Co

*SADDAD1*

CORACLE CHRISTIAN COMMUNITY

DATABASE DESIGN

Niall Curley

Contents

[**Problem Defination** 4](#_Toc451973800)

[**System Investigation** 6](#_Toc451973801)

[**Initial Research** 6](#_Toc451973802)

[**Life Church Belfast** 7](#_Toc451973803)

[**Interview & Questionnaire:** 7](#_Toc451973804)

[**Observation & Documentation** 10](#_Toc451973805)

[**Coracle Documentation** 12](#_Toc451973806)

[**Coracle Statement of Public Benefit** 13](#_Toc451973807)

[**Functional Specification** 14](#_Toc451973808)

[**Purpose** 14](#_Toc451973809)

[**Aim** 14](#_Toc451973810)

[**Scope** 14](#_Toc451973811)

[**Definitions** 15](#_Toc451973812)

[**Functional Requirements** 16](#_Toc451973813)

[**Entity Descriptions** 17](#_Toc451973814)

[**Church Member** 17](#_Toc451973815)

[**Course Booking** 17](#_Toc451973816)

[**Donation** 17](#_Toc451973817)

[**Church Duty Rota** 18](#_Toc451973818)

[Description 18](#_Toc451973819)

[Church\_Duty\_ID 18](#_Toc451973820)

[Identifies unique number of voluntary role on rota 18](#_Toc451973821)

[Church\_Member\_ID 18](#_Toc451973822)

[Uniquely identifies a Church Member 18](#_Toc451973823)

[Duty\_Type 18](#_Toc451973824)

[Identifies voluntary role 18](#_Toc451973825)

[Role Date 18](#_Toc451973826)

[Date of voluntary duty 18](#_Toc451973827)

[**Child Protection** 18](#_Toc451973828)

[**Youth Group** 18](#_Toc451973829)

[**System Constraints** 19](#_Toc451973830)

[**Stakeholders** 19](#_Toc451973831)

[**Precedence Table of Activities** 20](#_Toc451973832)

[**Project Schedule** 20](#_Toc451973833)

[**Gantt Chart (tracking)** 21](#_Toc451973834)

[**Timeline** 22](#_Toc451973835)

[**Data Base Technologies** 24](#_Toc451973836)

[**Flat- File Database** 25](#_Toc451973837)

[**Hierarchical Database** 26](#_Toc451973838)

[**Network Database** 27](#_Toc451973839)

[**Relational Database** 28](#_Toc451973840)

[**Conclusion** 29](#_Toc451973841)

[**Methodologies** 30](#_Toc451973842)

[**The Prototype Model** 31](#_Toc451973843)

[**The Incremental Model** 33](#_Toc451973844)

[**Agile Model** 34](#_Toc451973845)

[**The Spiral Model** 36](#_Toc451973846)

[**The Waterfall Model** 37](#_Toc451973847)

[**Conclusion** 38](#_Toc451973848)

[**Design Specification** 39](#_Toc451973849)

[**Purpose** 39](#_Toc451973850)

[**Initial Entities (ER Diagrams)** 40](#_Toc451973851)

[**Church Membership (entity) -** 40](#_Toc451973852)

[**Church Duty Volunteer (entity)** 40](#_Toc451973853)

[**Youth (entity)** 40](#_Toc451973854)

[**Donation (entity)** 40](#_Toc451973855)

[**Child Protection (entity)** 40](#_Toc451973856)

[**ER DIAGRAM (1)** 41](#_Toc451973857)

[**ER DIAGRAM 2** 42](#_Toc451973858)

[**PHYSICAL ER DIAGRAM 3** 43](#_Toc451973859)

[**Data Dictionaries & Queries** 44](#_Toc451973860)

[**Child Protection** 45](#_Toc451973861)

[**Church Duty Rota** 47](#_Toc451973862)

[**Church Member** 49](#_Toc451973863)

[**Course Booking** 53](#_Toc451973864)

[**Donation** 55](#_Toc451973865)

[**Course** 57](#_Toc451973866)

[**Training** 59](#_Toc451973867)

[**Youth Group** 61](#_Toc451973868)

[**Prototype Testing** 63](#_Toc451973869)

[**Prototype & Feedback** 64](#_Toc451973870)

[**Unit Testing** 68](#_Toc451973871)

[**Test 1 Church Member Form** 68](#_Toc451973872)

[**Test 2 Church Member Form** 69](#_Toc451973873)

[**Rectifying Test 2** 70](#_Toc451973874)

[**Church Member - Test 3 (DOB)** 71](#_Toc451973875)

[**Church Member – Test 4 (ZIP)** 72](#_Toc451973876)

[**Church Duty Rota – Test 5** 73](#_Toc451973877)

[**Test Queries** 74](#_Toc451973878)

[**Church Member1** 74](#_Toc451973879)

[**Church Leader** 75](#_Toc451973880)

[**Testing Reports** 77](#_Toc451973881)

[**Caretaker Duty Rota** 77](#_Toc451973882)

[**Youth Group** 78](#_Toc451973883)

[**Donation** 79](#_Toc451973884)

[**Acceptance Testing (Navigation Form)** 80](#_Toc451973885)

[**Testing Navigation Form** 80](#_Toc451973886)

[**Church Duty Rota (Test)** 81](#_Toc451973887)

[**Youth Group (Test)** 81](#_Toc451973888)

[**Child Protection & Course (Test)** 82](#_Toc451973889)

[**Coracle Course Booking & Training & (test)** 83](#_Toc451973890)

[**Coracle Donation** 84](#_Toc451973891)

# **Problem Defination**



**Coracle Christian Community**

**St Agnes Parish Centre**

**Belfast**

**BT119BW**

**1 February 2016**

**BMC Consulting**

**123 Millfield Complex**

**Belfast**

**BT1 1HS**

Dear Sir/Madam,

I am writing of behalf of the leadership of our church fellowship **(Coracle Christian Community)**, with reference to finding solutions to some of our organizational problems. Coracle has been ministering in this community for twenty plus years. For the last five years we have been based at St Agnes Parish Centre, Belfast.

In the early years when were a smaller organization we met in homes, community centres and other smaller venues but as we have grown in numbers and got involved in community projects we have found it necessary to keep more records about our organization. We feel at this time that we need a computerized system to keep record of our membership, volunteers, activities, caretaker’s schedules, counselling services and various other activities.

Any new system would need to be able to organize the different courses that we have available to our church membership in conjunction with the appointed leaders to who take the courses. We will need the ability to record our child protection information and create reports regarding issues like, vetting dates and vetting expiry dates and child protection training and renewal dates. Also the system would need to be able to keep note of the numbers of children and recognise their ages of the children who attend our crèche, youth group and drama group for organizing staff to meet correct child protection supervisory levels etc.

We need to be able to schedule for Church leaders to speak at certain times and carry out scripture studies in certain rooms, also our worship teams are to be allocated certain times and rooms to practise to prepare to minister to the congregation.

As we are a registered charity all our church member donations must be logged and submitted to the Inland Revenue in order to claim under our charitable status. So we need to be able to produce reports and queries for this purpose, how much was paid into our church by non-tax payers and taxpayers etc.

As a church we depend on our volunteers offering their services, so any new system would need to be user friendly. We also have a limited budget and would have certain financial restraints, so any cost would have to be considered by our Church Treasury. We would like to have the new system for the July 2016.

Could you please contact me at the following telephone number or email address to arrange for a consultation.

Mob (07761820781, T.Hoggcoracle@yahoo.com), (www.coracleireland.org)

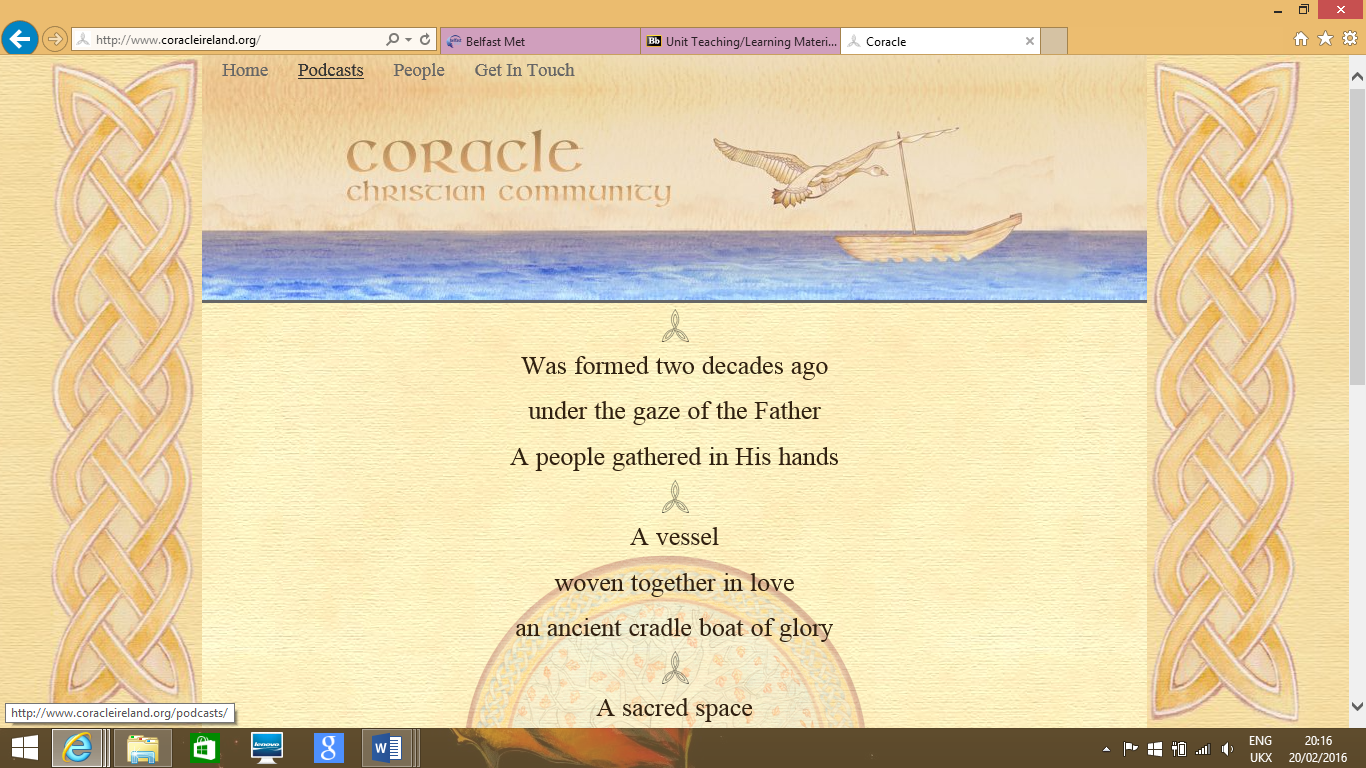
**Terence Hogg (Senior Pastoral leader)**

# **System Investigation**

**Fact Finding -** Having received written correspondence from Coracle Christian Fellowship regarding the need for a better system, a systems investigation will be carried out to ascertain what the real needs of the organization are. This will establish as much factual information as is necessary to produce a Functional Specification.

## **Initial Research**

Before carrying out on site observation or interviews, some initial research is carried out on **Coracles website** and that of another similar organization (**Life Church**).



The website seems very basic with only a few options and would itself need better developed for the organization’s needs. A brief history of the organization and its ethos and the various pod casts available for its own members to use.

## **Life Church Belfast**

Having examined this similar organizations website with its different divisions into activities, events, schedules and leadership, we have a better idea of the organization that we will be researching.



# **Interview & Questionnaire:**

Having carried out some research on Coracle and a similar organization, I arranged to meet with two of the leaders of the organization (Terence Hogg, Senior Pastoral) and (Norman Gibson, Treasurer & Designated Child Protection Officer) and I drew up a series of questions to find out about the processes and transactions of the organization.

1. How does your organization normally record its information?

*‘Our financial records are stored on spread sheets. Our rota system with volunteer schedules and course information on offer are created with Microsoft Office. Access NI Child protection forms are downloaded from Access NI website, these are manually filled in by applicants and when disclosure forms are sent back after checks have been carried out they are kept in a secure cabinet’.*

1. What information do you need to keep on your Church organization and why?

*‘We need a system that can store information and allows for updating of records. Information on church members like the name and address and contact details of our members. Useful information for contacting members like e mail address. We need to keep a log of the member’s date of birth for organizing crèche and youth groups in the case of children. We also need information to be stored on child protection (vetting). We need to store information on the donations our members contribute for the Inland Revenue and charitable status’.*

1. **What roles do people have in your organization?**

*‘We have leaders who take various courses, we have caretakers who open and close the building. We have cleaners who carry out their duties. We have volunteers who work with young children in the crèche and we have volunteers who work with the youth & drama’.*

1. **How many courses do you offer to your membership?**

*‘We currently have thirteen courses on offer which have various durations’.*

1. **How many rooms do you have for the courses and activities on offer and do they have different capacities?**

*‘We have 12 rooms and they have various capacities for different types of meetings and events’.*

1. **What information do you need stored for child protection purposes?**

*‘We need to record who has been vetted, when and when they need revetting with their child protection reference number. We also need to record who has received child protection training, when they received training and when they need to be retrained’.*

1. **How are donations made and what information do you need stored on donations?**

*‘Some donations are made manually by envelope and are recorded later to a spreadsheet and others are paid by direct debit into Church account. We need to record which church member has given a donation and whether or not that person is a taxpayer or not. We also need a means of calculating the total for the inland revenue’.*

1. **Is any of the information of a sensitive nature?**

*‘All information on church members particularly on their financial transactions and also information on children’.*

1. **Are there any reports and queries that need to be generated by the system?**

*‘We need to generate reports for leadership meetings detailing how many courses are on offer and how many will be attending these, which church leaders will be taking these. Reports about how many youths there are in our youth groups and how many children we have in our crèche groups at certain times and recording their ages etc. With regards to child protection we need to generate reports to know when certain members have been vetted/trained and when they need re-vetted/retrained. Church donations need recorded differentiating between TAX payers and Non Tax payers. We need to know how much has been donated to the Church’.*

1. **Who will be using the system?**

*‘The main administrators will be the church treasurers and leaders and designated persons who deal with child protection’.*

# **Observation & Documentation**

Having been invited to the Coracle Christian Community main meeting which was most enlightening (but that’s a different kind of enlightenment) and also having observed their volunteers group sessions times and duties I was able to gain first-hand knowledge of their current structure.

The church needs to store information on its members, various details like, name, address, contact details, date joined, gender, positon status, permanent members or temporary status and email address. The will be the first and primary entity.

Below is the church rota for various volunteers, it is displayed in their main meeting room and all volunteers are given a copy by e mail. It shows which volunteers open up and set up the main meeting rooms, which volunteers work with kids & tots and which volunteers carry out cleaning duties.

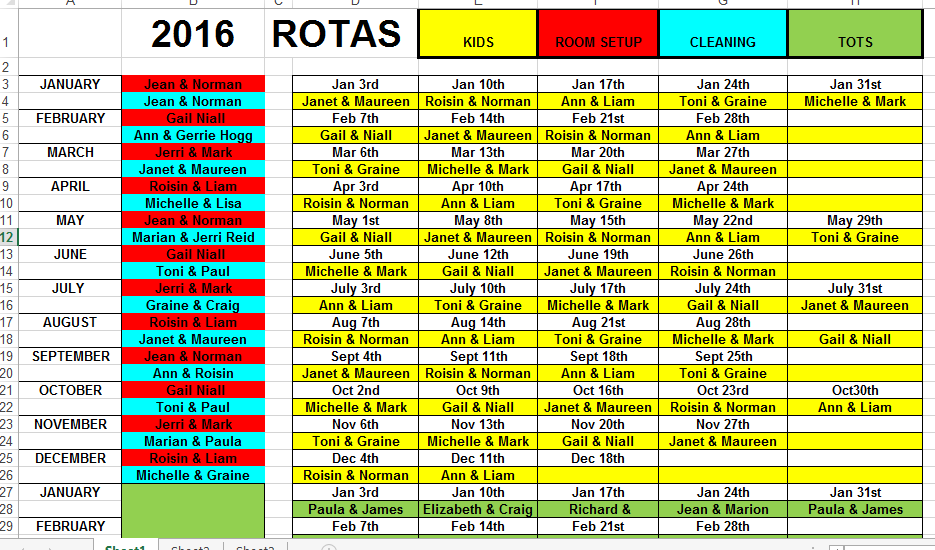
Some of the volunteers carry out *more than one role*. The ‘kids rota’ requires two volunteers at a time to work with a youth group, there are twelve volunteers in total for these roles. The ‘tots’ (children under 5) also requires two volunteers at a time, there are eight volunteers for this schedule. The opening up (Caretaker) rota requires two volunteers at a time who carry out this role for three months, there are twelve volunteers in total for this role. There is also a cleaner rota where two volunteers carry out this duty once a month. Church Duty Rota will be another entity for organizing these roles.

The church has been trying to organize itself better in terms the courses that it has to offer the community. It needs a system to organize what leaders will be taking the courses, on what dates the courses are being offered and which rooms the courses are scheduled to take place in. Course booking will be another entity for the organization of these activities.

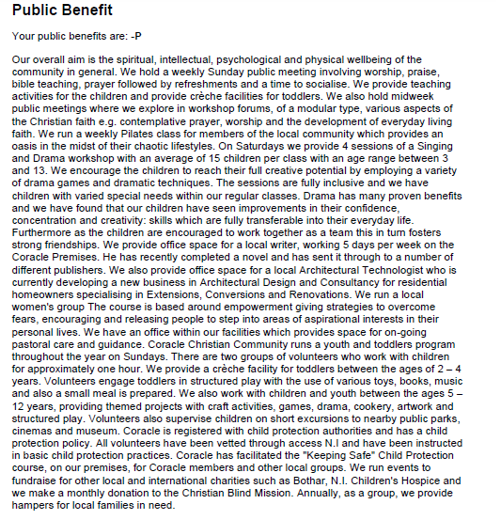
Church members give donations usually monthly by direct debit and also by cash into church donation box. This money has to be registered by the church treasurer as either given by taxpayers or non-taxpayers for the purpose of claiming on the grounds of charity status. This will be another entity called, ‘Donation’.

Having examined Coracles statement of public benefit, I can see that a drama school is taking place on Saturday and with a crèche (tots) and youth groups (kids) taking place on a Sunday meeting. The church wants to keep records of what groups that its child members are part of with their medical information. Coracle wants to store information on their volunteers who are supervising their children, details for vetting and child protection training courses. This will lead to the creation of another two entities called, ‘child protection and youth group’.

# **Coracle Documentation**



# **Coracle Statement of Public Benefit**



# **Functional Specification**

Having received a correspondence for Coracle Christian Community with details defining their system problems and requirements. With the initial research before meeting this organization and now having interviewed their leadership, financial representatives and carried out a fact finding systems investigation which involved viewing available documentation and how their current system works. It is now time to begin to develop a functional specification.

## **Purpose**

The purpose of this project is to design and build an administration system for Coracle Christian Community in order to store information on it Church membership, leaders, volunteers, courses, rotas, donations and child protection details.

## **Aim**

The basic aim of designing and building this Data Base System is to replace a manual system that is outdated and unable to cope with the growth of this organizations activities. The new system will be more accurate and will speed up their transactions making them a more efficient organization.

## **Scope**

In order to continue with the project, we need to define its parameters. These are listed below;

* To identify and meet all the needs of the organizations requirements in terms of the information to be stored and what main entities will be i.e church member details, rotas for volunteers, courses/rooms, child protection information, donations, child/youth groups distinctions.
* To create and design a system that will identify the status of full time members and that of temporary members and of leaders.
* To create a system that will allow for the updating of attributes and tables through use of forms and allow for filtering of the information by linking relevant entities for the purpose of querying, reports and presentations.
* To design and build a prototype of the system for testing.
* To create a system that is easy to navigate for the end user/administrator.
* To test the system thoroughly to make sure it works in practice before any handover.
* To produce user guide documentation that will explain how the system new will work.
* To work to a program timescale that will lead to the completion of the project on time.
* To create a system that is secure and allows only certain authorized users.

## **Definitions**

* **Church Member** – a member of the Church organization with all relevant details.
* **Volunteer** – an unpaid person who has agreed to carry out duties on behalf of the church e.g. (Caretaker volunteer, Youth volunteer, Crèche volunteer, Cleaner).
* **Youth Member** – a table identifying what group a youth is part of.
* **Course Leader** – a capable mature person within the organization who will act as a tutor or teacher.
* **Permanent** – defines whether a member is leader or just a member.
* **Rota –** an organized group of activities.
* **Status –** defines where a member is permanent or temporary.
* **Medical Info –** important information as to certain medical needs.
* **Course Type** - a designated subject matter for participation which is being offered by the church.
* **Course Room** – The particular part of the building where certain course is to take place.
* **Course Info** – Brief statement on the course contents.
* **Donation Taxpayer** – a financial contribution made by a Church Member who is currently paying income tax.
* **Donation Non Taxpayer** - a financial contribution made by a church member who is NOT currently paying income tax.
* **Time/Date –** date that person became a member of the Church, date child protection given, date child protection training expired, date that donation was given, date that course starts, date that course ends.
* **Vetted Date** – Date on which youth or crèche worker was vetted by Access NI.
* **Vetting Expiry** – Date on which vetting certificate expires and reapplication is necessary.
* **Trained** – Date on which youth or crèche worker was vetted by Access NI.
* **Training Expiry** – Date on which training expires and reapplication is necessary.
* **Access NI Reference** – the number allocated to the person who has been successfully vetted by Access NI.

# **Functional Requirements**

Having studied the problem definition and carried out a systems investigation, we have delivered the scope of our project. It is now necessary to list the scope of our objectives to see what is achievable with regards to our technical abilities and time allocated for the project.

(M: Must have, S: Should have, C: Could have, W: Wont have)

|  |  |  |
| --- | --- | --- |
| Item | Description | Rating |
| Entities | Should have all necessary entities required for various specific logging of information. | M |
| Church Membership | The system must store specific information about each church members and allow access for system queries, forms and reports. | M |
| Church Leader | The system must store information on leaders for organizing what course they are taking. | M |
| Volunteer | The system must differentiate between the various volunteers in order to create an orderly and effective rota. | S |
| Child Protection | The system must store information in relation to youth, crèche and drama volunteers needing to be vetted and re-vetted, trained and retrained. | M |
| Course Type | The system must detail list of various courses, dates, times and course information. | M |
| Course Room | Must allocate the various rooms in the building to particular courses | M |
| Donations | Calculate how much money has been donated in the tax year by tax payers and non-taxpayers. | S |
| Report | Allow the user to run reports on donations, no of volunteers, youth and children attending for leadership meetings. | M |
| Queries | The system must allow for queries into various different courses, donations, rotas, church volunteers. | M |
| Forms | The system must give the user access to key in data on new member data and to update records, change dates etc. | M |
| Security | The system must have security levels to allow certain members of the leadership to view certain records, and sensitive data on children etc. | C |
| Child/Youth | Should be able to differentiate between youth and crèche children. | S |

# **Entity Descriptions**

## **Church Member**

|  |  |
| --- | --- |
| Attribute | Description |
| ChurchMemberID | Uniquely identifies a member of the organization |
| Title | The title given to someone (Mr, Mrs, Miss, Ms, Dr) |
| Forename | Identifies first name |
| Surname | Identifies second name |
| Gender | Members Gender |
| D.O.B | Members date of birth |
| Address Line 1 | Identifies members address |
| Town | Identifies members town |
| Postcode | Identifies members postcode |
| Contact Number | Identifies members phone number |
| Mobile Number | Identifies members mobile |
| E mail Address | Identifies online contact |
| Date Joined | Date member joined organisation |
| Leader | Defines whether member is permanent or temporary |
| Permanent | Defines whether a member is a leader or ordinary member |
| Medical Information | Important medical information on church member |

## **Course Booking**

|  |  |
| --- | --- |
| Attribute | Description |
| Course\_Booking\_ID | Uniquely identifies a member of the organization |
| Church\_Member\_ID | Uniquely identifies the church member |
| Course Type | Type of course on offer |
| Course Leader | Identification of leader |
| Course\_Room | Room course held in |
| Course\_Start Date | Date course starts |
| Course End Date | Time course ends |
| Course\_Information | Basic Information on course |

## 

## **Donation**

|  |  |
| --- | --- |
| Attribute | Description |
| Donation ID | Uniquely identifies a donation |
| Church\_Membership\_ID | Uniquely identifies a Church Member |
| Donation TAX | Uniquely identifies a donation made by a TAX payer |
| Donation\_Non\_TAX | Uniquely identifies a donation made by a NON TAX payer |
| Donation\_Date | Unique date of donation |

## 

## **Church Duty Rota**

|  |  |
| --- | --- |
| Attribute | Description |
| Church\_Duty\_ID | Identifies unique number of voluntary role on rota |
| Church\_Member\_ID | Uniquely identifies a Church Member |
| Duty\_Type | Identifies voluntary role |
| Role Date | Date of voluntary duty |

## **Child Protection**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Child\_Protection\_ID | Uniquely identifies a person for child protection issues |
| AccessNIRef | Unique Access N.I reference |
| Vetted\_Date | Uniquely identifies a date |
| Vetted\_Expiry | Uniquely identifies vetting expiry date |
| Trained\_Date | Uniquely identifies child protection training date |
| Trainin\_Expiry\_Date | Uniquely identifies child protection training expiry date |

## **Youth Group**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Youth\_Group\_ID | Unique identity of Rota activity |
| Church\_Member\_ID | Unique identity of Church Member |
| Group | Type of youth Group |

**Six Entities in Total.**

# **System Constraints**

Having spoken with Coracle leadership, carried out systems investigation and viewed available documentation. There are a number of constraints placed on the proposed database system. These need to be taken into account from the outset of the project:

* This project will begin in the month of 16th February 2016 and will be completed and ready for handover by the dead line in the 4th May 2016.
* The cost for completing the project is £7500. The Church leadership have met and agreed these costs.
* The system must be set up to accommodate only authorized persons, due to sensitive nature of financial and child protection information.

## **Stakeholders**

|  |  |  |
| --- | --- | --- |
| Name | Organization | Role |
| Terence Hogg | Coracle Christian Community | Senior Pastoral |
| Niall Curley | BMC Consulting | Consultant/Developer |
| Toni McCann | Icons Drama | Drama Teacher |
| Craig Gibson | Writer | Writer |
| Michelle McCarthy | Architect | Designer |
| Children In Attendance | Coracle Youth/ Drama/Crèche | Students |

**The development costs having been agreed with BMC Consulting and the relevant contract having been signed the contract will commence.**

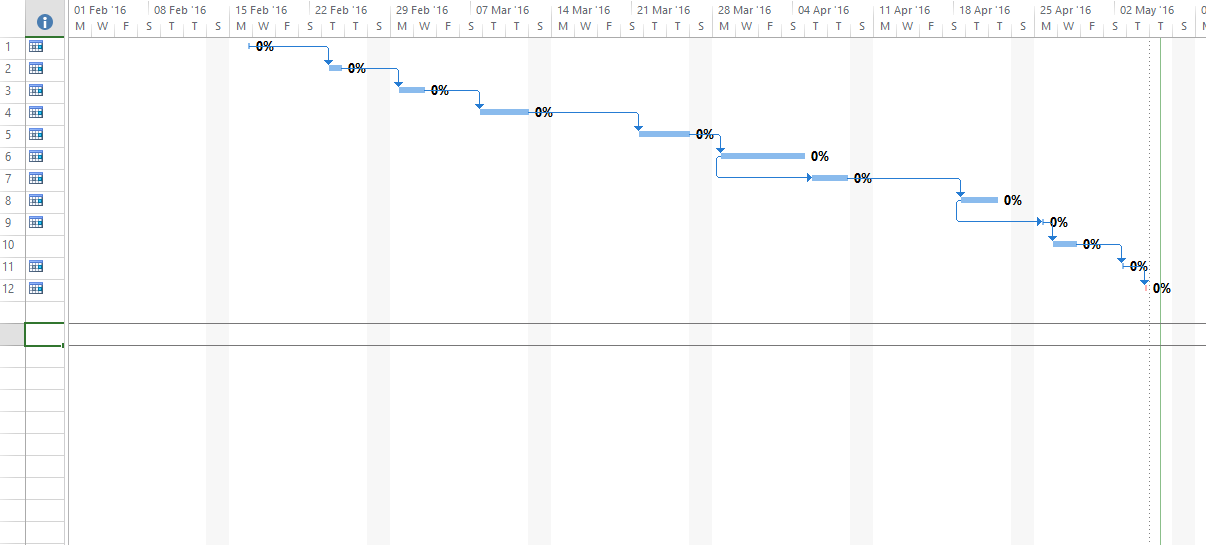
# **Precedence Table of Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Duration (hrs) | Predecessor |
| 1 | Problem Analysis | 3 | - |
| 2 | Systems Investigation | 5 | 1 |
| 3 | ER Modelling | 7 | 2 |
| 4 | Data Dictionary | 13 | 3 |
| 5 | Table Building | 15 | 4 |
| 6 | Form Building | 16 | 5 |
| 7 | Query Building | 12 | 6ss+2hrs |
| 8 | Report Building | 12 | 7 |
| 9 | Navigation | 3 | 8ss+4hrs |
| 10 | Formal Testing | 7 | 9 |
| 11 | Acceptance Testing | 3 | 10 |
| 12 | Handover | 3 | 11 |
|  |  |  |  |
|  |  |  |  |

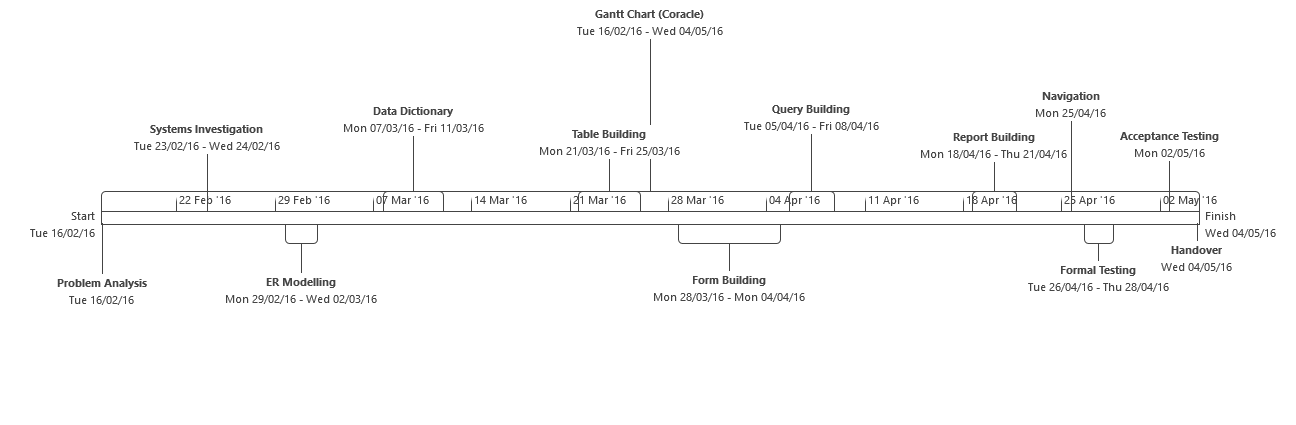
## **Project Schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task Mode | Task Name | Duration | Start | Finish | Predecessors |
| Auto Scheduled | Problem Analysis | 3 hrs | Tue 16/02/16 | Tue 16/02/16 |  |
| Auto Scheduled | Systems Investigation | 5 hrs | Tue 23/02/16 | Wed 24/02/16 | 1 |
| Auto Scheduled | ER Modelling | 7 hrs | Mon 29/02/16 | Wed 02/03/16 | 2 |
| Auto Scheduled | Data Dictionary | 13 hrs | Mon 07/03/16 | Fri 11/03/16 | 3 |
| Auto Scheduled | Table Building | 15 hrs | Mon 21/03/16 | Fri 25/03/16 | 4 |
| Auto Scheduled | Form Building | 16 hrs | Mon 28/03/16 | Mon 04/04/16 | 5 |
| Auto Scheduled | Query Building | 12 hrs | Tue 05/04/16 | Fri 08/04/16 | 6SS+2 hrs |
| Auto Scheduled | Report Building | 12 hrs | Mon 18/04/16 | Thu 21/04/16 | 7 |
| Auto Scheduled | Navigation | 3 hrs | Mon 25/04/16 | Mon 25/04/16 | 8SS+4 hrs |
| Auto Scheduled | Formal Testing | 7 hrs | Tue 26/04/16 | Thu 28/04/16 | 9 |
| Auto Scheduled | Acceptance Testing | 3 hrs | Mon 02/05/16 | Mon 02/05/16 | 10 |
| Auto Scheduled | Handover | 3 hrs | Wed 04/05/16 | Wed 04/05/16 | 11 |

# **Gantt Chart (tracking)**



# **Timeline**



# **Data Base Technologies**

In order to meet the requirements of our client, to produce a database for the use of storing & retrieving certain information required.

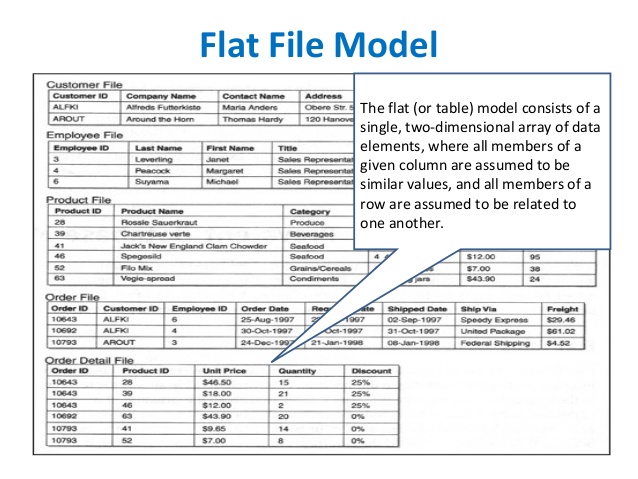
We must look at various types of data base available to determine which one is preferable There are various types of data base in use in the industry.

The main types are Flat-file Databases, Hierarchical Databases, Network Databases and Relational Databases. We need to explore these because here are advantages and disadvantages to each.

[](http://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwiJiqfp3aDMAhXFOhoKHegmAh4QjRwIBw&url=http://scoleotechnologies.com/database-technologies.html&bvm=bv.119967911,d.ZGg&psig=AFQjCNEUXwtvshb82tavQw7ECfKnls0rgw&ust=1461362401876781)

# **Flat- File Database**

Flat-File Databases are very straight forward, easy to use and easy to set up. All data is concentrated in one flat-file. This type of data base is limited as each field in this data base can only comprise of a single piece of data. This type of data base is limited where multiple fields are required. Flat-File databases cause redundancy as information is repeated.

[](http://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwinmpfxz5vMAhVDvBQKHZqHBm0QjRwIBw&url=http://www.slideshare.net/geoffreyawalker/lecture2-b-databasemanagementmodels&v6u=https://s-v6exp1-ds.metric.gstatic.com/gen_204?ip%3D2a02:c7d:7014:5a00:dc77:d76c:3a23:c83%26ts%3D1461100587443518%26auth%3Dw3obe5lyz3xygbbh65qougk3osopod4j%26rndm%3D0.4534264741518938&v6s=2&v6t=24682&psig=AFQjCNFNLtkB7c888iGWhfR_TpXY0wThow&ust=1461186987383591)

**Pros:**

* Simple to construct, easy input method.
* This is good for small set up, where not a lot of data requires storing.

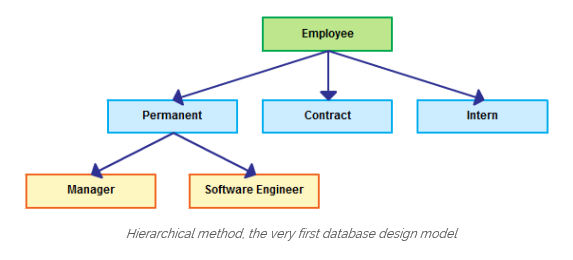
**Cons:**

* Not enough detail can be stored in this method, making it limited for queries.
* Any future need to adapt the system is expensive to modify.

# **Hierarchical Database**

This type of database, brings together databases, linking them according to their value. It consists of the primary parent database linked to many child databases. So one parent database is connected to many child databases, and all the child databases have only one parent database.

Therefore, all data that needs to be retrieved is through the parent database, making this system less flexible. Data cannot be shared between the child databases. This can lead to errors and redundancy in data being repeated (duplication of data).



**Pros:**

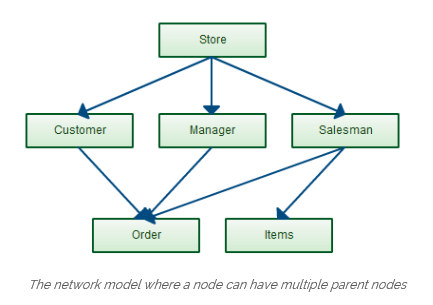
* There is a clear an ordered way to store information. This data can be collated and can be easily accessed in the specific part of the system.
* Is used in multiple databases where there are only certain data information is needed.

**Cons:**

* Because the child databases are connected and the information is not shared, it can lead to errors and redundancy.
* Time is lost due to parent databases being separate as users try to extract information and data by having to go through the parent database.

# **Network Database**

This type of database is similar to hierarchical model, it is linked in a network across each database, also at child data base level. This network structure is an improvement to the hierarchical system, with better links between child databases. This more complex system requires users with more technical knowledge due to the different levels and more complex links.



**Pros:**

* Better links between child databases & access to more data.

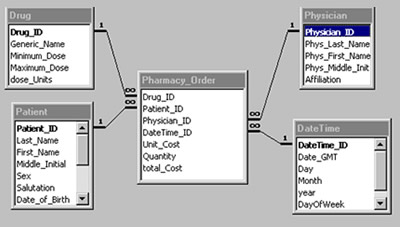
**Cons:**

* This is a more complex system for users to navigate, due to access to other parent databases.
* Users may need more training regarding this more intricate system, the emphasis being able to understand the destination of the data that is being inputted. Without this knowledge it can lead to duplication and redundancy.

# **Relational Database**

Relational Databases is a database that is divided up into tables of information that are linked by certain key attributes in order to manipulation information for queries. This very efficient system that seeks from the outset to prevent redundancy and the repetition of information. Information is entered into tables only once and can be easily accessed by the user, as well as being easily linked with other tables to form queries and reports to user’s specific needs.

The language of the relational database is SQL (structured query language).

[](https://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjUirrQ1KDMAhWEvRoKHcIgAHcQjRwIBw&url=https://www.soa.org/news-and-publications/newsletters/compact/2012/april/speaking-dat-properly.aspx&bvm=bv.119967911,d.ZGg&psig=AFQjCNH0rZkn9dkQ4yzsQr4xRQk5NRkeDg&ust=1461360021417897)

**Pros:**

* You can link tables together to manipulate data for queries and reports.
* New tables can easily be added.
* Information (data) only needs to be entered or corrected once saving errors, repetition or redundancy.
* It allows the user to gain access to data using SQL.
* This is a very flexible system and can be used in wide range of applications.

**Cons:**

* Information stored may be seldom used.
* When designing the relational database information needs to be broken down into small tables sometimes leading to islands of information (not connected to other tables).
* Requires much thought regarding these relationships in the design stages, through normalization and ER Modelling.

# **Conclusion**

Having carried out research into the various database technologies available and weighed up the pros and cons, it seems that it would be best to proceed with the **relational database model** as it is best suited to the needs of our client.

As Coracle Christian Community are keeping their records mostly in written files, Excel and Microsoft Word Documents (rotas etc). It makes sense to bring all these together in one system where information can be accessed, entered and updated easily. Information on Church Members, Rotas, course information, new leaders, donations, vetting, training details can be updated when necessary.

Reports will be easily generated for all their organizational needs due the linking of the various tables of information with the necessary attributes. There will be less redundancy and errors.

New tables can be created easily and linked if required. This database could also be connected directly to their website using the navigation tools, regarding the courses they have available (future development).

# **Methodologies**

There are different types of methods used to develop databases systems. Developers use these methods according to the one best suited depending on the circumstances of the project and the scale of it. These various methodologies are used by the developer to insure that the project will meet its deadlines and various stages on time. These are explained below and also my choice of methodology.

These are the different models commonly in use:

* Prototype Model
* Incremental Model
* Agile Model
* Spiral Model
* Waterfall Model



## **The Prototype Model**

This type of model is where a mock form of the original system is created for testing by the user or customer. The point of this is to see if it will function according to their needs as they practically use it. It is also an opportunity for the user to suggest changes and improvements. It is a throw away version, that can take the form of illustrative mock screens or reports. It can be a simulation with basic or limited information using mock data. Sometimes using sample data or a functional prototype which has some of the finished features with actual data being used.



Pros

* The Prototype model gives the project the benefit of a system already in existence.
* The customer and the user is fully involved at all stages through actual sampling the system.
* Any problems or improvements can be flagged up by the user and acted upon.
* This gives the customer/user more control and less likely to be disappointed with end product.

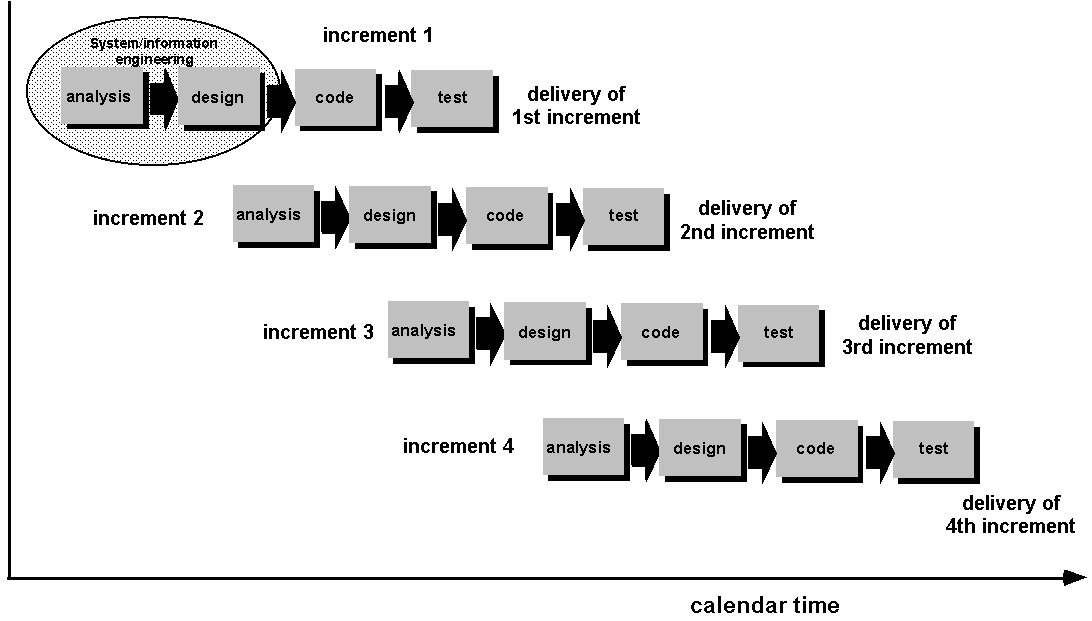
Cons

* As the proposed system is in existence in a simple form or is an older system that is to be adapted to the new needs, all the functionality requires testing and the design may require remodelling.
* The prototype may not be able to adjust to the new requirements. The older system may be inadequate or offer a reduced functionality.
* The Prototype model gives the project the benefit of a system already in existence but there may be additional expense and time in developing a functional system which may ultimately be made obsolete due the prototype not being adequate.

Prototyping is used in situations like business websites where it will require comments on the suitability from the customer. There is usually lots of customer participation to get the end product right.

## **The Incremental Model**

This model works through stage by stage design, build and testing. Any problems or recommendations are sorted through testing before moving on to the next stage. It has aspects of the prototype model and the waterfall method.

[](http://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwix4Jbm35vMAhWFuhoKHUvrCLYQjRwIBw&url=http://kulpreetwalia.blog.com/system-analysis-and-design/sdlc/incremental-model/&bvm=bv.119745492,d.ZGg&psig=AFQjCNHvd1XHRyGm3n7ylgVWkJRezij5YA&ust=1461191263352439)

Pros

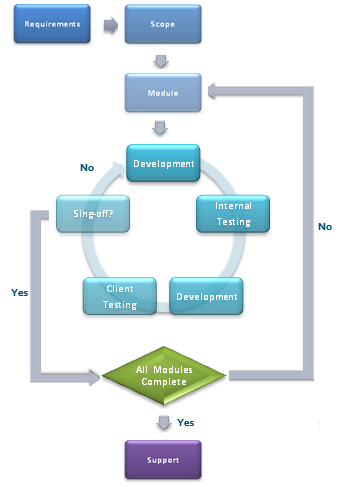
* This practical method that can be effected at both ends of the proposed system.
* Most problems are detected earlier on, helping to bring each phase to completion successfully.
* As each stage is fully completed it can be utilised right away.
* The initial costs are reduced as only parts of the system are designed, built and tested at a time.

Cons

* As each stage requires testing and proving, the users input is required frequently to correct any functionality problems before you can move on to the various stages.
* As each stage is costed individually, if the company runs out of capital then the system may not be completed to its full functionality.

## **Agile Model**

This model has some similarities with the incremental as there is a lot of user feedback. The user is part of the team that helps the design and development of the system but time is of the essence. This is a faster method with the system being completed in sections with full functionality allowing for use within the organization or business. Then further improvements and upgrades are planned and added as time goes on.



Pros

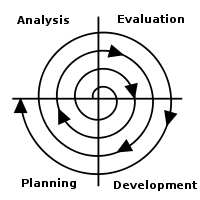
* The users are constantly involved in the development and are able to keep track of the new system. They receive updates to their system and see it grow continuously.
* As the user is in constant contact with developer last minute changes can be effected fairly quickly to the satisfaction of the customer.
* Any new user requirements can be scheduled into the next phase or updates.
* The agile method gives user functionality and focuses on producing a system that works in small and rapid stages.

Cons

* There is less focus on design and documentation and more on functionality.
* The agile method depends on close relationship with the customer and what they want, if the customer doesn’t know what they want it can stall the project.
* The rapid nature of this method means that the project may lose it direction due to moving away from the initial design.
* Constant updates to software means employing people with technical skills which can be bring more cost.

## **The Spiral Model**

The spiral method is used in high risk projects where there could be high costs involved. This model has four parts to it, Planning, Risk Analysis, Engineering and Evaluation. It is like the incremental model but there is more focus on the risks involved. The project goes through each step in what is called, ‘iterations’ or spirals. At the planning stage the user’s requirements are taken note of and all risks are taken into account. Usually a prototype is produced at this stage and any risks identified, steps are then taken and alternate solutions are found. This method is built upon over successive spirals, assessing the risk and finding solutions.



Pros

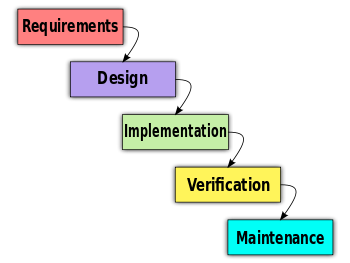
* Problems can be avoided by a strong focus on risk assessment.
* This process enables software to be produced in the first stages so that further improvements can be added.
* This model works well for larger costing projects which are large in scale, that are continually developing (long term).

Cons

* The first stage (risk assessment) - if the risks are not identified then it will carry on into the other phases.
* This model doesn’t work for smaller developments; it is mainly for larger costlier developments that are continually developing.

## **The Waterfall Model**

This method works for smaller projects with a definable scope. There is a flowing order to it in a methodical cascading process. Only after each stage is completed do you move on to the next phase. In this model the phases do not overlap. Testing is carried out at the end of the project. This was one of the earliest models.

[](http://www.google.co.uk/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=0ahUKEwjB1ND53ZvMAhWEmBoKHUAEDnkQjRwIBw&url=http://www.ceptara.com/blog/agile-scrum-waterfall-comparison&bvm=bv.119745492,d.ZGg&psig=AFQjCNEeYFc_viCo5sW4LcMsfBC04dVTDQ&ust=1461190768700060)

**Pros:**

* This model is the most straight forward to use, simple to grasp.
* Due to the definable scope it can be completed on time.
* You can work through each of the phases one after another with clear goals, no overlapping.
* Useful for small projects with well thought out goals.

**Cons**

* Due the cycle of this model the software is only available at the end.
* Not for projects that are costly or high risk of changing its requirements.
* Not good for long projects.
* Once the project has advanced it is harder to change things.
* Doesn’t work well with complex projects with changing needs or requirements.

## **Conclusion**

Having examined five different methodologies in use by developers, describing the pros and cons of these different life cycles. It is time to choose a methodology that will work best for our project. Without careful thought as to what System Development Life Cycle we use, or if no life cycle is used, our project could run off schedule or over the projected cost. This could cause a breach in our contract. Types of cycles;

* Prototype Model
* Incremental Model
* Agile Model
* Spiral Model
* Waterfall Model

The Prototype Model is not suitable method as there is no initial system to start with, also this system requires the end user to be present or available most of the time from the design to the testing. This would not be practical or necessary, as all requirements are made clear in the scope from the start.

The Incremental Model is not a suitable method given the size of the project it would not be necessary to test after each build as there is only one building phase. Excessive contact with the end user would not be required. Suitable for a middle size to large project.

The Agile Model being very similar in style to the Incremental Model, which requires constant contact with the user and is for middle to large scale development. As this project is small and the end user’s requirements are well documented, it will not require the constant presence of the end user. Therefore, it would not be suitable.

The Spiral Model would not be suitable for this project as it tends to be used for higher risk projects with higher costs and is used for longer term, more complex systems that require much more technical ability than is required for this system.

The Waterfall Model suits our project because we have carried out a detailed systems investigation, clearly understanding the problem definition. We can understand the scope of the proposed system, removing the need for constant access to the end user. The various phases should be easily set out and adhered to. The length of the project and its functions are definable and obtainable. A Gantt Chart will be used to assist in the development of this system.

# **Design Specification**

## **Purpose**

After receiving the problem definition and carrying out research into the current system and created a functional specification, we are now ready for the design specification. We will examine more closely the various entities and attributes and how these will link together to meet the specific needs of the organization.

As Coracle is a voluntary organisation I had no trouble with its members cooperating with my investigation as they were very keen to better their current system and explained to me with detail how their system works and made available any documentation.

This organization is still in its infancy in terms of its operating system and it is clear that Coracle would benefit from a better information system that will record the exact details of their members without having to update and maintain a manual record system. The computerized system will help them to effectively organise their volunteers, create rotas, store important information regarding child protection vetting and training details that can be accessed and updated.

Data Base ER Modelling techniques will be used to create the right kind relational data base to store this information thus removing the need to repeat and reduce any redundant information. The new system will also be designed to be user friendly for the organizations enthusiastic volunteers.

# **Initial Entities (ER Diagrams)**

There were initially six main entities

* Church Membership
* Course
* Donation
* Child Protection
* Church Duty Rota
* Youth Group

## **Church Membership (entity) -**

The main entity is the church membership; it contains the most information on church members. It was initially thought that we would have to make a separate leader entity table, however a small tick box will be inserted in church membership table to state whether church member is a leader or not. This church membership entity will be related to another entity named **‘Course’**. The course entity and the church membership entity cannot be directly related as this would create many to many relationships. Therefore, *another entity will be required called* **Course booking** *to bridge* the two tables (church member and course) together for organizing the dates and booking **(see intuitive ER Diagrams).**

**Church Duty Volunteer (entity)** – As a church member can volunteer for many voluntary roles.

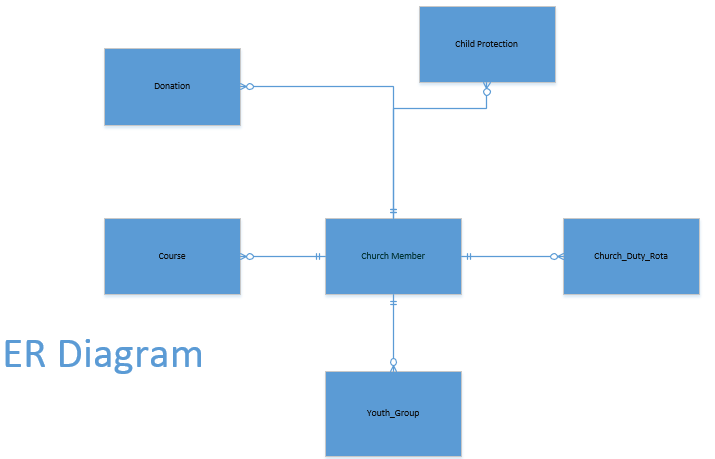
**Youth (entity)** – As youth are part of the church membership and can be part of more than one youth group.

**Donation (entity)** – the donation entity is related to the church member entity and as some donations are made by NON TAXPAYERS and some by TAXPAYERS therefore a yes or no tick box will be included in the design in order to prevent another table having to be created. This is seen in the next stage of the ER Diagram.

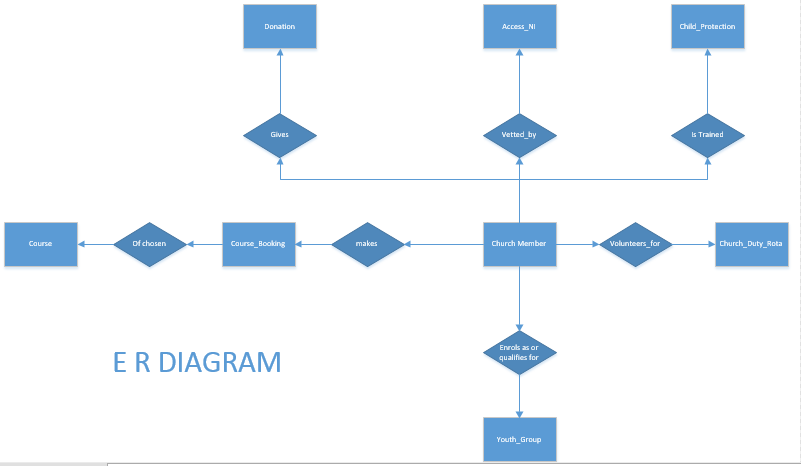
**Child Protection (entity)** – the church member is related to the Child Protection entity. However as there is a training aspect and a vetting aspect to child protection, there will be two tables one for Access NI vetting and another for training. This will remove redundancy.

**This means there will be a total of eight entities as seen in 2nd E R Diagram. See 1st and 2nd ER Diagram below.**

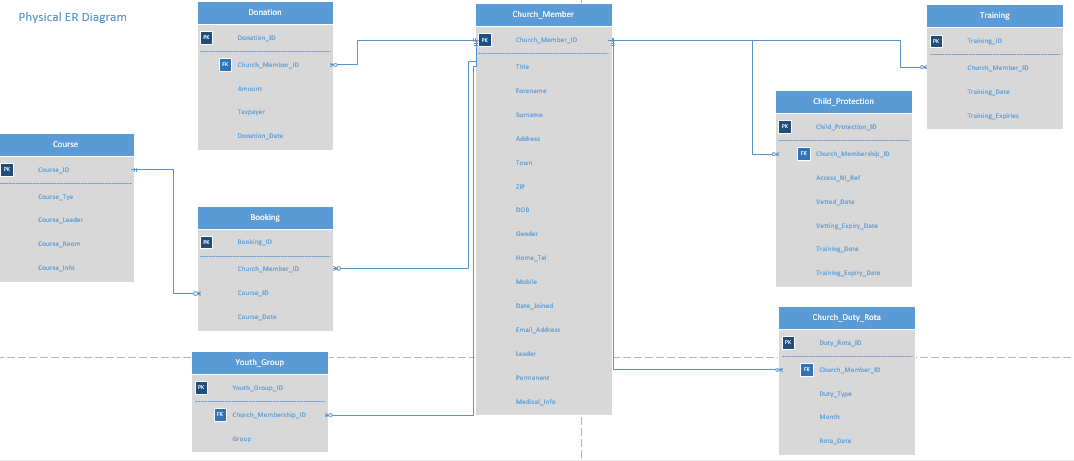
# **ER DIAGRAM (1)**



# **ER DIAGRAM 2**



# **PHYSICAL ER DIAGRAM 3**



# **Data Dictionaries & Queries**

Below there is a total of eight tables and nine queries that have been created. The data dictionaries detail the nature of these tables and their relationships and the queries into them. The queries are given first and then the tables. From these tables, their relationship and queries arise nineteen reports.

* Child Protection & Query
* Church Duty Rota & Query
* Church Member & Query
* Course Booking & Query
* Donation & Query
* Course & Query
* Training & Query
* Youth Group & Query

## **Child Protection**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_data\_Base.accdb*** | ***Purpose of Query – to check expiry of vetting dates.*** |
| ***Query Name*** | ***qryChild\_Protection\_Vetting*** | **This query will produce a report for the Child Protection Officer.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Church\_Member\_ID | Forename | Surname | Vetted\_Date | Vetting\_Expiry\_Date | Days\_To\_Expiry |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | Yes | Yes | Yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base1.***accdb*** | ***Table Name*** | **tblChild\_Protection** | ***(Composite)Key Field*** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Membership | Church\_Membership\_ID |  |  |

|  |
| --- |
| **General table description: Table containing information on persons relating to child protection.** |
|  |

**Related to**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Child\_Protection\_ID | P | Auto Number |  |  |  |  | Uniquely identifies a person involved in child protection | 71 |
| Church\_Membership\_ID | F | Number |  | 4 | 0000 |  | Uniquely identifies a church member | 27 |
| Vetted\_Date |  | Date/Time | Short Date |  | <=DATE() |  | Date that person was vetted. | 23/11//20015 |
| Access\_NI\_Ref |  | Short Text |  | 10 | >LLL099999 |  | Unique Access NI training Reference | ANI000089 |
| Vetting\_Expiry\_Date |  | Date/Time | Short Date |  |  |  | Date Person was trained | 20/04/2011 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

## **Church Duty Rota**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base.accdb*** | ***Purpose of Query – to produce a rota for church duties with dates*** |
| ***Query Name*** | ***qryChurch\_Duty\_Rota*** | **This query will produce a report for a church leader to organize various volunteers.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Church\_Membership\_ID | Forename | Surname | Duty\_Type | Rota\_Date |  |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | Yes | Yes | Yes  No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base.***accdb*** | ***Table Name*** | **tblChurch\_Duty\_Rota** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Member | Church\_Member |  |  |
|  |  |  |  |

|  |
| --- |
| **General table description: To store details of which church members are on which duty and on what date.** |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Duty\_Rota\_ID | P | Auto Number |  |  |  |  | Uniquely identifies a Rota Slot | 6 |
| Church\_Member\_ID | F | Number |  |  |  |  | Unique identification of Church member | 10 |
| Duty\_Type | F | Short Text |  |  | Drop List |  | Uniquely identifies a duty | 2 |
| Rota\_Date |  | Date/Time | Short Date |  |  |  | Date when church member is on the rota | 10/10/2016 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base1.accdb*** | | ***Purpose of Query – find out member’s contact details, whether they are a leader. Also if permanent member.*** |
| ***Query Name*** | ***qryChurch\_Member*** | **This query will produce reports for church leaders and church administrators.** | |

## **Church Member**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Church\_Member\_ID | Forename | Surname | Address | Town | ZIP |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes  No | Yes  No | Yes  No | Yes  No | Yes  No | Yes  No |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Home | Mobile | Date\_Joined | E\_Mail\_Address | Permanent | Leader |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted |
| ***Show***  (tick) | Yes  No | Yes  No | Yes  No | Yes  No | Yes  No | Yes  No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base1.***accdb*** | ***Table Name*** | **tblChurch\_Member\_ID** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChild\_Protection |  | tblDonation |  |
|  |  |  |  |

|  |
| --- |
| **General table description: This table stores information on church members.** |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Church\_Member\_ID | P | Auto Number |  | Long Integer |  |  | Uniquely identifies a church member | 36 |
| Title |  | Short Text |  | 6 | Drop Down List |  | Title of person | Mr.Mrs |
| Forename |  | Short Text |  | 25 |  |  | Members First Name | John |
| Surname |  | Short Text |  | 30 |  |  | Members Surname | Thompson |
| Address1 |  | Short Text |  | 30 |  |  | 1st line of members address | 22 Lindon Way |
| Town |  | Short Text |  | 20 |  |  | Town of members address | Dunmurry |
| ZIP |  | Short Text |  | 8 | >LL09 0LL |  | Customers Post Code | BT17 0TH |
| DOB |  | Date\Time | Short Date | 11 | <=DATE() |  | Date Customer was born | 12/10/1992 |
| Gender |  | Short Text |  |  | Drop Down List |  | Gender of Church Member | Male/Female |
| Home\_Telephone |  | Short Text |  | 13 | 00000 000 000 |  | Home telephone number | 02890 875 673 |
| Mobile |  | Short Text |  |  | 00000 000000 |  | Mobile number of Church Member | 07084367843 |
| Date\_Joined |  | Date/Time |  |  | <=DATE() |  | Date Church Member Joined | 18/06/2005 |
| Email\_Address |  | Short Text |  | 30 | Is Null Or ((Like "\*?@?\*.?\*") And (Not Like "\*[ ,;]\*")) |  | E mail address of member | JMulvenny@live.co.uk |
| Permanent |  | Yes/No |  |  |  |  | Identifying if member is Permanent or Temporary | Permanent |
| Leader |  | Yes/No |  |  |  |  | Identifying if a person is a leader or ordinary member | Leader |
| Medial\_Info |  | Short Text |  | 100 |  |  | Short note on any medical information on member | Nut allergies |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

## **Course Booking**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base.accdb*** | ***Purpose of Query – to provide information on courses scheduled with details on leaders, persons attending, date and room number.*** |
| ***Query Name*** | ***qryCourse\_Booking*** | **This query will produce a report for church leaders as to their schedules for courses.** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |  |  |
| ***Field*** | Church\_Member\_ID | Forename | Surname | Course\_Type | Course\_Leader | Course\_Start\_Date | Course\_Time | Course\_Room |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Church\_Data\_Base1.***accdb*** | ***Table Name*** | tblCourse\_Booking | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Member | Church\_Member\_ID | tblCourse | Course\_ID |
|  |  |  |  |

|  |
| --- |
| **General table description: This table stores information on course bookings for church members** |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Booking\_ID | P | Auto Number |  |  |  |  | Uniquely identifies a course booking | 121 |
| Church\_Member\_ID | F | Number |  |  |  |  | Unique identity of church member | 86 |
| Course\_ID | F | Number |  |  |  |  | Uniquely identifies a course | 15 |
| Course\_Date |  | Date/Time | Short Date |  |  |  | Unique date of booking | 22/10/2016 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base.accdb*** | ***Purpose of Query- to calculate the amount of donations made by tax payers and Non Taxpayers.*** |
| ***Query Name*** | ***qryDonation*** | **This query will produce a report for church treasurer on donations.** |

## **Donation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Donation\_ID | Church\_Member\_ID | Donation | Taxpayer | Donation\_Date |  |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | Yes | Yes | Yes  No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base1.***accdb*** | ***Table Name*** | **tblDonation** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Member | Church\_Member\_ID |  |  |
|  |  |  |  |

|  |
| --- |
| **General table description: To store information on donations, including amount, date and tax payer or non taxpayer.** |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Donation\_ID | P | Auto Number |  |  |  |  | Uniquely identifies a donation | 6 |
| Church\_Membership\_ID | F | Number |  |  |  |  | Unique identification of church member | 82 |
| Donation |  | Number | Currency | 6 |  |  | Uniquely identifies a donation made by a tax payer | £120.00 |
| TAXPAYER |  | Yes/No |  |  |  |  | Uniquely identifies a donation made by a taxpayer or non-tax payer | Tick Box |
| Donation\_Date |  | Date/Time |  |  |  |  | Date that donation is made | 10/02/2016 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

# **Course**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Date\_BAse.accdb*** | ***Purpose of Query – to view courses on offer with course leader, start date*** |
| ***Query Name*** | ***qryCourse\_Detail*** | **This query will produce a report for church leaders on courses being offered with proposed leader and course details and dates.** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Table*** |  |  |  |  |  |  |  |  |
| ***Field*** | Course\_ID | Course\_Type | Course\_Leader | Course\_Room | Course\_Information | Course\_Start\_Date | Course\_End\_Start | Course\_Time |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | No | Yes | Yes | Yes | Yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base1.***accdb*** | ***Table Name*** | **tblCourse** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblCourse\_Booking | Booking\_ID |  |  |

|  |
| --- |
| **General table description: This table contains information on the types of courses available.** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Course\_ID | P | Auto Number |  |  |  |  | Uniquely identity of course | C28 |
| Course\_Type |  | Short Text |  | 30 |  |  | The name of the course | Alpha |
| Course\_Leader |  | Short Text |  | 25 |  |  | Uniquely identifies a leader (tutor) | Norman Gibson |
| Course\_Room |  | Short Text |  | 25 |  |  | Unique Room to attend course | 6 |
| Course\_Info |  | Short Text |  | 150 |  |  | Information on course | Initial exploration of Christianity for beginners |
| Course\_Date |  | Date/Time | Short Date |  | 00/00/0000;0; |  | Unique date of booking | 13/10/2016 |
| Course\_Time |  | Date/Time |  |  | 00:00;0; |  | Time of course | 12:00 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

## **Training**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base.accdb*** | ***Purpose of Query- To find out which members have received training and when their training expires.*** |
| ***Query Name*** | ***qryTraining*** | **This query will supply information for the training report, read by Child Protection officer.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Training\_ID | Church\_Member\_ID | Training\_Date | Training\_Expires | Days\_To\_Expiry |  |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted |
| ***Show***  (tick) | Yes | Yes | Yes | Yes | Yes | Yes  No |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base.***accdb*** | ***Table Name*** | **tblTraining** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Member | Church\_Member\_ID |  |  |

|  |
| --- |
| **General table description: To store information church member, child protection training dates and expiry.** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Training\_ID | P | Auto Number |  |  |  |  | Uniquely identifies a Volunteer who requires training | 28 |
| Church\_Member\_ID | F | Number |  |  |  |  | Uniquely identifies a church member | 16 |
| Training\_Date |  | Date/Time | Short Date |  |  |  | Date that training is carried out. | 22/02/2016 |
| Training\_Expires |  | Date/Time | Short Date |  |  |  | Date that training expires and retraining is required. | 22/02/2019 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

## **Youth Group**

|  |  |  |
| --- | --- | --- |
| ***Database File*** | ***Coracle\_Data\_Base.accdb*** | ***Purpose of Query- to find out which groups children are in within the church and their numbers and ages.*** |
| ***Query Name*** | ***qryYouth\_Group*** | **This query will provide a report on which groups that youth attend for the purposes of organizing volunteers due to youth numbers.** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Tables*** |  |  |  |  |  |  |
| ***Field*** | Youth\_Group\_ID | Church\_Member\_ID | Group | Forename | Surname | DOB |
| ***Sort***  (tick) | Ascending  Descending  Not Sorted | Ascending  Descending  Sorted | Ascending  Descending  Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted | Ascending  Descending  Not Sorted |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Database File*** | Coracle\_Data\_Base.***accdb*** | ***Table Name*** | **tblYouth\_Group** | ***(Composite)Key Field*** |  |

**Related to:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Table Name*** | ***Foreign Key*** | ***Table Name*** | ***Foreign Key*** |
| tblChurch\_Member | Church\_Member\_ID |  |  |
|  |  |  |  |

|  |
| --- |
| **General table description:** |
| This table holds data about which groups youth are part of for the purposes of organizing volunteers according to youth levels. |

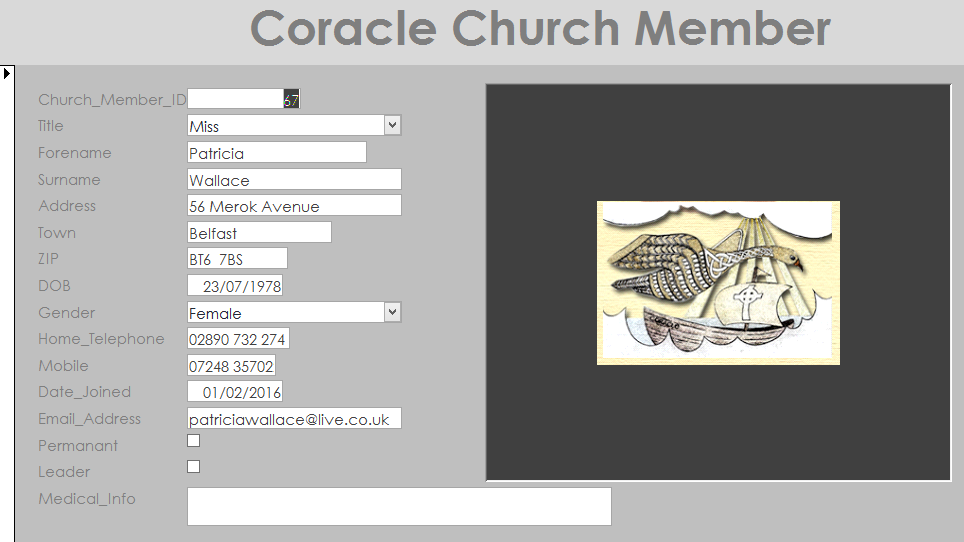
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Field Name*** | ***Key*** | ***Data Type*** | ***Format*** | ***Length*** | ***Input Mask/Validation Rule/Validation Text*** | ***Default Value*** | ***Description*** | ***Typical Data*** |
| Youth\_Group\_ID | P | AutoNumber |  |  |  |  | Uniquely identifies the group that church member attends | 23 |
| Church\_Member\_ID | F | Number |  |  |  |  | Uniquely identifies the church member | 66 |
| Group |  | Short Text |  |  | Drop List |  | The group that child attends | Drama |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Legend:** | ***Key*** | **P** | (Primary) | **F** | (Foreign) |  |  |

# **Prototype Testing**

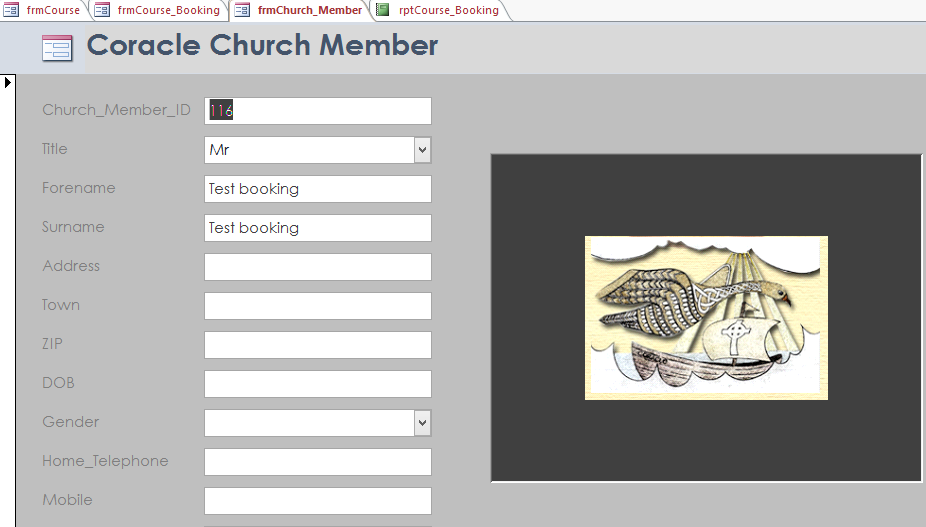
The tables have been designed and built according to their relationships in the Physical ER Model. The forms, queries and reports have been designed, but before they are put into the Navigation Menu (**User Interface**), we must get some feedback from our client.

I **made contact** with Coracle Christian Community and spoke their main administration leader and treasurer (Norman Gibson). I showed him the proposed **Prototype system** to try and get some indication if we were fulfilling their **specifications** and to have the end user carry out a few trials, below is typical **Form**.

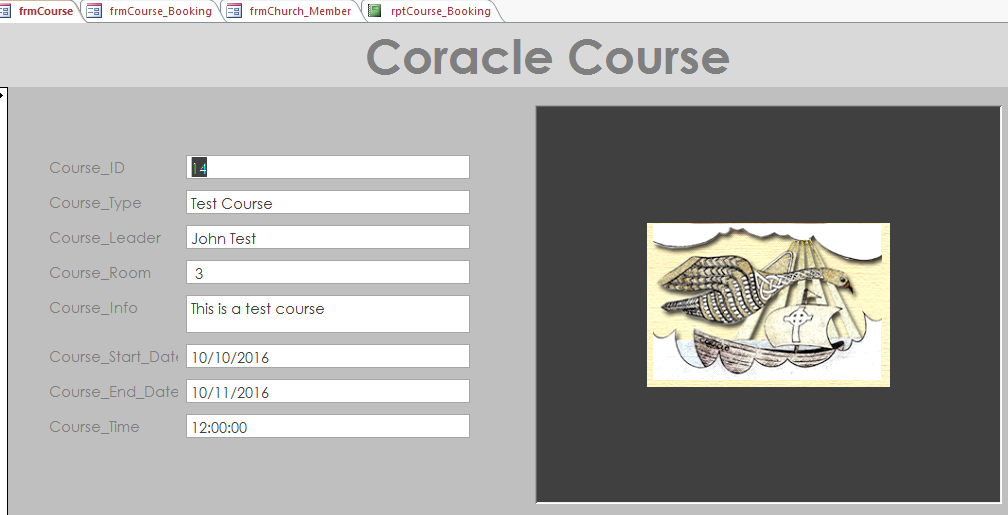


## **Prototype & Feedback**

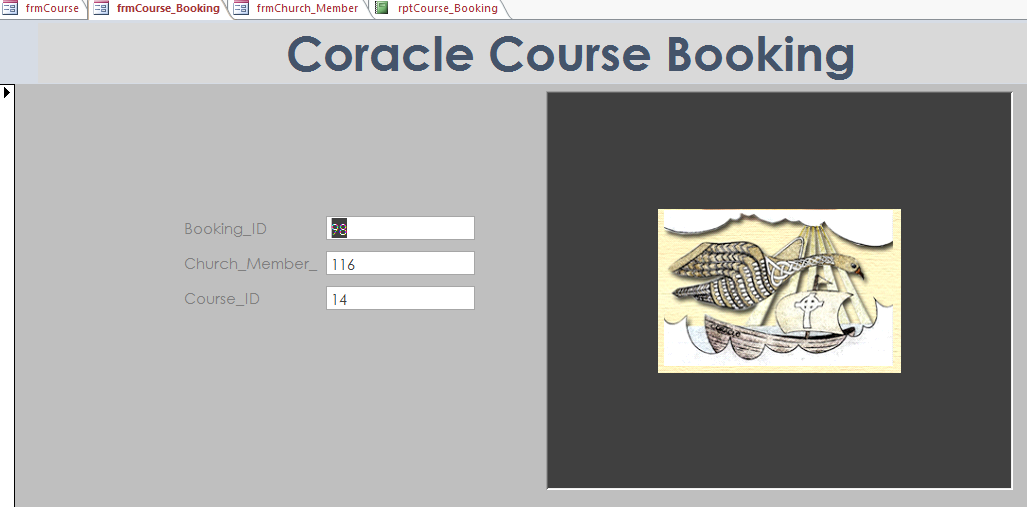
Our client was quite impressed with the type of fields and the design of the forms, especially as each one had the Coracle logo. Having looked over these forms and some of reports. I gave the client a test form to fill in using the ‘**Coracle Church Member’** form, the client was able to do this easily.



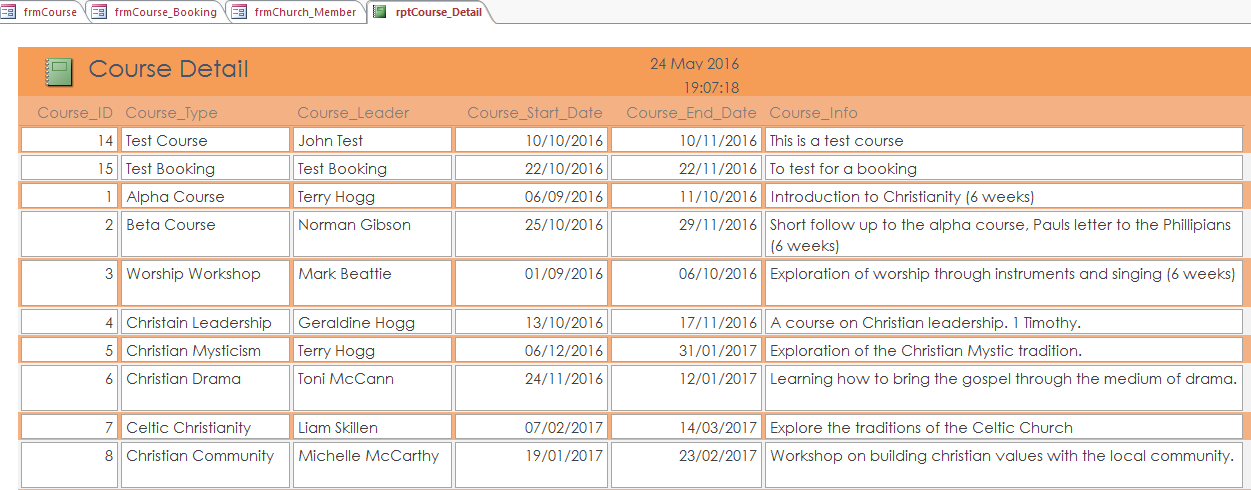
Having created a test Coracle Church Member (116). I then had the client create a fictional test ‘**Course’,** in order create a Course ID (14) to be used in the, ‘Coracle Course Booking’, as a test booking.

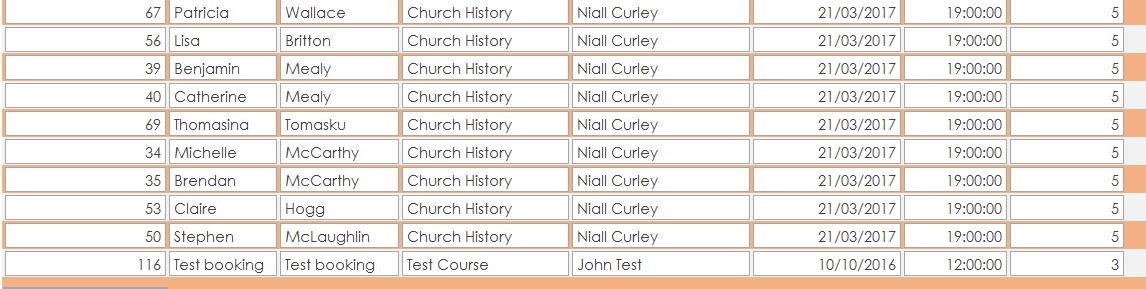


I then showed the client how to book a course in the ‘Coracle Course Booking’ form by bringing the ‘**Church\_Member\_ID (116)’** and the ‘**Course\_ID (14)’** and imputing them to create the **Booking ID (98).** This was done easily by the user.



I then showed the client the report for Course ID (14), and the report for test booking for Church Member (116).

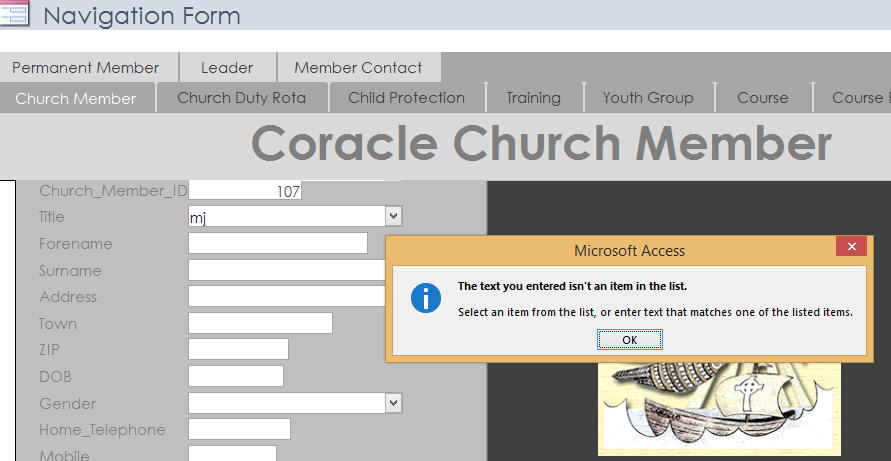




The client was generally impressed with the product wanted to know if this would be finished on schedule. This prototype testing was carried out on the 24 April 2016 which is on schedule project finish date. Having now carried out some trials with our Prototype and got positive feedback, it is now necessary to test this system and see if there is any problems or adjustments to be made before the Database is handed over to the End User.

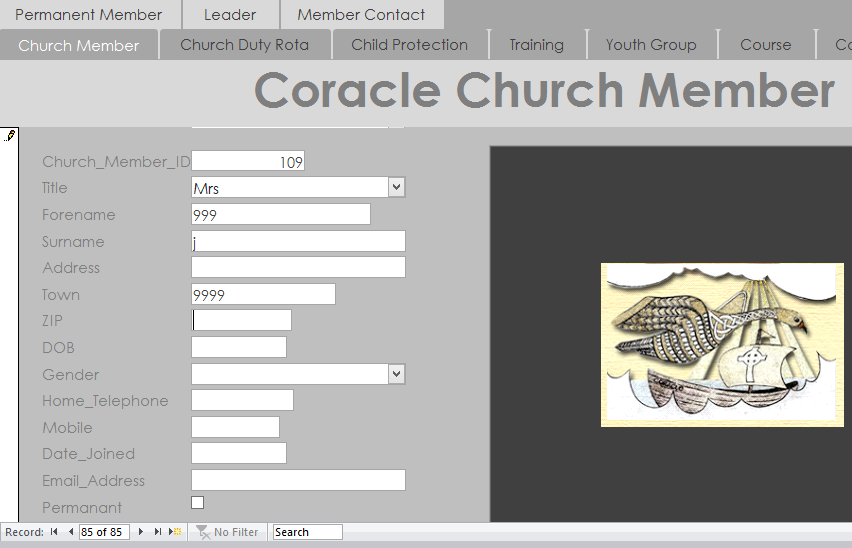
# **Unit Testing**

## **Test 1 Church Member Form**



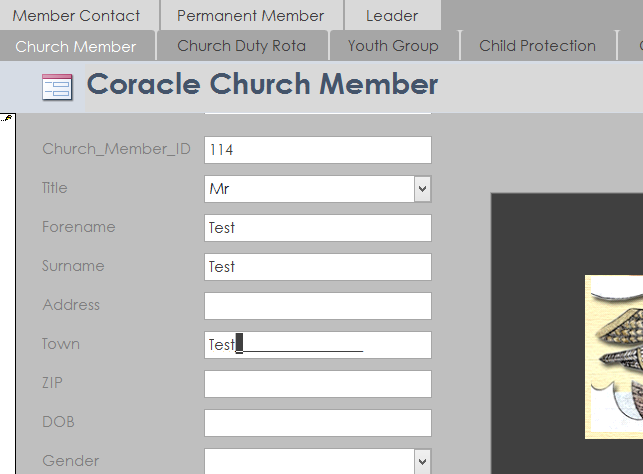
Test 1 – Attempting to enter incorrect data in the Church Member form **‘title’** (below) produces a message from the system, **‘the text you entered isn’t an item in the list’.** This means that only the limited data that was entered into the drop menu can be entered. Therefore the test has proved that this part of the system is functioning as it should.

## **Test 2 Church Member Form**



Test – 2 During this test I was able to enter numeric data into **‘Forename, Surname, Town’** only letters should be applicable here. **Also there is no capital at start of forename, surname or town’**. This means that there needs to be changes in the input masks.This will mean going back to original ‘Church Member’ table and changing the input masks for, ‘Forename, Surname and Town to >L<???????????????????.

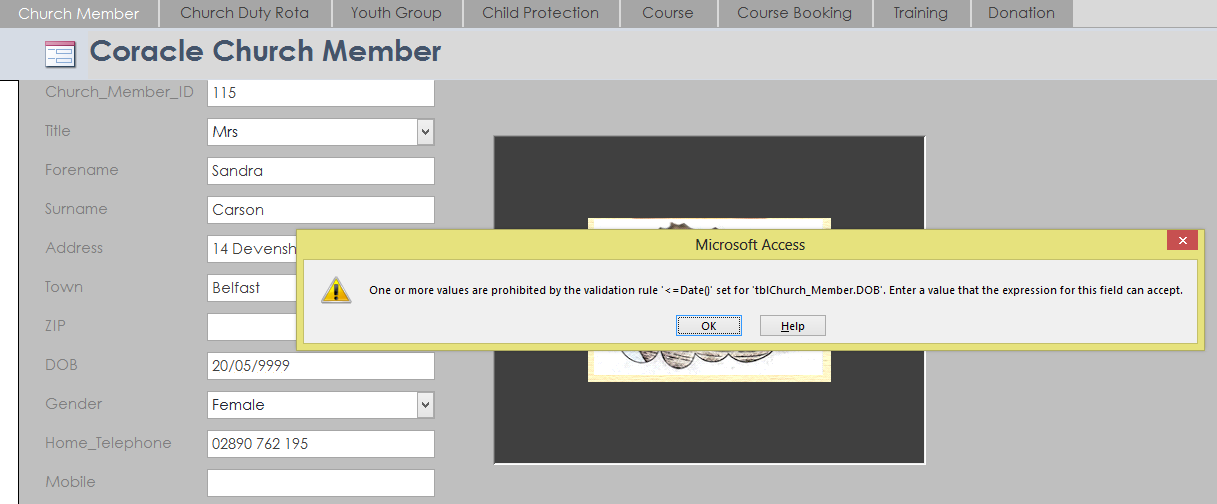
## **Rectifying Test 2**



It was necessary to delete original form and create a new one, also to make changes in the ‘**Church Member’**, table by imputting information into the input masks for, ‘Forename, Surname and Town. This has resulted in these fields, not accepting numeric values and also beginning each with a capital letter as demonstrated in the above.

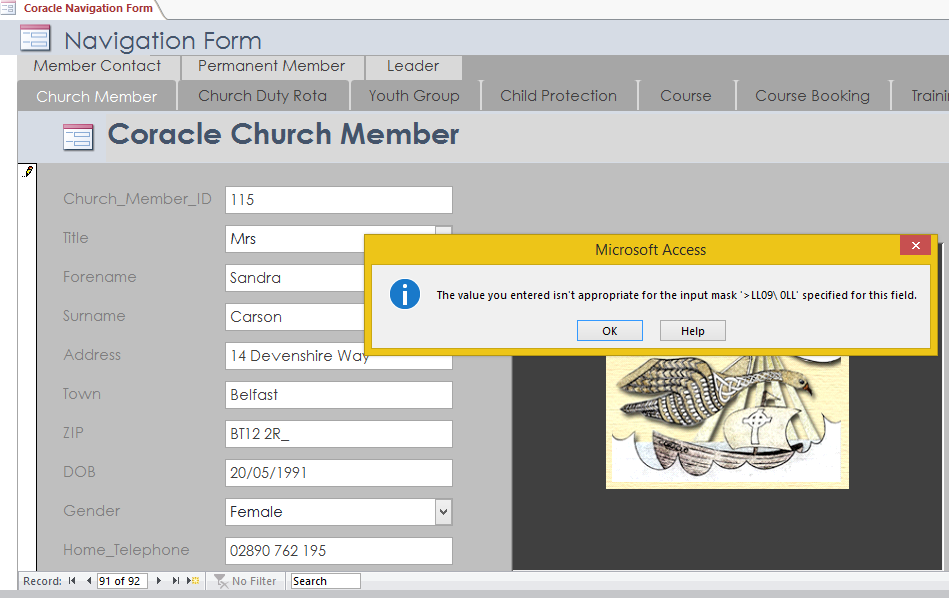
## **Church Member - Test 3 (DOB)**

When entering data into the field for DOB (20/05/9999), that was greater than the validation rule, this message is displayed. Therefore this validation rule is proven to work correctly so that incorrect data cannot be entered.



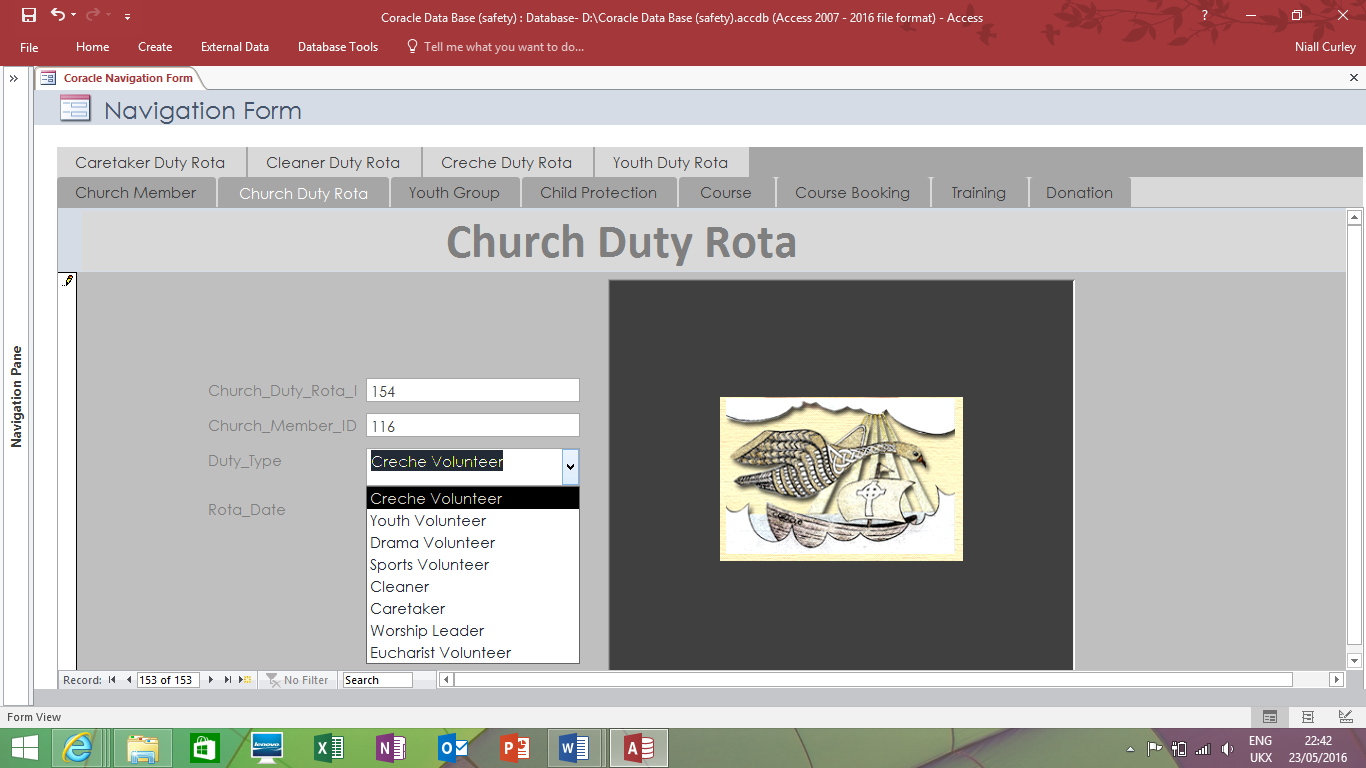
## **Church Member – Test 4 (ZIP)**

In the example below when incorrect data is entered into the field (ZIP) post code, this message appears (The value you entered isnt appropriate for the input mask >LL09 0LL), which means it only allows you to enter two capital letters the one compulsory numeric and optional numeric, then one compulsory numeric and two capitals. This proves that only the correct data can be entered.



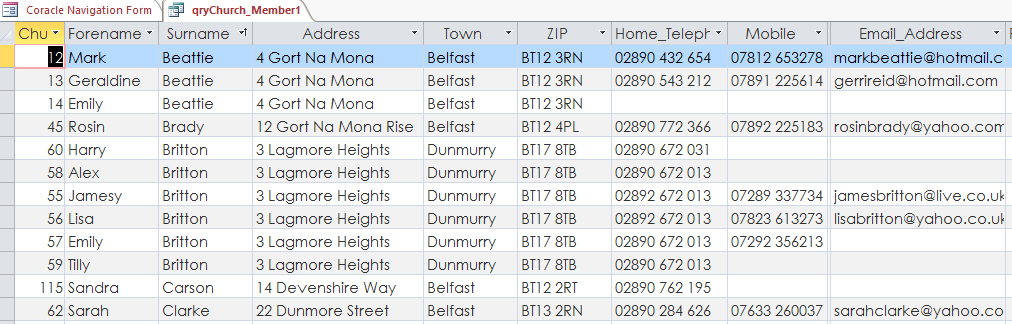
## **Church Duty Rota – Test 5**

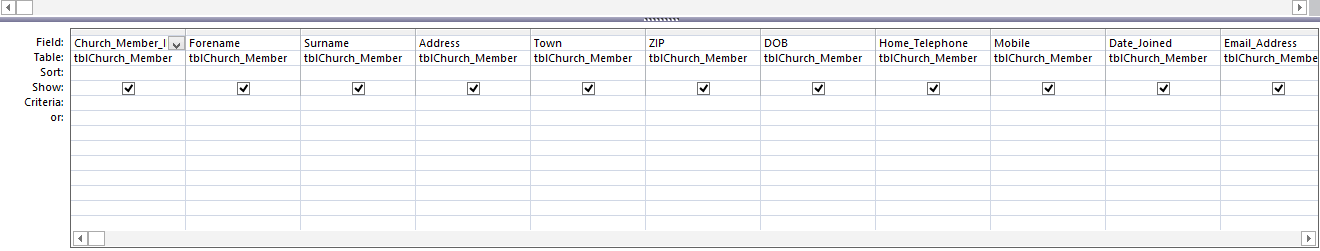
In the **‘Church Duty Rota’** the drop down menu works as designed allowing you to choose from the selection of roles. This speed up the Duty Rota when allocating the different roles.



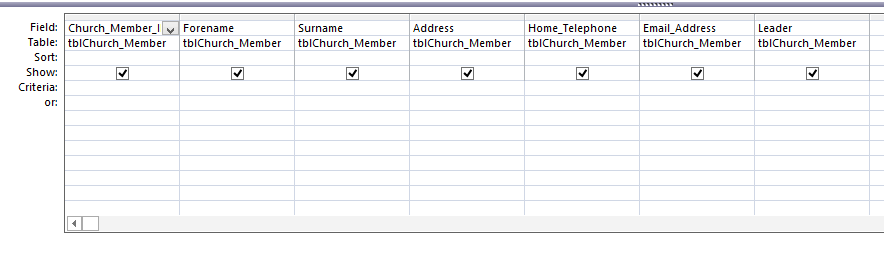
# **Test Queries**

**Church Member1** – the point of this query is to provide contact details for any church member. Sorting according to surname.



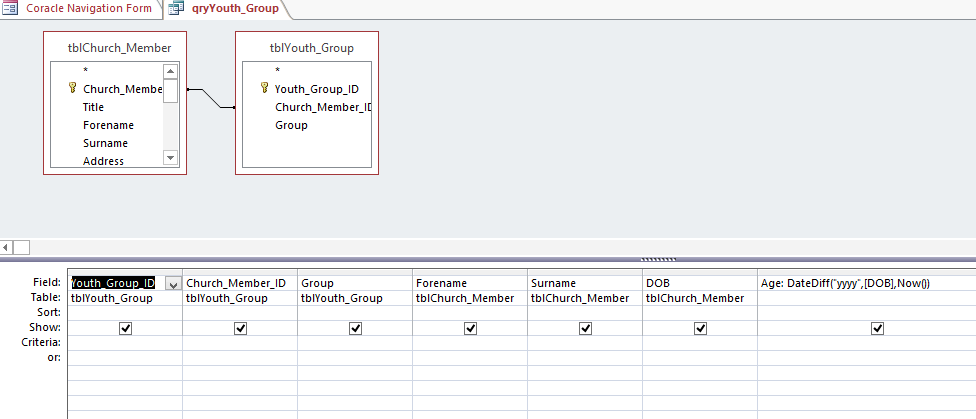


**Church Leader** – the point of this query was to filter the Church Member table to only give details of leaders. Sorting according to tick box.





**Youth Group** – the point of the query was to produce a list of youth and the activities they would be part of. Also to calculate their age according to their date of birth.



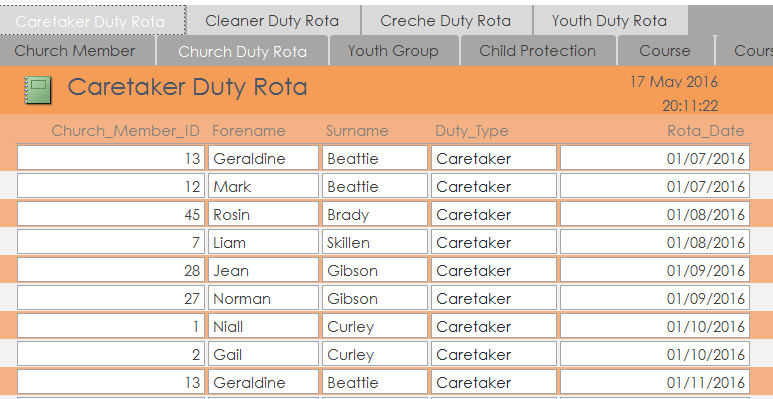
As you can see below the query has met these requirements.



## **Testing Reports**

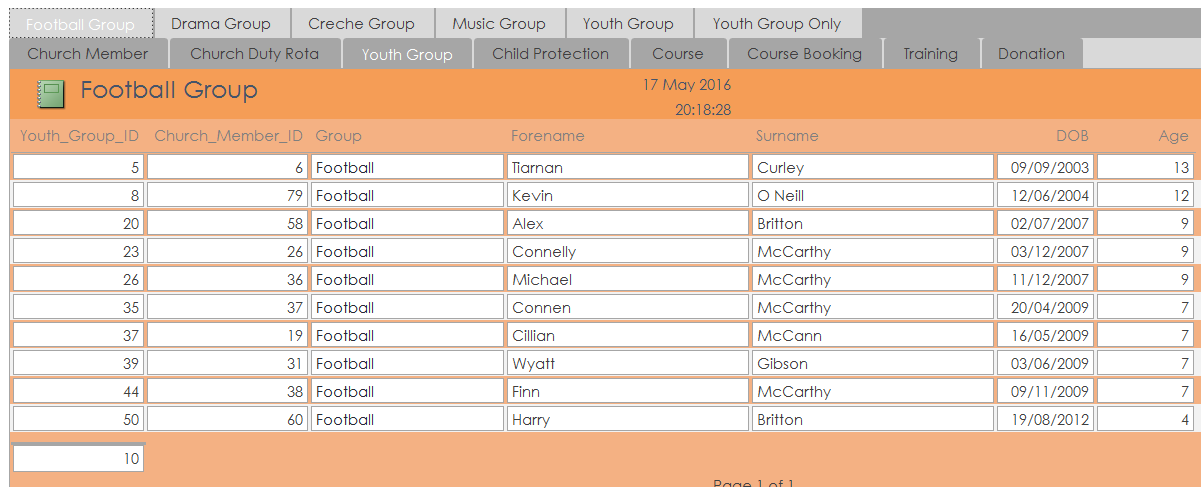
There is a total of nineteen reports, we will now look at some of these reports to see if they have fulfilled the requirements set out by our customer.

**Caretaker Duty Rota**



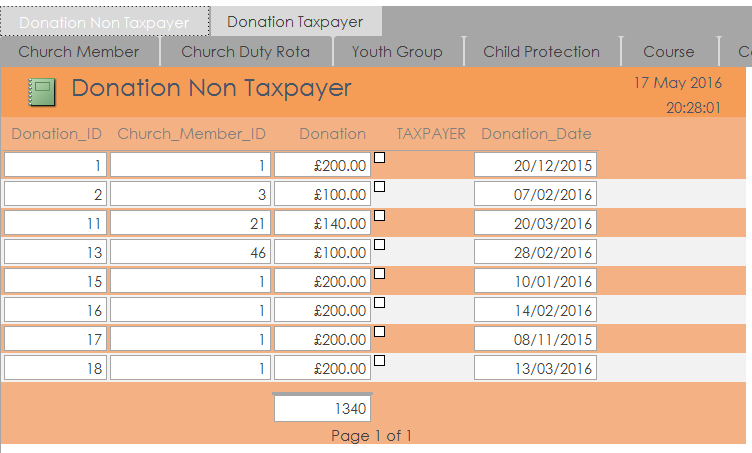
The names of each volunteer is displayed with the specific Duty Type (Caretaker) and the date on which that volunteer will begin their duty for that month. This report has fulfilled its criteria in the filtering of the Duty Type.

## **Youth Group**



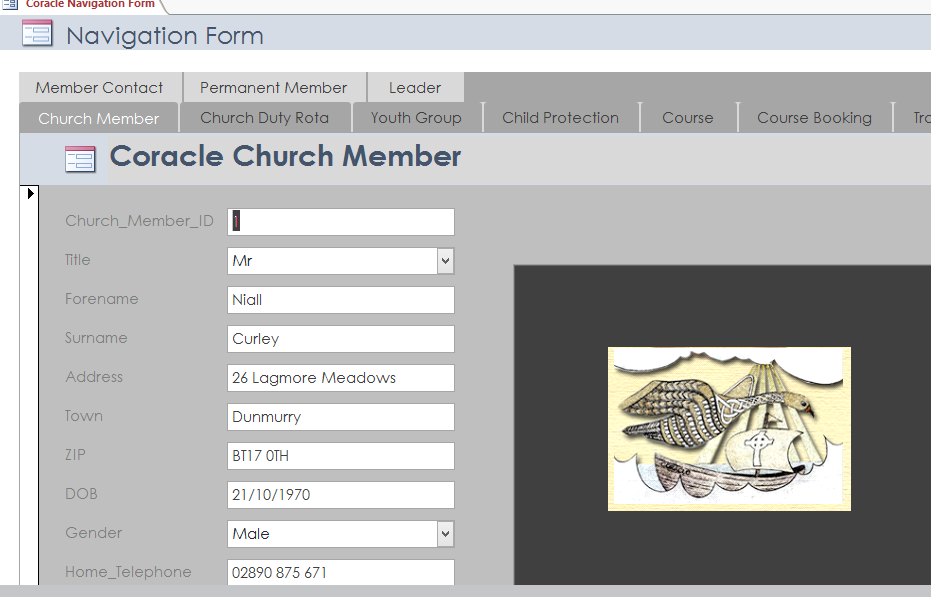
This report accessed by pressing the ‘Youth Group’ tab, displays which youth are in a certain group in this case ‘Football Group’ and what their age is, this means that the correct filtering has been applied in the ‘Group’ and ‘Age’ fields. This is useful for estimating number of volunteers for supervision etc.

## **Donation**



This report accessed by pressing the ‘Donation’ tab displays information on which Church Member has made a donation, on what date and how much was donated. It also displays correctly whether that person was a taxpayer or not. The correct filtering has been used to discern which donations are from taxpayers or not. This helps the church treasurers to use this specific information claiming under their charitable status.

# **Acceptance Testing (Navigation Form)**

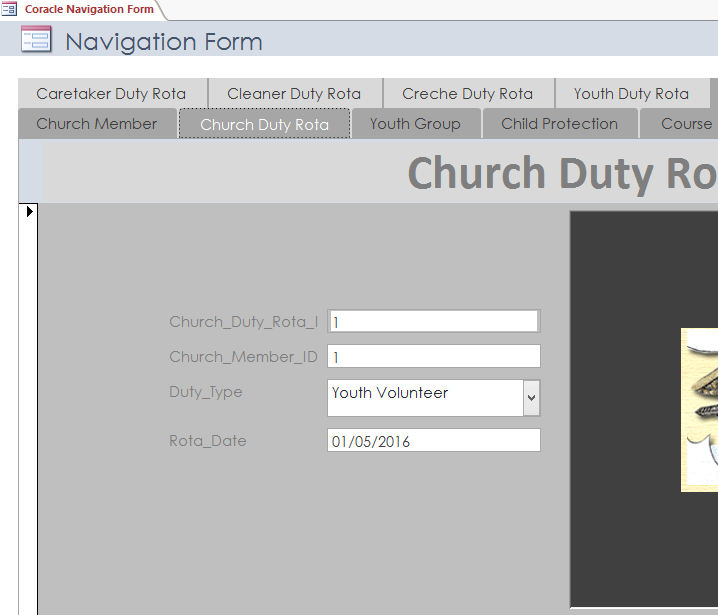


**Testing Navigation Form** - There are eight main tabs on the main navigation screen – Church Member, Church Duty Rota, Child Protection, Training, Youth Group, Course, Course Booking, Donation. When each tab is pressed a number of reports will appear above each.

When testing the **Church Member tab**, three reports appeared above as expected (Permanent Member, Leader, and Member Contact, as above).

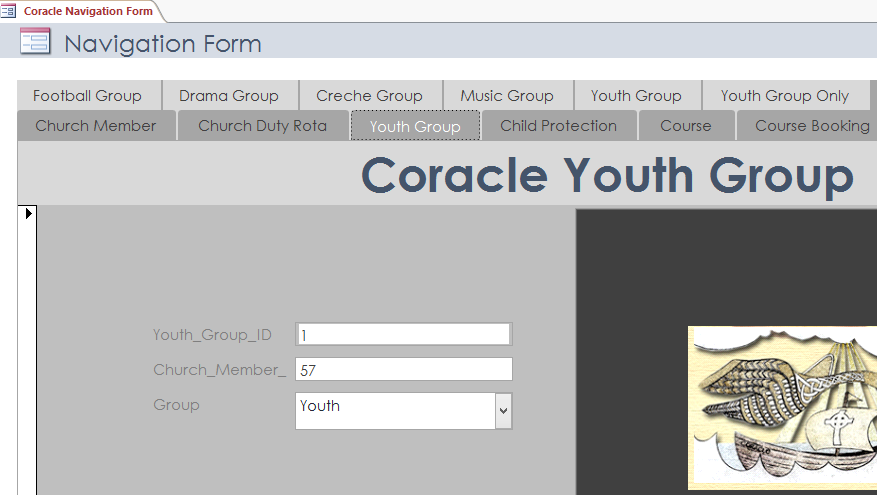
# **Church Duty Rota (Test)**

When testing the Church Duty Rota tab, four reports appeared above as expected, (Caretaker Duty Rota, Cleaner Duty Rota, Creche Duty Rota,Youth Duty Rota).



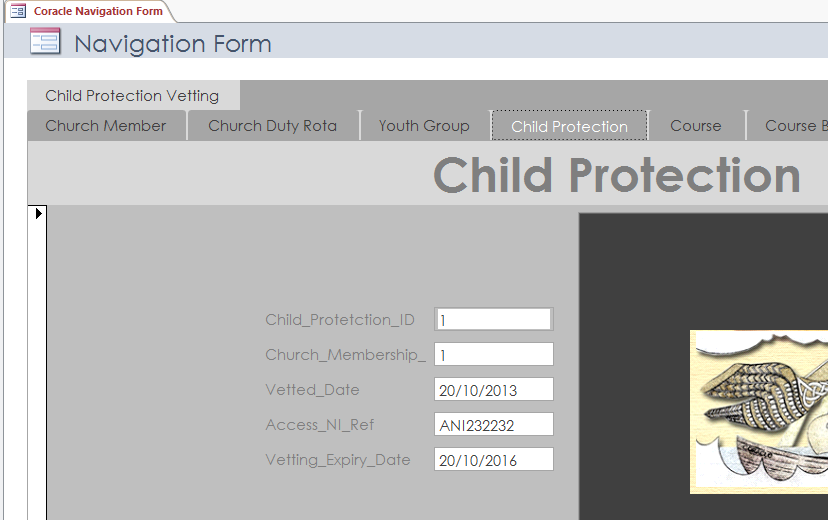
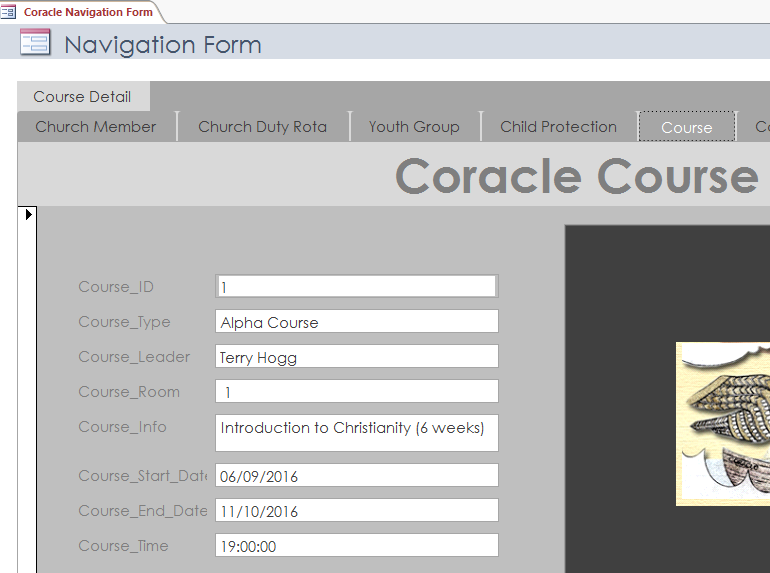
## **Youth Group (Test)**

When testing the Youth Group tab, six reports appeared above (Football Group, Drama Group, Creche Group, Music Group, Youth Group, Youth Group Only).



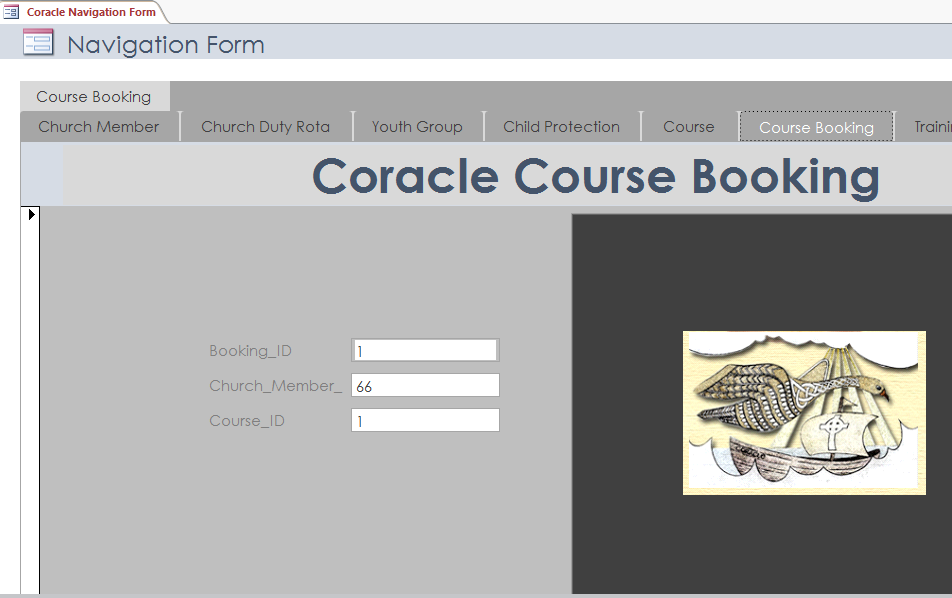
## **Child Protection & Course (Test)**

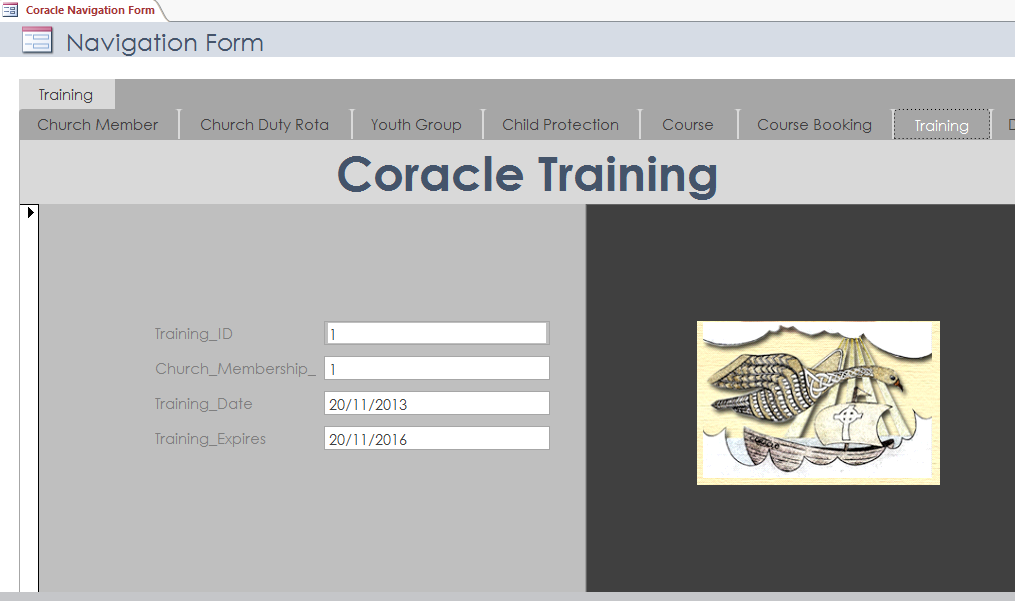
When testing the Child Protection tab, one report appeared above as expected (Child Protection Vetting). When testing the **Course Booking tab**, one report appeared above (Course Booking).



## **Coracle Course Booking & Training & (test)**

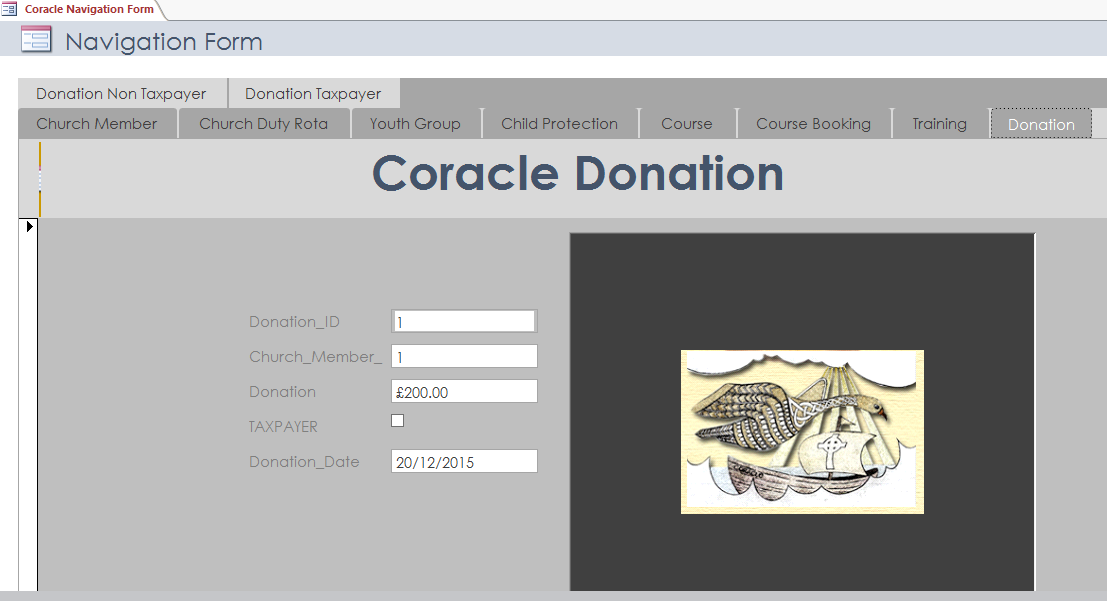
When testing the **Course Booking tab**, one report appeared above **(Course Booking).** When testing the **Training tab**, one report appeared above **(Training).**





## **Coracle Donation**

When testing the Donation tab, two reports appeared above (Donation Non Taxpayer & Donation Taxpayer)



# **Evaluation**

## **Introduction**

Our company was approached by Coracle Christian Community to provide them with a solution to their organizational problems. They wanted a system that could store details of their church membership, their permanent members and temporary members. They required the ability to organize courses with a booking system, a record of their child protection, a rota system for their volunteers and a financial section for their donations. At first the sheer number of areas covered in such a system seemed daunting as the original system was basically manual, and also because of my own limited experience. With a well thought out plan to investigate this organization, to interview their leaders, to examine any documentation and with hands on experience, I felt I could obtain accurate information to design and build the system envisioned.

## **Systems Investigation**

At first I carried out an initial investigation into the churches website and with the website of a similar church group this helped to get a feel for this type of organizations activities, I gleaned from this that a lot of these organizations have some kind of child protection set ups, records of their church membership and various types of volunteers. Later when I met the leaders and administrators of this church I prepared a questionnaire with specific questions to understand their needs. Through this questionnaire I understood that they needed to store information on their church members which would include details on adults and children, this meant that any table would have to include date of birth and be able to work out the ages of the church members. They also needed the ability to organize courses for new member’s and existing ones so any new system would need to know the difference between permanent members and temporary ones which would require a tick box in the new system. I also examined two documents, the statement of public benefit and the rota system they had on view for their volunteers and from these I realized that the proposed system would not only organize the churches volunteers but also its youth as to what youth groups they attended, so that the church could work out how many volunteers would be required. Reports would be required to be printed and viewed by the organization. Vital child protection details would also need to be stored which could work out from the vetting/training dates when the expiry and revetting/training dates would be required. Various rules would need to be inputted into the tables to enable this. The financial side of the churches activities required the ability to differentiate between taxpayers and non-taxpayers, that could be accomplished through simple tick box. Having considered the various methodologies, to see which System Development Life Cycle was appropriate for this project. The waterfall method was most suited due to the very clear parameters for the project provided at the systems investigation, and because I would not need constant client participation.

## **Data Analysis**

Armed with all the information I had taken from the questions asked and the *little* documentation I now had due to their system being very manual. I understood that I did not have enough documentation to carry out the Normalization approach, where clear entities and fields would have been more easily discerned. So I intuitively worked through the information and I was able to discern there were six main entities required for this organization; **Church Member, Course Booking, Donation, Church Duty Rota, Child Protection and Youth Group.** I decided that intuitively working with ER Modelling would be the best way to work out the new system. By this I soon realised that the system would need an extra entity called **‘Course’** to bridge between the booking table and the Church Member entity. Also that another entity called, **‘Training’** would be needed with the **Child Protection** vetting, as there is a difference between records on vetting and on training. By putting these two extra entities in it would reduce any redundancy in the system and organize its records better.

## **Design**

Having understood the requirements for type of reports from the interviews and documentation. It was necessary when producing data dictionaries that the input masks and validation rules where right in order to fulfil the specifications of our client. Whilst producing data dictionaries for Church Duty Rota, drop boxes were entered for easier user input. For queries for child protection and training it was necessary to be able to calculate the difference between a completed vetting/training date and the expiry date for vetting and training. This rule was added into the queries section - [Training\_Expires]-Date(). Tick boxes were included in Church Member to differentiate between permanent members and leaders. In the query into the Youth Group, the rule, DateDiff("yyyy",[DOB],Now()) was entered to calculate the ages of the children for organizing volunteers etc. Drop boxes were used in the Church Duty Rota to speed up data input for the user.

## **Testing of System**

I contacted the client with the prototype and carried out a few trials with the proposed system. The client seemed to be able to use the system quite well and was particularly happy with the logo. They system seemed to cover most of the clients functional requirements **(MSCW)**. Most of the, ‘must haves’ were met. It was recommend that in the Donation tables that it **‘should have’** a distinction between taxpayers and non-taxpayers, this was achieved. The client was also told that the system ‘**should have’** distinctions between different types of volunteers and this was achieved. Also it was that the system, ‘**could have’** different levels of security for certain personnel e.g (child protection and donations), but only a basic overall password was achievable in the time frame.

## **Evaluation**

The database turned out just like the ER Model, it works well and allows the manipulation of the data through the relational model chosen. With an exception of a few changes to the data dictionary for instance in the **‘Church Member table’** where the input masks were not correct and had to be changed in the **forename, surname and town** fields, because *numeric* data could be entered and the first letter needed to begin with an *uppercase letter*, the dictionaries turned out much to the plan. The clients seems to be able to use the Navigation menu (user menu) very well and can input data with ease. The user can access reports that are specific to their need. The new system looks impressive and is very easy to get around, all the reports are connected to each of the main tabs e.g Youth and various reports on each activity e.g football, drama etc. I am very happy with the appearance and structure of the database design.

This project has taught me the importance of methodology and why you should complete each part in its phase, or you have to go back and redo things, which can be a pain. That following your method keeps you on track and within your timeframe. I have learned from this project the importance of carrying out a full investigation and gaining accurate information as to the client’s needs, that you can’t ask enough questions and collect enough information. For instance, the creation of a *simple role call* would have been good for the volunteers who work with the children, which would have been a very simple table with children’s names and a tick box for confirming which children were in attendance etc.

All in all its been a learning process, one which will help me to improve my skills.

-<http://creately.com/blog/diagrams/database-modeling-basics/>

Hierarchical Model - <http://creately.com/blog/diagrams/database-modeling-basics/>