr	Josephina	Allen					
					227	19/11/2019	12:41:12	19/11/2019	12:42:2
					258	21/11/2019	06:57:31	21/11/2019	06:58:3
103	96	Ms	Scott	Altman					
					242	19/11/2019	22:50:48	19/11/2019	23:54:1
					251	20/11/2019	19:46:55	20/11/2019	19:59:2
104	131	Mr	William	Anders					
					240	19/11/2019	22:31:39	19/11/2019	23:25:0
					239	19/11/2019	22:25:33	19/11/2019	22:29:2

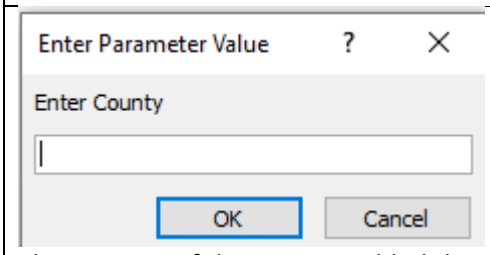
This Report includes a parameter query which was desigend to show the admissions of every member including the clock in's and out. This is an excellent report listing the statistics per member and to evaluate the busy hours in the gym and associated decision making.

Members Report

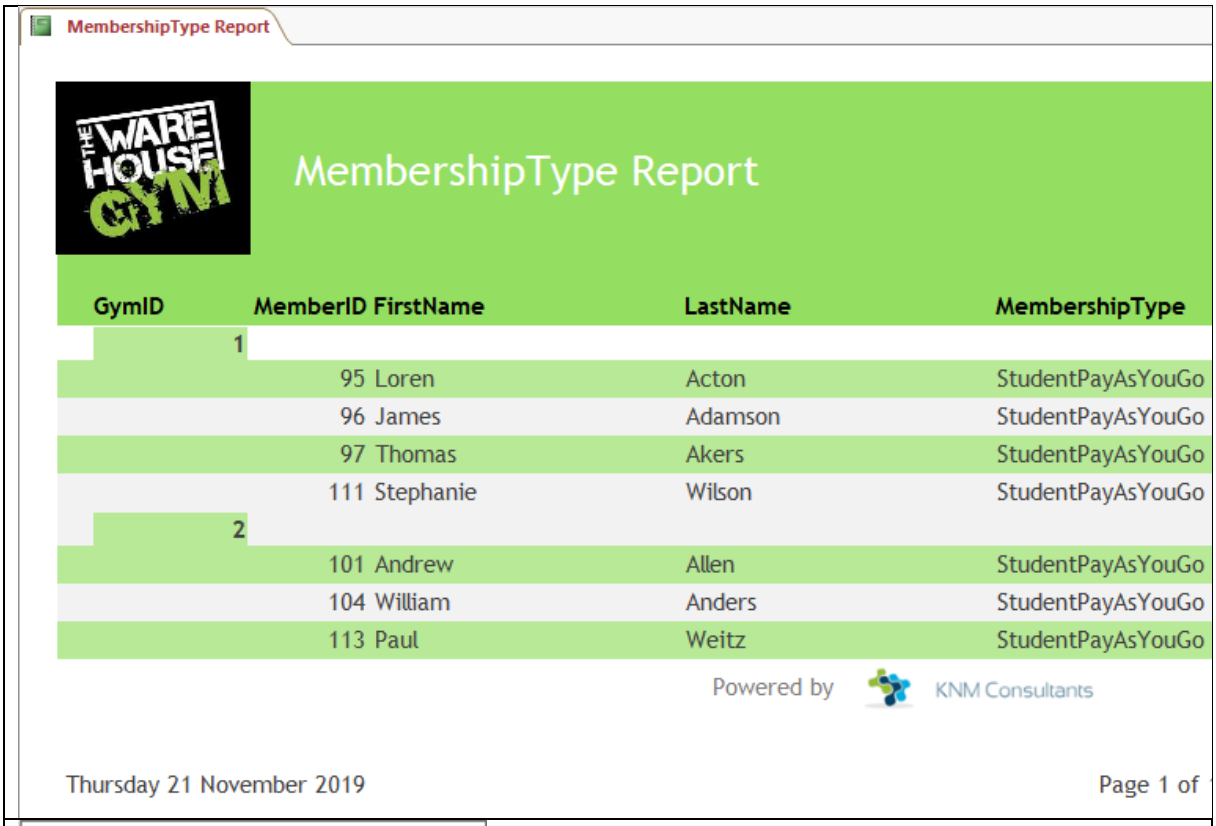
Members Report

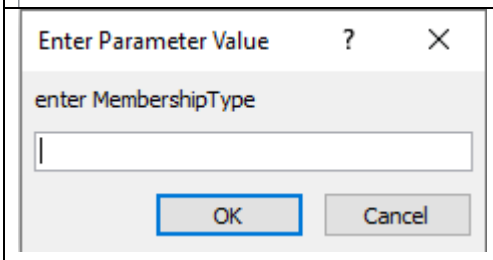
MemberID	FirstName	LastName	Title	County
94	Joseph	Acaba	Mr	Galway
95	Loren	Acton	Ms	Galway
96	James	Adamson	Mr	Galway
97	Thomas	Akers	Mr	Galway
101	Andrew	Allen	Mr	Galway
102	Josephina	Allen	Mr	Galway
108	James	Harbaugh	Mr	Galway
109	Laurel	Clark	Mrs	Galway
110	Sunita	Williams	Mr	Galway
111	Stephanie	Wilson	Ms	Galway
112	George	Zamka	Mr	Galway
113	Paul	Weitz	Mr	Galway
114	James	Wetherbee	Mr	Galway
115	Peggy	Whitson	Mr	Galway
116	Taylor	Wang	Ms	Galway
117	Peggy	Whitson	Mrs	Galway

Thursday 21 November 2019 Page 1 of 1



The creation of this query enabled the output and creation of the members report, there is a parameter query contained against the County field. To create the report, the user double clicks on the report and the parameter query box pops up. The user enters the county that the member is from and the report is subsequently created from the query result.





This report was created from the Membership type query. This report has a perameter query and will only accept data from the field 'MembershipType' in the membership table.The perameter was [Membershiptype], creating a pop up asking to enter a membership type.This is dependant on the user entering the correct characters relating to the membership type options.

Testing procedures that were carried out to test the database

For the testing procedures we managed to do a lot of this during the actual development of the database through regression testing. Following on to the completed project the most testing was done around data entry and navigating records. Examples of testing we conducted was modifying records within forms and checking associated tables for the correct operation. We relooking for fields with errors populating or blank fields, incorrect data entry or data duplication.

We conducted; Schema testing, GUI, UI, table and column testing. We carried out structural, functional and non-functional testing to our satisfaction. Once this was completed we were totally satisfied of the completed and fully functioning Gym Database.

KNM Consultants

WE MAKE YOUR BUSINESS BETTER THAN OTHERS

Our Services

MARKETING

We offer bespoke Digital Marketing services to both large as well as small businesses. Digital Marketing based in Ireland. We are experts in SEO, PPC, Social, Content. Increase Your Sales. Gains Online Visibilty. Amenities Unbeatable results.

FINANCE

KNM Consultants now have expanded into the financial markets leading Ireland's Financial Services Consulting team is a considerable task which provides our customers with the latest expertise on regulatory compliance, operations integration.

DESIGN

Reach new customers with e-mail address files of the highest quality. Request a sample directly via the website. Generate more leads with our addresses. Dentists. Hotels. Hospitals. Hairdressers. Restaurants. Schools. Wholesalers.

E-COMMERCE

Build your site on our most powerful eCommerce platform. See a free demo. Drive New Revenue Growth. Reduce Business Costs. Get to Market Quickly. 24/7 Support. Get A Free Demo. Services: Commerce Cloud, Digital Transformation.

Why work with us?

- Unrivaled aftercare service and suport with all our software and database packages.
- Solid work ethic with an emphasis on strategic management and corporate governance.
- Latest industry knowledge in application development, database design and implementations.

About Us

Established in 2019 as part of an application development project in year four business information systems in college, Kevin, Nigel and Michael founded KNM Consulting. We specialise in building digital transformative digital solutions for your business. Contact us below to find out more or visit our website on www.knmconsulting.com

Presented to Dr. Trevor Clohessy as part of a GMIT Business Information Systems BSc (Hons) 4th Year —Application Development Project

by: Kevin Blaney, Nigel Waldron and Michael O'Grady

www.knmconsultants.com