BELTEL

HOTEL | SPA & BEAUTY COMPLEX

A2 DIGITAL TECHNOLOGY UNIT 2 COURSEWORK

Table of Contents

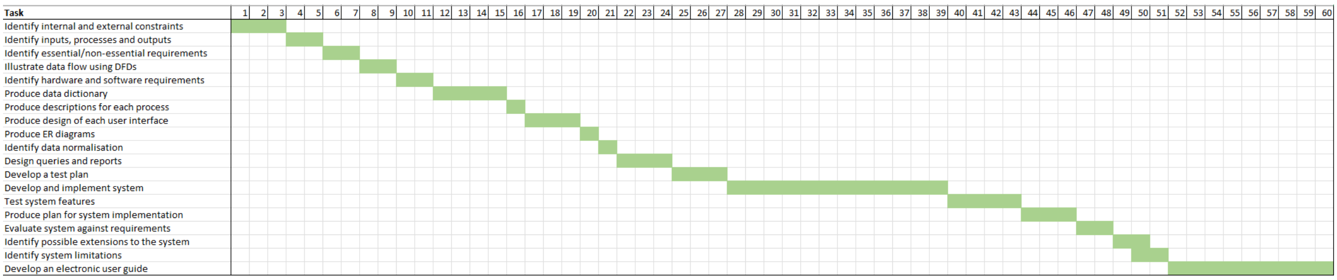
Section 1 – Analysis

Introduction

Beltel is a large hotel, based on the outskirts of Belfast that includes a spa and beauty complex. The complex offers a wide range of packages, including an eating experience with refreshing drinks and healthy snacks. Overtime, their current way of operating the business has grown inefficient; this presents the risk of details being lost or forgotten, resulting in inconvenience for not only management but also the member (who may decide not to stay again). As such, Beltel have approached us to replace the obsolete paper-based system for an ultramodern, digital ordering system that is also capable of storing additional details (such as member, medical, staff etc.) in an organised database.

Gantt chart

Below illustrates how the project will be scheduled (measured in hours):



Internal constraints

Hardware

Beltel's hardware is substandard – the computers used at reception have been discontinued for 15 years. The POS system for the restaurant, an old IBM, suffers from reliability issues and as a result staff have tended to resort to written tickets instead. These machines cannot be retrofitted to work with the new system; it will be dependent on brand new hardware that will allow for efficient operations of Beltel. New adequate hardware including computers, EPOS, keyboards, monitors, and printers will need to be installed to ensure compatibility with the new system throughout the complex. When a waiter inputs an order into the new ordering system, it will be quickly sent to the kitchen and a ticket will be printed. However, receipt printers will need to be installed. Furthermore, new servers will be required for data storage – new data is going to be continuously added and it is important that information such as member details are not lost, otherwise it will cause unnecessary inconvenience for that certain member who might choose to not stay at Beltel again.

Software

The computers throughout Beltel run Windows XP – Microsoft ended support for this operating system in 2014, making it vulnerable to security risks. As for the restaurant POS, it runs OS/2 which was initially released in the 1980s. Microsoft Access will be the backbone of the new system, from storing customer details to the list of dishes served at the restaurant. Without up-to-date hardware to accommodate this, they cannot work in unison. The new machines will come installed with Windows 11 although to use Access new licenses will need to be purchased by Beltel, specifically the Microsoft 365 Enterprise plan. They can also use Microsoft's other products, such as Publisher for designing brochures or Outlook for staff emails, allowing for flexibility. In addition, the new EPOS tablets will have iPadOS installed along with the Microsoft Office app and can be activated using the same license. Overall, the new software will be intuitive for those not technologically skilled with computers or Office although some form of training will be required.

Personnel

According to a survey, it concluded that older members of staff are not computer literate. 20% of Beltel's workforce are over 40. This is a problem – without the necessary training, it is impossible for them to efficiently work with the new system. Although Microsoft Access is designed to be intuitive, it can be overwhelming to someone who has never used it before. Therefore, new training courses will be established so that members can familiarise themselves with the new system.

Budget

It is important that the system's cost is within Beltel's price range, otherwise they will be unwilling to purchase everything that is required if it means taking on debt. In addition to paying for the system, Beltel must purchase new hardware to work with it, ranging from computers to printers. Beltel must also purchase the license for Microsoft Access which gives them access to not only the database but also updates to that application. The training courses means the hotel will pay their staff during their time learning the new system. Overall, this will cost Beltel approx. £15,000 which will put slight strain on their budget.

Time

Beltel has requested that the system be completed within a 60-hour time frame – not a lot for a big project such as this. As a result, there will not be enough time to develop more complex features; search bars will not be included, for example, and will come in a later patch. The amount of time testing the system will not be adequate, therefore preparation is needed in the event of bugs in the system's first release. Development time will be further constrained because employees will need to spend time training themselves on how to use the new system.

External constraints

Legal issues

The system will need to comply with the Data Protection Act – this means that information stored within the system must follow the data protection principles. For example, member details cannot be kept for longer than is necessary. If a customer is no longer a member, Beltel is responsible for deleting the relevant data from the system. Beltel must also be responsible for collecting adequate, relevant data. For example, they cannot collect excessive member details (e.g., orientation) and they must be aware of stronger legal protection regarding sensitive information such as race, religion etc. Security measures (e.g., access via a username and password, and/or a firewall) must be present to protect against unauthorised access of data (e.g., credit card details), an offense under the Computer Misuse Act.

The Copyright, Designs and Patents Act prevents redistribution and modification of this system outside of Beltel – similar systems will be developed from scratch. If it occurs that this system is being used without authorisation, action will be taken, and violators will be prosecuted. Only the owner, or their exclusive licensee (in this case, Beltel) can make proceedings in the courts against an infringement. The Microsoft 365 Enterprise plan must be purchased using a valid license to conform to legal utilisation of the application suite (including Access), otherwise it will be regarded as an infringement by developing the system using pirated software; this may also result in a lack of support for updates or a ban from using Microsoft services, as stated under the Microsoft Services Agreement.

User requirements

Inputs

1. **MEMBER table**

   Data that will be input:
   - MemberID - This uniquely identifies each member in the complex
   - Title - Stores the member's title
   - Forename - Stores the member's first name
   - Surname - Stores the member's last name
   - Gender - Stores the member's gender
   - Date of Birth - Stores the member's DOB to verify that they are over 18
   - MembershipTypeID - Stores the membership tier they are a part of
   - MembershipDateStarted - Stores when their membership starts or has started
   - Address Line 1 - Stores the member's house no./street they live at
   - Address Line 2 - Stores the member's second address
   - Town/City - This stores the member's town/city
   - County – Stores what county the member lives in
   - Postcode - Stores the postcode of each member
   - Telephone No. - Stores the telephone no. of each member
   - Email - Stores the email address of each member

   This data will be captured from a new member using a member application form.

2. **MEDICAL table**

   Data that will be input:
   - MedicalID – This uniquely identifies each medical condition
   - MedicalCondition – Identifies the name of each medical condition

   This data will be captured from a doctor using a medical form.

3. **MEMBERSHIP TYPE table**

   Data that will be input:
   - MembershipTypeID – Uniquely identifies each membership type
   - MembershipType – Identifies the name of each membership type
   - Price Per Month – Stores the cost per month depending on the type of membership
   - Discount (%) – Stores the number of discounts applied depending on the type of membership

   This data will be captured from the owner or a senior manager when, for example, deciding on discounts using a membership details form.

4. **PRODUCT table**

   Data that will be input:
   - ProductID - Uniquely identifies each dish/drink
   - DishName – Identifies the name of each dish/drink

- DishType - Identifies the type of dish served whether it is a starter, main course, or dessert
- Calories - Stores how many calories the dish has
- Price – Stores how much each dish costs
- VeganSuitable – Identifies if the dish is suitable to eat for vegans
- GlutenFree – Identifies if the dish is suitable to eat for members with gluten intolerance
- ContainsMilk – Identifies if the dish is suitable to eat for members who are lactose intolerant

This data will be captured from the suppliers using a product form.

5. **STAFF table**
   Data that will be input:
   - StaffID – Uniquely identifies each staff member working at Beltel
   - Title – Stores the staff member's title
   - Forename – Stores the staff member's forename
   - Surname – Stores the staff member's surname
   - Date of Birth – Stores the staff member's DOB to verify that they are old enough to work at Beltel
   - Position – Describes each member's job
   - PayGradeCode – Defines the amount each staff member is paid hourly
   - TelephoneNo – Stores the telephone no for each staff member
   - Email Address – Stores the email address for each staff member

   This data will be captured from a manager when someone has been employed using a staff application form.

6. **ORDER table**
   Data that will be input:
   - OrderID – Uniquely identifies each order that's been placed
   - MemberID – Uniquely identifies what member placed the order
   - StaffID – Uniquely identifies what staff member took the order
   - OrderDate – Stores when the order was placed
   - Paid? – Identifies whether the member has paid for their order
   - PaymentMethod – Stores how the member paid for their order
   - ProductID – Uniquely identifies the product that's been ordered
   - Quantity – Defines the amount that's been ordered

   This is all the data needed to take an order; it will be captured from a staff member when taking a customer's order using an order form.

Most importantly, these forms must be intuitive for the user to ensure all data is captured. They also must look professional e.g., following the hotel's colour scheme.

Processes

The processes for the system must run very quickly – for example, searching for unpaid orders must aim to take less than five seconds.
This will include the following:

1. **Calculations**

   For the order form, invoice and receipt, there will be three fields requiring calculations to be displayed correctly. This will be the subtotal, the discount applied, and the total money owed for that order.

   a. The subtotal will be calculated using a SUM formula whereby the prices for products a customer has ordered will be totalled. This identifies how much money needs to be paid for the items ordered.
   b. The discount will be calculated by dividing the subtotal by 100 and then multiplying by the discount amount. This identifies how much the member saves on their order depending on the type of membership they have.
   c. The total owed will be calculated by subtracting the discount from the subtotal. This gives us the final price that the member must pay for their order.

   Regarding members with medical conditions, there's only one field requiring a calculation to be displayed correctly. This will be the age – using the DateDiff formula, a difference will be calculated between the member's DOB and current date which is then formatted/displayed in years. This identifies not only the member's age but also ensures that they are over 18.

   For monthly sales, there are four fields requiring calculations to be displayed properly. This will be the order year, month number, month, item total, and the discount applied.

   a. The order year will use the Year() function, taking the year of the order date and displaying it only.
   b. The month number will use the Month() function, taking the month number of the order date and displaying it only.
   c. The month will be calculated using the MonthName() function. It returns the month name based on the number of the order date. This identifies how many orders were placed in a specific month.
   d. The item total will be calculated using a SUM formula whereby the prices for products a customer has ordered will be totalled. This identifies how much money was made from an order.
   e. The discount will be calculated by dividing the item total by 100 and then multiplying by the discount amount. This identifies how much has been lost on an order depending on the type of membership the member has.

   For best-selling products, two fields will require calculations to be displayed properly – the number purchased and the total sales.

   a. The number purchased will be calculated using a SUM formula whereby the quantity for a product every customer has ordered will be totalled. This tells us how many times a product has been ordered.

   b. The total sales will also be calculated using a SUM formula whereby the price of a product is multiplied by the number purchased. This gives us the total profit made from a product.

2. **Modifications/updating of data**
   Adding/Deleting data will require this:
   a. Certain records will be updated whenever details are changed. This ensures that the details of all data are up-to-date and that the chance of errors are little to none.
      i. An example would be updating a member's email address.
   b. Data can be changed whenever necessary; the system will not be read-only otherwise problems will arise if the data needs amended
      i. An example of this would be removing an item that has accidentally been ordered twice.
   c. A special case would be the order archive process; it requires both the modifications and updating of data. By running the associated macro, the system removes orders placed more than six months ago and adds them to their own dedicated table.
      i. This removes unnecessary storage space and ensures the system remains efficient.

3. **Searching of data (Criteria)**
   Data will be searched to find all the details. In general, this makes it easier to find the necessary specifics. Some examples include:
   a. Members - This makes it easier to find details of a certain member, like members who have a specific membership or medical condition.
   b. Orders – Useful for finding members who've either paid for their recent order, or who have yet to pay.
      i. If the customer has paid, the data will be used to produce a receipt which is sent to the customer and stored to show proof of purchase.
      ii. If they haven't paid, the data will be used to produce an invoice – this is sent to the customer and stored to tell them that they owe Beltel money.
   c. Finding the list of products served at the Beltel restaurant. This is ideal for waiters who want to check the details of a certain dish. The data can also be filtered so that, for example, only gluten free products are shown.
   d. Finding the monthly sales generated at the Beltel restaurant; Useful for management for defining what months produced the most profit and when it produced the least.
      i. Discounts can also be applied to each month; this defines how much profit was lost because of Beltel's membership scheme.
   e. Searching for the best-selling products at the Beltel restaurant; ideal for management for evaluating what products generated the most profit and what generated the least. The data can also be filtered so that free products are omitted.
   f. Finding the most used payment methods - for Beltel, this is useful when evaluating whether support for less popular methods should be ended to save costs; they can invest more time and money on other methods. For example, installing brand new card readers with support for Apple/Google Pay. The data can also be filtered so that unpaid orders are omitted from the query.

4. **Sorting data**

Data can be sorted in a variety of ways. This makes it easier to view the necessary data and useful for producing reports or other analytics of the hotel. Examples include:

   a. Products that can be sorted in descending order in terms of calories. This is ideal for members who may be limited in their food intake.
   b. Sorting staff members with who gets the most pay. This is useful when deciding on raises/reduces.
   c. Sorting monthly restaurant sales with what month produced the most profit. This is ideal for identifying why that is the case and deciding how to increase profits for the other months.
   d. Sorting products by what produced the most profit. This is ideal for identifying why that is the case and deciding how to increase sales for the other products.
   e. Sorting the most popular payment methods by descending order. This is useful when deciding whether to abandon a less popular method.
   f. Unpaid orders that can be sorted in ascending order in terms of when the order was placed. This is ideal for contacting members with the most outstanding invoices.

Outputs

This process involves the generation of reports that the system will be responsible for. It must aim to produce reports in less than five seconds and be displayed in a layout that is easy to understand. Examples of this include:

1. **Invoice report**

   This is printed and sent to customers who have yet to pay for an order. It lists customer details, the products they have ordered (including the price, quantity, and item total), and the total money owed.

   For Beltel, this is useful for recouping profits that would've otherwise been lost indefinitely. It also ensures that the customer knows about the unpaid order.

2. **Receipt report**

   This is printed and sent to customers who've recently paid for an order. It lists customer details, the products they have ordered, and how much they paid depending on their membership tier.

   This is useful for Beltel management to keep track of, for example, the most popular dish sold at the restaurant; it evaluates whether any changes should be made to the menu. Another example includes lost profits because of membership discounts, whereby management assesses possible changes to the membership scheme.

3. **Members with medical conditions report**

   This is printed and sent to management to notify them of members with medical conditions. It lists member details, the conditions they suffer from, and their contact details in case of an emergency.

   For Beltel, this ensures that they are given special consideration by staff during their stay; this also improves Beltel's reputation. For example, a member with a milk allergy will not be given products containing milk.

4. **Monthly sales report**
   This is printed and sent to management at the end of the year. It lists the profits from each month of the year, then the revised profits after applying discounts, and the grand total for the year.
   For Beltel, this is useful for identifying when they generated the most and least profit; it enables management to evaluate why.

5. **Most popular payment methods report**
   This is printed and sent to management to keep track of the payment methods used by their members. It details how frequently each method is used when placing an order.
   For Beltel, this is useful when evaluating whether support for less popular methods should be ended to save costs; they can invest more time and money on other methods. For example, installing brand new card readers with support for Apple/Google Pay.

6. **Best-selling products report**
   This is printed and sent to management listing the price of each product, how frequently it has been ordered, and the total sales made from a product.
   For Beltel this is useful when, for example, identifying the lower selling products; it evaluates whether any changes should be made to the menu.

Essential requirements

Listed below are requirements that the system must perform in its first release; without them, the system is redundant. For example:

- **Adding/modifying member details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot print receipts/invoices without knowing who ordered.
- **Adding/modifying product details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot place orders when you don’t have products to order from.
- **Adding/modifying order details** – The backbone of the ordering system. The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot keep track of, for example, unpaid orders without a table to store them.
    - **In addition, a query to view/store main order details must be created** – it provides the control source necessary for the MAIN ORDER form, otherwise it is redundant. As such, a list of orders must be returned no later than 10 seconds after the query is run.
    - **There should also be a query for the order subform** – it provides the control source necessary for the ORDER SUBFORM and is also vital for some fields in the MAIN ORDER form, such as the subtotal. A list of products purchased for each order must be returned no later than 10 seconds after the query is run.
- **Adding/modifying medical details** – Vital for knowing who needs special care from staff. The form will be an intuitive way to input the data and must be processed in less than 5 seconds. You don’t want a member with a peanut allergy allowed to order a product containing peanuts.
- **Calculating totals** – Necessary for orders/sales. Without a subtotal, it’s hard to tell how much Beltel has made from an order. The calculations must be accurate, otherwise problems will arise if false profits/losses are recorded.

- **Calculating miscellaneous fields** – Also necessary for orders/sales. Without an order year, for example, it will be very difficult to calculate the monthly sales for each year. The calculations must be accurate, otherwise problems will arise if the wrong fields are recorded.
- **Adding/modifying membership details** – Necessary when applying discounts to orders, otherwise members will be inconvenienced. The form will be an intuitive method to input the data and must be processed in less than 5 seconds.
- **Adding/modifying staff details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot place orders when you don't know who took the order.
- **Producing a receipt/invoice** – Vital for knowing what a member has ordered, and how much they paid/owe. Management can easily contact members with an invoice. The report must be produced in less than 10 seconds and presented in a professional manner.
- **Creating queries (searching) for invoices/receipts** – they provide the control source for their respective reports, otherwise they are inessential. As such, a list of paid/unpaid orders must be returned no later than 10 seconds after the query is run.
- **Creating a query (searching) for members with medical conditions** – it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of members suffering with medical conditions must be returned no later than 10 seconds after the query is run.
- **Creating a query for monthly sales** – it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of profits from each month of the year (with the discount applied) must be returned no later than 10 seconds after the query is run.
  - **In addition, a query must be added that applies the discounts for all orders which then must be grouped and linked to the query above.** A list of orders (with the discount applied) must be returned no later than 10 seconds after the query is run. They rely on each other, and hence vital for management when evaluating losses due to discounts.
- **Creating a query (searching) for the best-selling products** - it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of the highest selling products at the restaurant must be returned no later than 10 seconds after the query is run.
- **Creating a query (searching) for the most popular payment method** - it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of payment methods (starting by the most frequent) must be returned no later than 10 seconds after the query is run.

Non-essential requirements

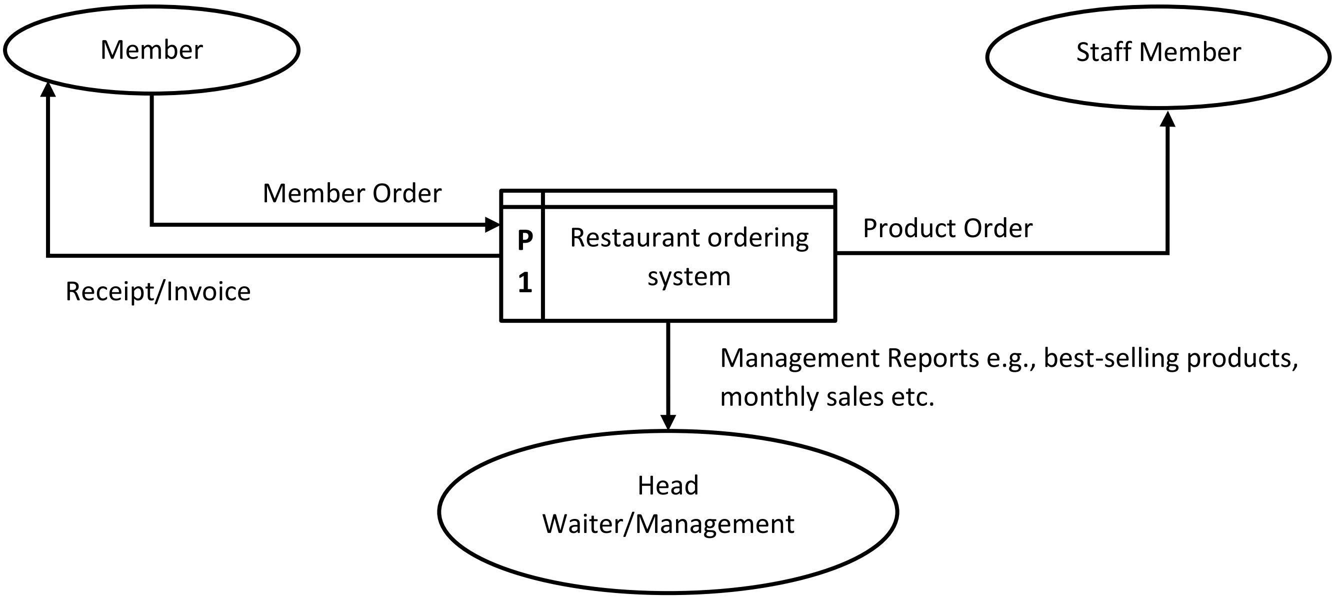
Listed below are extra requirements that the system will perform in later releases:

- **Monthly sales report system** – this assists management when evaluating the restaurant's performance. It must be produced in less than 10 seconds and presented in a professional manner. All it does is give insight into Beltel's performance; the system can function without it.
- **Best-selling products report system** – this assists management when evaluating the restaurant's menu. It must omit free products (since they will not make a profit), be produced in less than 10 seconds and should be presented in a professional manner. However, all it does is give insight into Beltel's performance; the system can function without it.

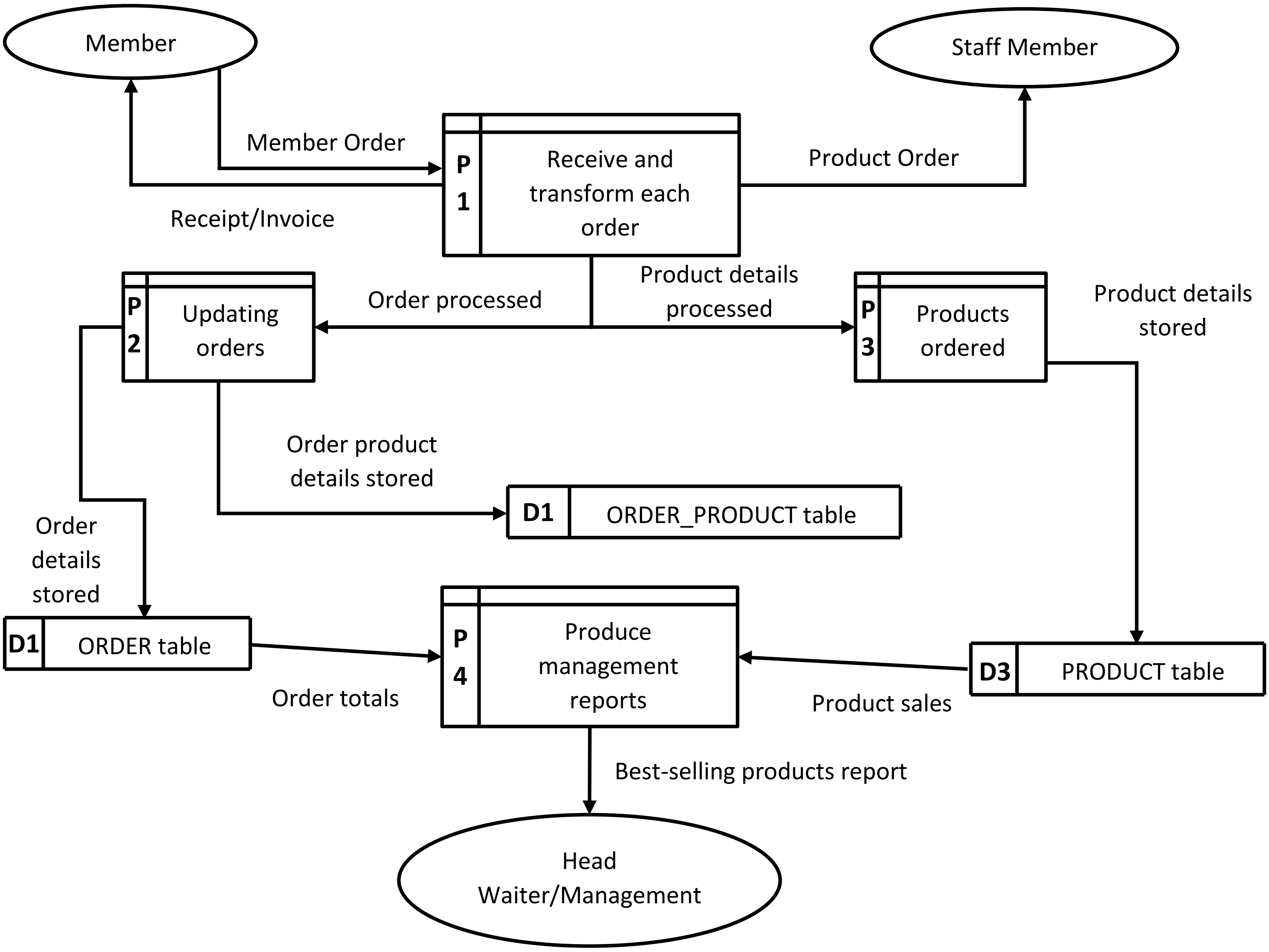
- **Members with medical conditions report** – this notifies management of members suffering from medical conditions; however, this can be discovered by other means, not just through a report. Nevertheless, once it is developed, it must be produced in less than 10 seconds and presented in a professional manner.
- **Developing a switchboard user interface** – this provides a more user-friendly experience when interacting with the database. Navigation should be quick; load times should take less than 4 seconds along with an intuitive layout. Nevertheless, interaction is possible without the switchboard at the expense of intuitiveness.
- **Archiving orders** – orders older than six months, for example, will be archived to increase efficiency and decrease storage space. The macro required for this operation must process the data in less than 5 seconds, with message boxes to notify the user of its status. Development can begin after the first release since there'll be no orders to archive, although it must be completed before the deadline.
    - **Two queries are required for this macro** – one must append the data required to a new table, while the other must delete the specific data from their old tables. Both must be performed no more than 5 seconds.
- **Preventing members with allergies from ordering flagged products** – for example, a staff member should receive an error message no later than 3 seconds after a member with a milk allergy attempts to order a product containing milk. However, this can be achieved by other means, such as highlighting allergens on the menu.
- **Show a splash screen on start-up** – this should be presented professionally to the user as the system starts up. It must show the Beltel logo and follow the brand's colour scheme. All it does is notify the user that the system is loading; it can function without it.

Data Flow Diagrams (Ordering system)

Level 0


Member
Staff Member
Member Order
P 1
Restaurant ordering system
Product Order
Receipt/Invoice
Management Reports e.g., best-selling products, monthly sales etc.
Head Waiter/Management

Level 1



Hardware requirements

Specific hardware will be required for performing specialised functions; these include:

- Desktop computers capable of supporting and running the latest versions of Microsoft Access/365. Recommended would be the HP ProOne 440 G9 Full-HD All-in-One.
- Printers capable of colour printing, copying, scanning and faxing. Mostly suited for reports. Additionally with Wi-fi functionality to enable wireless printing. A recommended printer would be the HP Colour Laser 179fnw Wireless Multifunction printer with Fax.
- The EPOS system will require Apple iPads designed for placing orders. An example would be the cloud based Lightspeed POS system. In addition, a card reader that supports contactless to process orders and authorise transactions. This also includes a printer that supports Apple Airprint to enable the iPads to print receipts wirelessly.
- External webcams for holding video calls and meetings. For example, the Logitech C920s.

Software requirements

Since Access is part of Microsoft 365 suite of apps, the following software requirements to run the system are detailed below:

- **Computer and processor**: 1.6 GHz or faster, 2-core. 2 GHz or greater recommended for Skype for Business.
- **Memory**: 4 GB RAM (64-bit) or 2 GB RAM (32-bit)
- **Hard disk:** 4GB of available disk space.
- **Resolution:** 1280 x 768 screen resolution (32-bit requires hardware acceleration for 4K and higher). Web apps require the same minimum resolution as the OS they are running on. In addition, apps running inside of Microsoft Teams adhere to the application’s minimum resolution.
- **Graphics:** DirectX 9 or later required for graphics hardware acceleration. Skype for Business requires DirectX 9 or later, 128 MB graphics memory, and 32-bits-per-pixel-capable format.
- **Operating System**: Windows 11, Windows 10, Windows 8.1, Windows Server 2019, or Windows Server 2016.
- **Browser (for web apps):** The latest versions of Microsoft Edge, Chrome, or Firefox.
- **Other:** Microsoft Edge WebView2 must be installed to use additional Outlook features.
- **Video calls and meetings:** A 2 GHz processor with 4GB RAM or higher is recommended. The optional blurred background effect requires a processor with Advanced Vector Extensions 2 (AVX2) support.
- **Teams live events:** For holding live events, an Intel Core i5 Kaby Lake processor with 4GB RAM and a hardware decoder is recommended.

Section 2 - Design

Data Dictionary

Database File	Beltel Database.mdb	***Table Name***	MEMBER.tbl	***(Composite) Key Field***	MemberID

General table description: To store and provide an outline of all current members of the Beltel complex

Field Name	***R***	***Data Type***	***Length***	***Validation Rule***	***Default Value***	***Description***	***Example of Typical Data***
MemberID	Y	AutoNumber				This uniquely identifies each member in the complex	6
Title	N	Short Text	4			Stores the member's title (if they have one)	Miss
Forename	Y	Short Text	20			Stores the member's first name	Sophie
Surname	Y	Short Text	25			Stores the member's second name	Hunt
Gender	Y	Short Text	20	Lookup: Male, Female, Non-binary, prefer not to say		Stores the member's gender	Female
DOB	Y	Date/Time		Format: DD/MM/YYYY <=DateAdd("yyyy",-18,Date())		Stores the member's DOB to verify that they are over 18	27/01/1997
MembershipTypeID	Y	Number		Lookup from MEMBERSHIP TYPE table: MembershipTypeID, MembershipType		Stores the membership tier they are a part of	3
MembershipDateStarted	Y	Date/Time		Format: DD/MM/YYYY		Stores when their membership starts or has started	09/09/2022
Address Line 1	Y	Short Text	20			Stores the member's house no./street they live at	75 Hyde Road
Address Line 2	N	Short Text	45			Stores the member's second address	Hyde
Town/City	Y	Short Text	45			This stores the member's town/city	Paignton

County	Y	Short Text	30			This stores what county the member lives in	Devon
Postcode	Y	Short Text	7	Format: L?09 0LL		Stores the postcode of each member	TQ4 5BP
TelephoneNo	Y	Short Text	12	Format: 00000 000000		Stores the telephone no of each member	08455 441231
Email	Y	Short Text	40			Stores the email address of each member	sophhunter96@yahoo.com

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	MEDICAL.tbl	***(Composite) Key Field***	MedicalID

General table description: To store a list of medical conditions that some members of Beltel may suffer from

Field Name	***R***	***Data Type***	***Length***	***Validation Rule***	***Default Value***	***Description***	***Example of Typical Data***
MedicalID	Y	AutoNumber				This uniquely identifies each medical condition	2
MedicalCondition	Y	Short Text	50			Identifies the name of each medical condition	Asthma

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	MEMBER_MEDICAL.tbl	***(Composite) Key Field***	MemberID, MedicalID

General table description: To store what member suffers from what type of medical condition or conditions

Field Name	***R***	***Data Type***	***Length***	***Validation Rule***	***Default Value***	***Description***	***Example of Typical Data***
MemberID	Y	Number		Lookup from MEMBER table: MemberID, Forename, Surname		Uniquely identifies each member who suffers from a medical condition	7
MedicalID	Y	Number		Lookup from MEDICAL table: MedicalID, MedicalCondition		Uniquely identifies each medical condition that the member suffers from	14

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	MEMBERSHIP TYPE.tbl	***(Composite) Key Field***	MembershipTypeID

General table description: To store details of each membership type provided by Beltel

Field Name	*R*	*Data Type*	*Length*	*Validation Rule*	*Default Value*	*Description*	*Example of Typical Data*
MembershipTypeID	Y	AutoNumber				Uniquely identifies each membership type	2
MembershipType	Y	Short Text	10			Identifies the name of each membership type	Bronze
Price Per Month	Y	Currency		<=100	0	Stores the cost per month depending on the type of membership	£9.99
Discount (%)	Y	Number		>=10 And <=50	0	Stores the number of discounts applied depending on the type of membership	15%

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	PRODUCT.tbl	***(Composite) Key Field***	ProductID

General table description: This stores details of the dishes and drinks served at Beltel

Field Name	*R*	*Data Type*	*Length*	*Validation Rule*	*Default Value*	*Description*	*Example of Typical Data*
ProductID	Y	AutoNumbe r				Uniquely identifies each dish/drink	1
DishName	Y	Short Text	50			Stores the name of each dish/drink	Mushroom and chilli burger
DishType	Y	Short Text	10	Lookup: Starter, Main, Dessert, Side, Drink		Stores the type of dish served	Main
Calories	Y	Number		< = 10000		Stores how many calories the dish has	813
Price	y	Currency			0	Stores how much each dish costs	£12.00
VeganSuitable	Y	Yes/No			No	Identifies if the dish is suitable to eat for vegans	No
GlutenFree	Y	Yes/No			No	Identifies if the dish is suitable to eat for members with gluten intolerance	No
ContainsMilk	Y	Yes/No			No	Identifies if the dish is suitable to eat for members who are lactose intolerant	Yes

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	STAFF.tbl	***(Composite) Key Field***	StaffID

General table description: To store and provide an outline of all current employees of the Beltel complex

Field Name	*R*	*Data Type*	*Length*	*Validation Rule*	*Default Value*	*Description*	*Example of Typical Data*
StaffID	Y	AutoNumber				Uniquely identifies each staff member working at Beltel	1
Title	N	Short Text	4			Stores the staff member's title	Mr
Forename	Y	Short Text	20			Stores the staff member's forename	Max
Surname	Y	Short Text	25			Stores the staff member's surname	Knight
DOB	Y	Date/Time		Format: DD/MM/YYYY <=DateAdd("yyyy",-18,Date())		Stores the staff member's DOB to verify that they are old enough to work at Beltel	31/10/1995
Position	Y	Short Text	30			Describes each member's job	Sous Chef
PayGradeCode	Y	Short Text	1	Lookup: S - £17.90, A - £15.50, B - £13.25, C - £11.50, D - £10.90		Defines the amount each staff member is paid hourly	A
TelephoneNo	Y	Short Text	12	Format: 00000 000000		Stores the telephone no for each staff member	02893 445076
Email Address	Y	Short Text	40			Stores the email address for each staff member	Max.KNIGHT95@outlook.com

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	ORDER.tbl	***(Composite) Key Field***	OrderID

General table description: This stores details of the orders placed at the Beltel restaurant

Field Name	*R*	*Data Type*	*Length*	*Validation Rule*	*Default Value*	*Description*	*Example of Typical Data*
OrderID	Y	AutoNumber				Uniquely identifies each order that's been placed	1
MemberID	Y	Number		Lookup from MEMBER table: MemberID, Title, Forename, Surname		Uniquely identifies what member placed the order	6
StaffID	Y	Number		Lookup from STAFF table: StaffID, Title, Forename, Surname		Uniquely identifies what staff member took the order	2
OrderDate	Y	Date/Time		Format: DD/MM/YYYY		Stores when the order was placed	27/10/2022
Paid?	Y	Yes/No			No	Identifies whether the member has paid for their order	Yes
PaymentMethod	N	Short Text		Lookup: Card, Cash, Cheque, PayPal, Apple Pay, Google Pay	N/A	Stores how the member paid for their order	Cash

Key: R = Required, Y = Yes, N = No

Database File	Beltel Database.mdb	***Table Name***	ORDER_PRODUCT.tbl	***(Composite) Key Field***	OrderID, ProductID

General table description: To store details of what product/dish has been ordered from the Beltel restaurant

Field Name	***R***	***Data Type***	***Length***	***Validation Rule***	***Default Value***	***Description***	***Example of Typical Data***
OrderID	Y	Number		Lookup from ORDER table: OrderID		Uniquely identifies each order that's been placed	1
ProductID	Y	Number		Lookup from PRODUCT table: ProductID, DishName		Uniquely identifies the product that's been ordered	4
Quantity	Y	Number		>=1 And <=100	1	Defines the amount that's been ordered	2

Key: R = Required, Y = Yes, N = No

Macros

1. **ORDER ARCHIVE MACRO**
   1. Message box appears to ask for confirmation before the archive process begins.
   2. ORDER ARCHIVE QUERY is run – this appends old orders and adds them to a specially created table.
   3. DELETE ORDER query is run – this deletes old orders from the order table.
   4. Message box appears to notify that the process has been completed.

SQL Queries

1. **Best-selling products**
   SELECT DishName, Price, Number Purchased: Quantity
   FROM Product, Order_Product
   WHERE Total Sales >0
   SUM Number Purchased: Quantity
   Total Sales: [Price]*[Quantity]
   ORDER BY Total Sales DESC

2. **Delete order**
   DELETE FROM Order
   WHERE OrderDate<Date()-182

3. **Invoice**
   SELECT MemberID, Forename, Surname, MembershipTypeID, Discount (%), OrderID, OrderDate, Paid, PaymentMethod, ProductID, DishName, Price, Quantity
   FROM Membership Type, Member, Order, Order_Product, Product
   WHERE Paid = “No”
   Item Total: [Quantity]*[Price]

4. **Main Order**
   SELECT OrderID, MemberID, StaffID, OrderDate, Paid, PaymentMethod, Forename, Surname, TelephoneNo, Email, MembershipTypeID, Discount (%), Forename, Surname, TelephoneNo, Email Address
   FROM Member, Membership Type, Order, Staff

5. **Members with medical conditions**
   SELECT MemberID, Forename, Surname, Gender, DOB, MedicalID, MedicalCondition, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email
   FROM Member, Member_Medical, Medical
   GROUP BY MemberID
   ORDER BY Forename ASC
   Age: DateDiff("yyyy",[DOB],Date())

6. **Monthly Sales**
   SELECT Item Total
   FROM Order, Order Subform
   WHERE Paid = “Yes” AND Order Year = [Please specify order year]
   GROUP BY Order Year
   ORDER BY OrderDate ASC
   Order Year: Year([OrderDate])

Month: MonthName(Month([OrderDate]))

7. **Monthly Sales Discount**
   SELECT OrderID, Item Total, Discount (%)
   FROM Order, Order Subform, Membership Type
   GROUP BY OrderID
   Discount Applied: Sum([Item Total]/100*[Discount (%)])

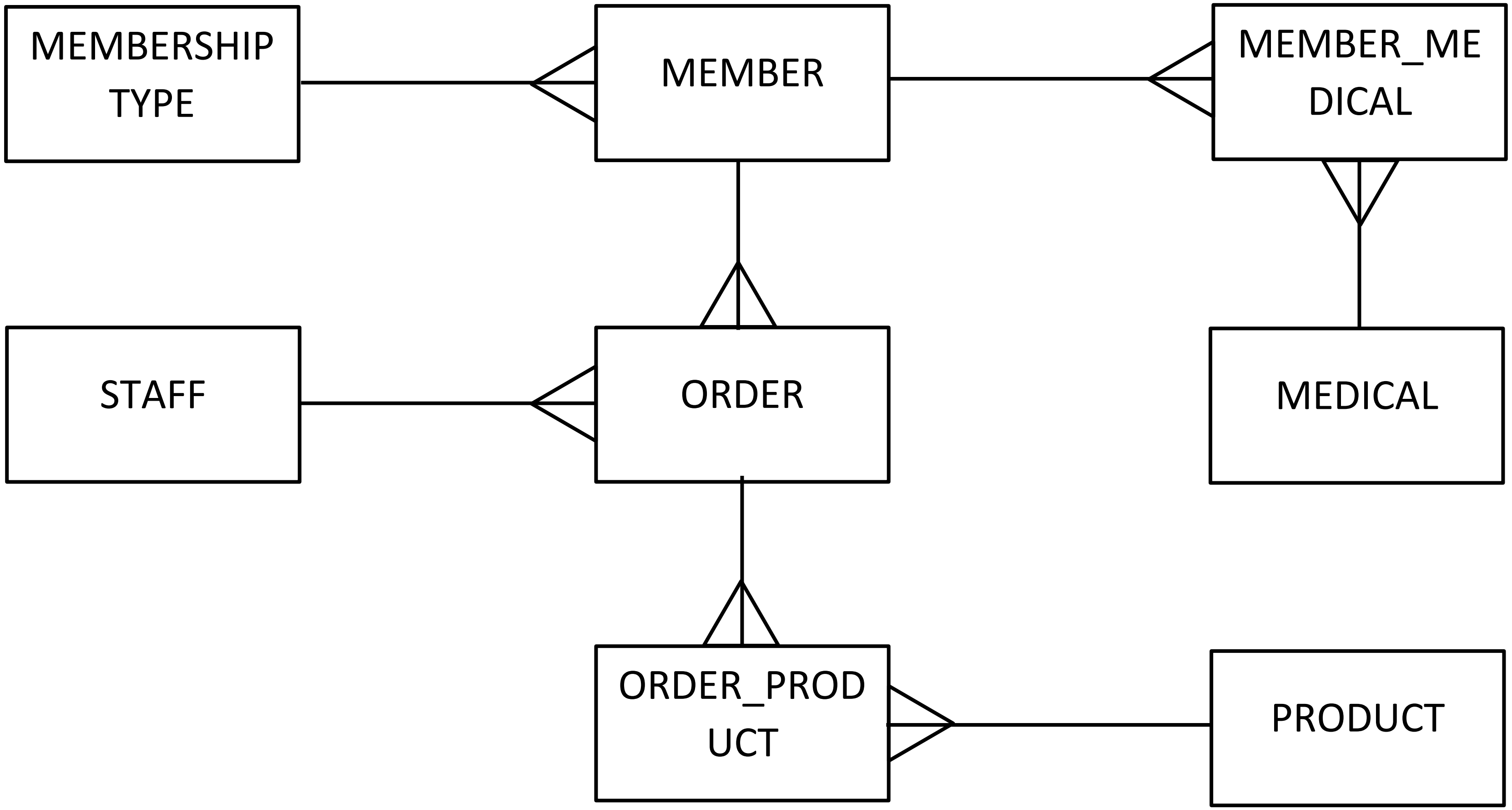
8. **Most popular payment method**
   SELECT PaymentMethod
   FROM Order
   GROUP BY PaymentMethod
   COUNT PaymentMethod
   ORDER BY CountOfPay DESC

9. **Order archive**
   INSERT INTO Order Archive
   VALUES (Forename, Surname, TelephoneNo, OrderID, OrderDate, DishName, Price, Quantity)
   WHERE OrderDate<Date()-182

10. **Order Subform**
    SELECT OrderID, ProductID, Quantity, DishName, Price, DishType, Calories, VeganSuitable, GlutenFree, ContainsMilk
    FROM Order_Product, Product
    Item Total: [Quantity]*[Price]

11. **Receipt**
    SELECT MemberID, Forename, Surname, MembershipTypeID, Discount (%), OrderID, OrderDate, Paid, PaymentMethod, ProductID, DishName, Price, Quantity
    FROM Membership Type, Member, Order, Order_Product, Product
    WHERE Paid = “Yes”
    Item Total: [Quantity]*[Price]

ER Diagram


MEMBERSHIP TYPE
MEMBER
MEMBER_MEDICAL
STAFF
ORDER
MEDICAL
ORDER_PRODUCT
PRODUCT

Normalisation

1. Ordering System

0NF

ORDER (OrderID, OrderDate, Paid, PaymentMethod, MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, , MembershipType, Price Per Month, Discount (%), MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email, StaffID, Title, Forename, Surname, DOB, Position, PayGradeCode, TelephoneNo, Email Address, ProductID, DishName, DishType, Calories, Price, VeganSuitable, GlutenFree, ContainsMilk, Quantity)

- Here are all the fields required for the order system.

1NF

ORDER (OrderID, OrderDate, Paid, PaymentMethod, MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipType, Price Per Month, Discount (%), MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email, StaffID, Title, Forename, Surname, DOB, Position, PayGradeCode, TelephoneNo, Email Address)

ORDER_PRODUCT (OrderID, ProductID, DishName, DishType, Calories, Price, VeganSuitable, GlutenFree, ContainsMilk, Quantity)

- Repeating attributes have been identified (in this case, product info.) and removed from ORDER. This forms a new entity called ORDER_PRODUCT.
- OrderID has become the primary key – it is repeated in both tables to link them together.
- ProductID is designated the composite key.

2NF

ORDER (OrderID, OrderDate, Paid, PaymentMethod, MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipType, Price Per Month, Discount (%), MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email, StaffID, Title, Forename, Surname, DOB, Position, PayGradeCode, TelephoneNo, Email Address)

ORDER_PRODUCT (OrderID, ProductID, Quantity)

PRODUCT (ProductID, DishName, DishType, Calories, Price, VeganSuitable, GlutenFree, ContainsMilk)

- Partial key dependences have been identified (in this case, product info.) and removed from ORDER_PRODUCT to a new entity called PRODUCT.
- However, this excludes quantity because it's functionally dependent on both keys.
- ProductID is repeated in ORDER_PRODUCT and PRODUCT to create a relationship between the tables.

3NF

ORDER (OrderID, MemberID, StaffID, OrderDate, Paid, PaymentMethod)

ORDER_PRODUCT (OrderID, ProductID, Quantity)

PRODUCT (ProductID, DishName, DishType, Calories, Price, VeganSuitable, GlutenFree, ContainsMilk)

MEMBER (MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email)

MEMBERSHIP TYPE (MembershipTypeID, MembershipType, Price Per Month, Discount (%))

STAFF (StaffID, Title, Forename, Surname, DOB, Position, PayGradeCode, TelephoneNo, Email Address)

- Non-key dependencies have been identified (in this case, member, membership, and staff info.) and removed to form three new entities – MEMBER, MEMBERSHIP TYPE and STAFF.
- To link relationships, MemberID and StaffID remain in the ORDER table as a foreign key. Likewise, MembershipTypeID remains in MEMBER as a foreign key.

2. Member Details

0NF

MEMBER (MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email, MedicalID, MedicalCondition)

- Here are all the fields required for the member details.

1NF

MEMBER (MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email)

MEMBER_MEDICAL (MemberID, MedicalID, MedicalCondition)

- Repeating attributes have been identified (in this case, medical info.) and removed from MEMBER. This forms a new entity called MEMBER_MEDICAL.
- MemberID has become the primary key – it is repeated in both tables to link them together.
- MedicalID is designated the composite key.

2NF

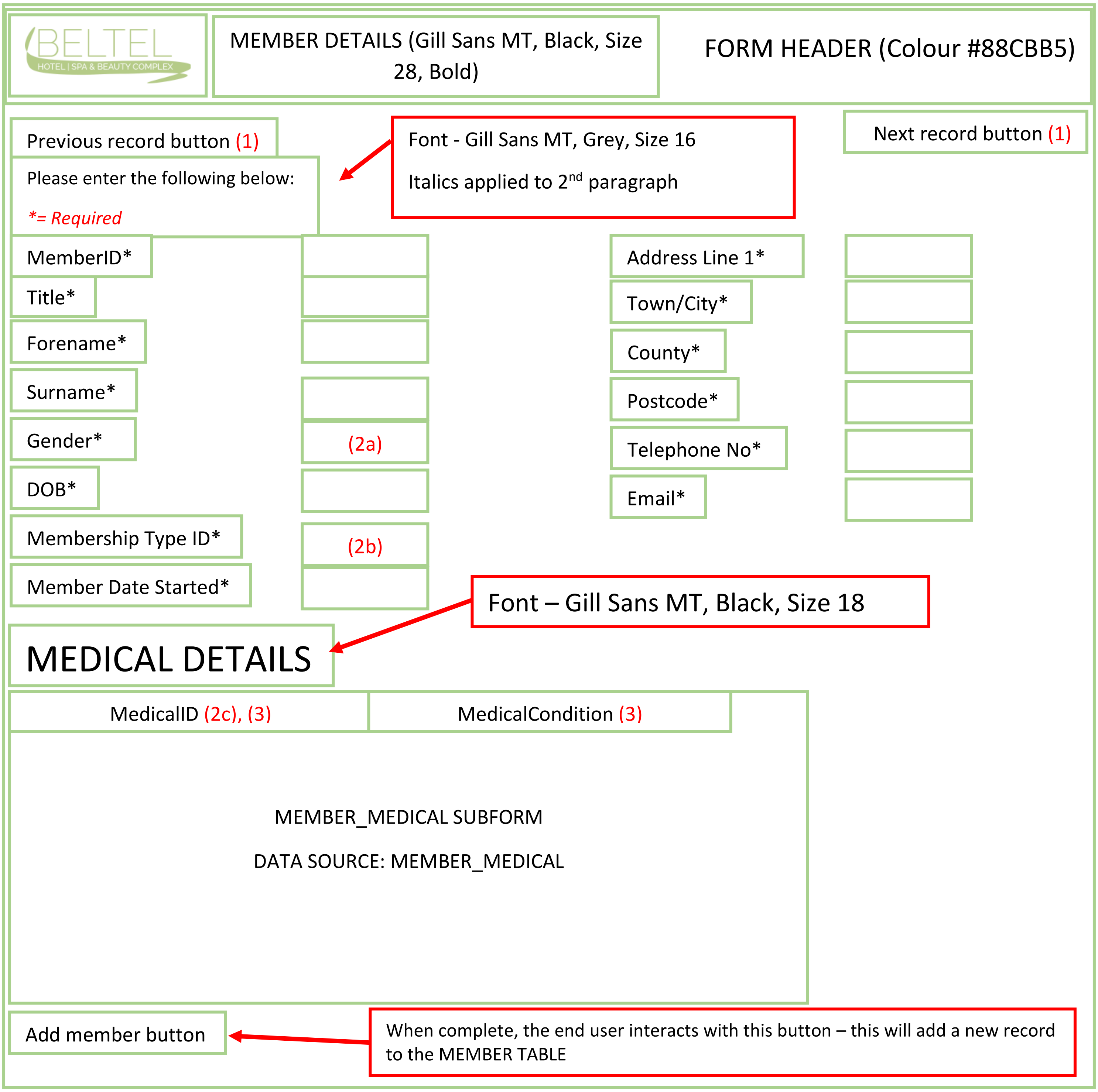
MEMBER (MemberID, Title, Forename, Surname, Gender, DOB, MembershipTypeID, MembershipDateStarted, Address Line 1, Town/City, County, Postcode, TelephoneNo, Email)

MEMBER_MEDICAL (MemberID, MedicalID)

MEDICAL (MedicalID, MedicalCondition)

- Partial key dependences have been identified (in this case, medical info.) and removed from MEMBER_MEDICAL to a new entity called MEDICAL.
- MedicalID is repeated in MEMBER_MEDICAL and MEDICAL to create a relationship between the tables.
- It has now been fully normalised. Therefore, the database is already in 3NF.

MEMBER form interface



BELTEL HOTEL | SPA & BEAUTY COMPLEX

MEMBER DETAILS (Gill Sans MT, Black, Size 28, Bold)

FORM HEADER (Colour #88CBB5)

Previous record button (1)

Font - Gill Sans MT, Grey, Size 16

Italics applied to 2nd paragraph

Next record button (1)

Please enter the following below:

**= Required*

MemberID*

Title*

Forename*

Surname*

Gender*

(2a)

DOB*

Membership Type ID*

(2b)

Member Date Started*

Address Line 1*

Town/City*

County*

Postcode*

Telephone No*

Email*

Font – Gill Sans MT, Black, Size 18

MEDICAL DETAILS

MedicalID (2c), (3)

MedicalCondition (3)

MEMBER_MEDICAL SUBFORM

DATA SOURCE: MEMBER_MEDICAL

Add member button

When complete, the end user interacts with this button – this will add a new record to the MEMBER TABLE

Additional Information

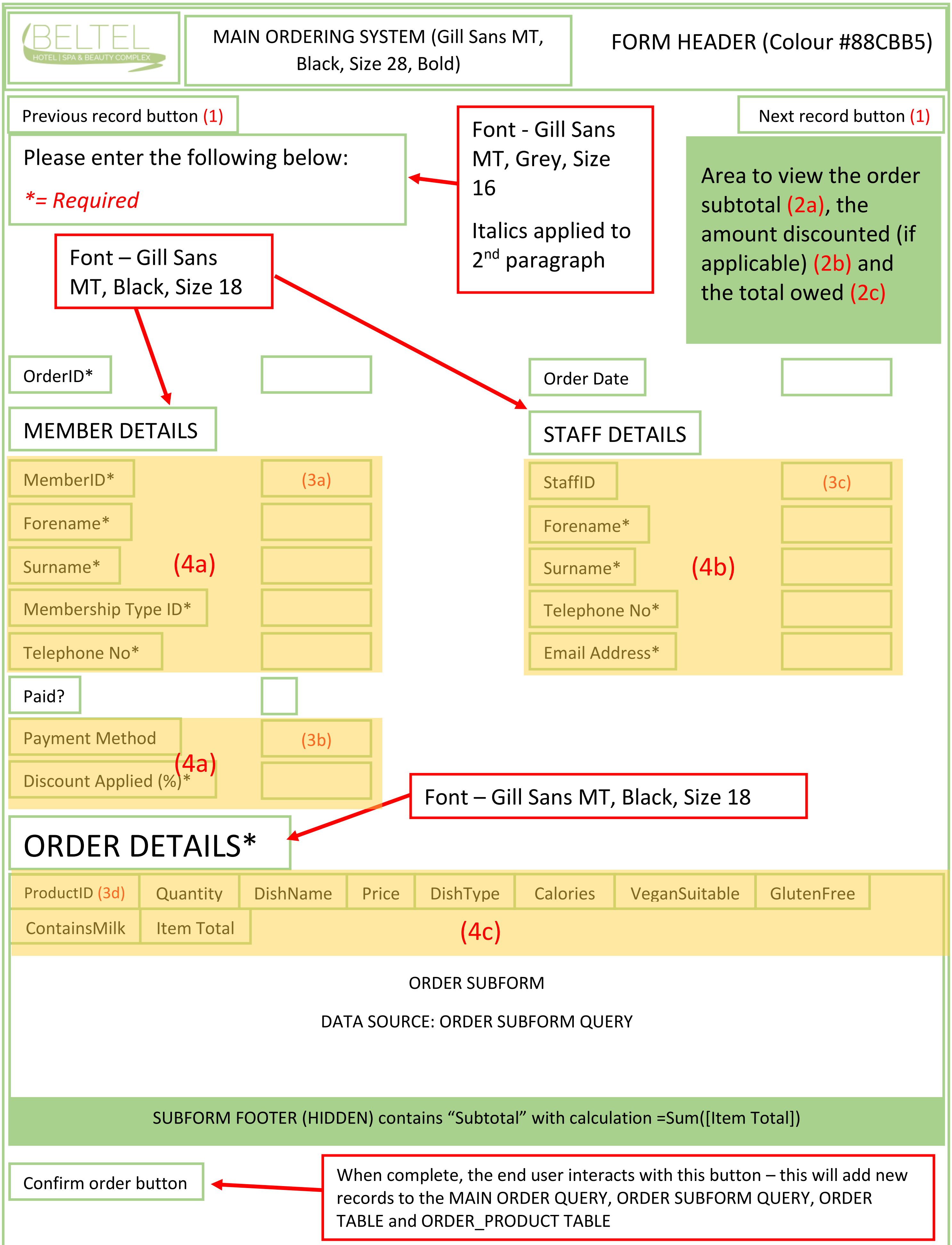
Record source – MEMBER TABLE
Text size – 12pt
Text colour – Grey (#7F7F7F)
Font – Gill Sans MT
Background colour – Light Grey (#D8D8D8)

1. Navigation buttons to view details of different members at Beltel.
2. Drop down menus with data sourced from the following:
   a. **Gender** - MEMBER TABLE (Male, Female, Non-binary, prefer not to say)
   b. **Membership Type ID** – MEMBERSHIP TYPE TABLE (MembershipTypeID, MembershipType)

    c. **MedicalID** – MEDICAL TABLE (MedicalID, MedicalCondition)

3. Autofill – When MedicalID is selected, data for the MedicalCondition field will be automatically filled in from the MEDICAL table.

MAIN ORDER form interface



BELTEL HOTEL | SPA & BEAUTY COMPLEX

MAIN ORDERING SYSTEM (Gill Sans MT, Black, Size 28, Bold)

FORM HEADER (Colour #88CBB5)

Previous record button (1)

Next record button (1)

Please enter the following below:

= Required

Font - Gill Sans MT, Grey, Size 16

Italics applied to 2nd paragraph

Area to view the order subtotal (2a), the amount discounted (if applicable) (2b) and the total owed (2c)

Font – Gill Sans MT, Black, Size 18

OrderID*

Order Date

MEMBER DETAILS

MemberID* (3a)

Forename*

Surname* (4a)

Membership Type ID*

Telephone No*

STAFF DETAILS

StaffID (3c)

Forename*

Surname* (4b)

Telephone No*

Email Address*

Paid? □

Payment Method (3b)

Discount Applied (%)* (4a)

Font – Gill Sans MT, Black, Size 18

ORDER DETAILS*

ProductID (3d) | Quantity | DishName | Price | DishType | Calories | VeganSuitable | GlutenFree

ContainsMilk | Item Total

(4c)

ORDER SUBFORM

DATA SOURCE: ORDER SUBFORM QUERY

SUBFORM FOOTER (HIDDEN) contains “Subtotal” with calculation =Sum([Item Total])

Confirm order button

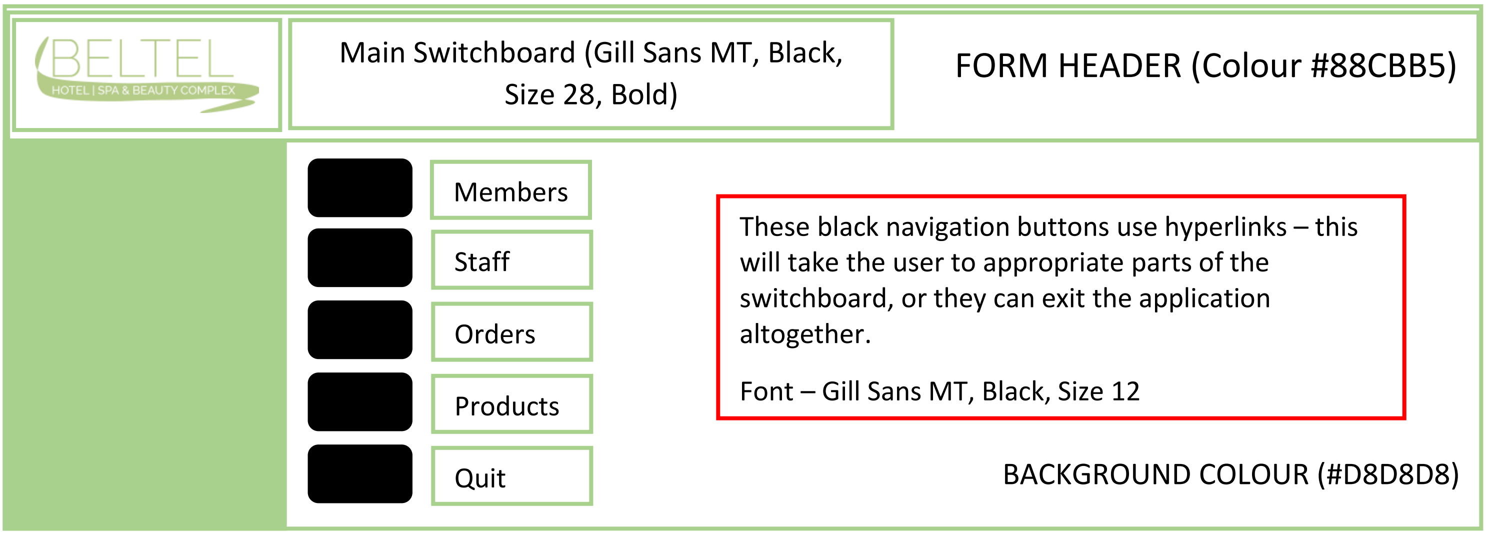
When complete, the end user interacts with this button – this will add new records to the MAIN ORDER QUERY, ORDER SUBFORM QUERY, ORDER TABLE and ORDER_PRODUCT TABLE

Additional Information

Record source – MAIN ORDER QUERY

1. Navigation buttons to view details of different orders placed at the Beltel restaurant.
2. Calculations for the following:
    a. **Main Order Subtotal** - =[ORDER SUBFORM].[Form]![Subtotal]
    b. **Discount Applied -** =[Main Order Subtotal]/100*[Discount (%)]
    c. **Total Owed** - =[Main Order Subtotal]-[Discount Applied]
3. Drop down menus with data sourced from the following:
    a. **MemberID** – MEMBER TABLE (MemberID, Title, Forename, Surname)
    b. **Payment Method** – ORDER TABLE (PaymentMethod)
    c. **StaffID** – STAFF TABLE (StaffID, Title, Forename, Surname)
    d. **ProductID** – ORDER SUBFORM QUERY (ProductID, DishName)
4. Autofill
    a. When **MemberID** is selected, data in the forename, surname, membership type, telephone, payment method and discount fields will be automatically added from their respective tables.
    b. When **StaffID** is selected, data in the forename, surname, telephone, and email address fields will be automatically added from their respective tables.
    c. When **ProductID** is selected, data in the quantity, dish name, price, dish type, calories, vegan suitable, gluten free, contains milk, and item total fields will be automatically added from the query.

Main Switchboard interface



Main Switchboard (Gill Sans MT, Black, Size 28, Bold)

FORM HEADER (Colour #88CBB5)

These black navigation buttons use hyperlinks – this will take the user to appropriate parts of the switchboard, or they can exit the application altogether.

Font – Gill Sans MT, Black, Size 12

BACKGROUND COLOUR (#D8D8D8)

Layout and subsequent commands

Main Menu

- Members (Go to Member Menu Switchboard)
- Staff (Go to Staff Menu Switchboard)
- Orders (Go to Order Menu Switchboard)
- Products (Go to Product Menu Switchboard)
- Quit (Exit Application)

Member Menu

- View Member Details (Open MEMBER Form in Edit Mode)
- View membership details (Open MEMBERSHIP TYPE Form in Edit Mode)
- View medical conditions (Open MEDICAL Form in Edit Mode)
- View members with medical conditions (Open Members with medical conditions report)
- Add New Member (Open MEMBER Form in Add Mode)
- Add membership tier (Open MEMBERSHIP TYPE Form in Add Mode)
- Add new medical condition (Open MEDICAL Form in Add Mode)
- Back (Go to Main Switchboard)

Staff Menu

- View Staff Member Details (Open STAFF Form in Edit Mode)
- Add Staff Member Details (Open STAFF Form in Add Mode)
- Back (Go to Main Switchboard)

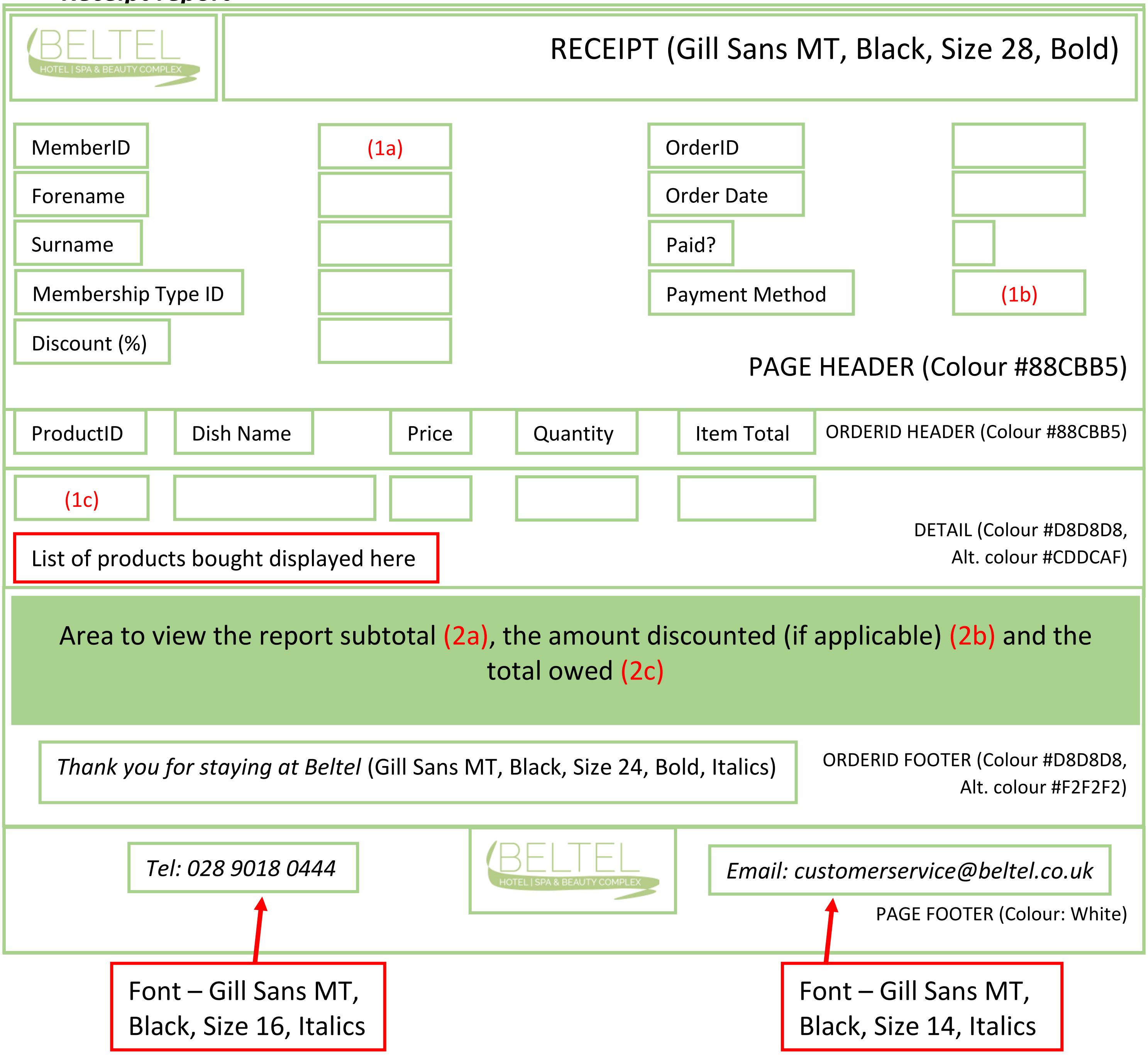
Order Menu

- View Orders (Open MAIN ORDER Form in Edit Mode)
- Add New Order (Open MAIN ORDER Form in Add Mode)
- Archive Old Orders (Run ORDER ARCHIVE MACRO)
- View Restaurant Performance (Open Monthly Sales report)
- View Invoices (Open Invoice report)
- View Receipts (Open Receipt report)
- View most used payment methods (Open Most popular payment methods report)
- Back (Go to Main Switchboard)

Product Menu

- View Products (Open PRODUCT Form in Edit Mode)
- Add New Products (Open PRODUCT Form in Add Mode)
- View best-selling products (Open Best-selling products report)
- Back (Go to Main Switchboard)

Receipt report



RECEIPT (Gill Sans MT, Black, Size 28, Bold)

MemberID (1a)

Forename

Surname

Membership Type ID

Discount (%)

OrderID

Order Date

Paid?

Payment Method (1b)

PAGE HEADER (Colour #88CBB5)

ProductID | Dish Name | Price | Quantity | Item Total

ORDERID HEADER (Colour #88CBB5)

(1c)

List of products bought displayed here

DETAIL (Colour #D8D8D8, Alt. colour #CDDCAF)

Area to view the report subtotal (2a), the amount discounted (if applicable) (2b) and the total owed (2c)

Thank you for staying at Beltel (Gill Sans MT, Black, Size 24, Bold, Italics)

ORDERID FOOTER (Colour #D8D8D8, Alt. colour #F2F2F2)

Tel: 028 9018 0444

Email: customerservice@beltel.co.uk

PAGE FOOTER (Colour: White)

Font – Gill Sans MT, Black, Size 16, Italics

Font – Gill Sans MT, Black, Size 14, Italics

Additional Information

Record source – RECEIPT QUERY

Text size – 12pt

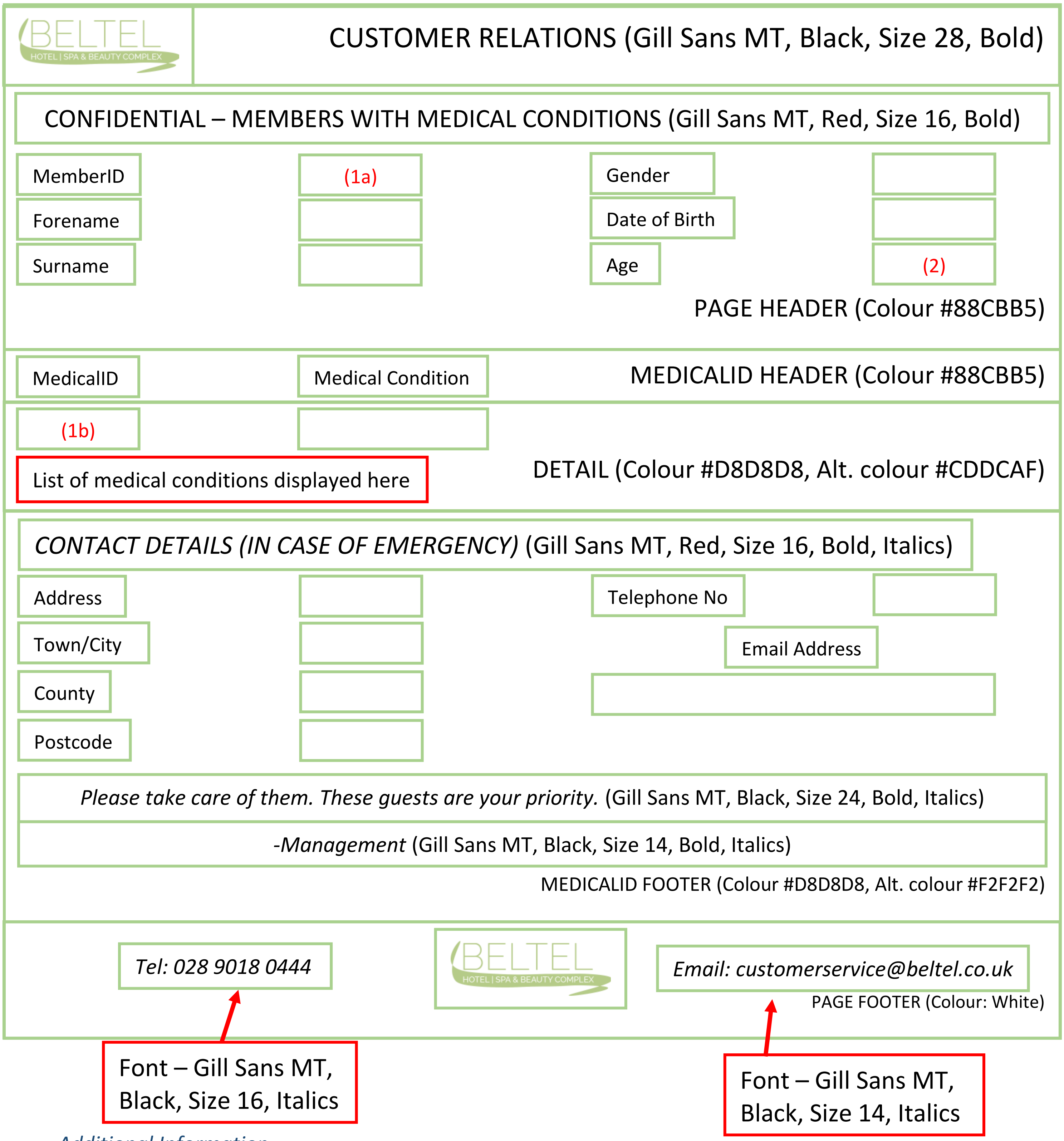
Text colour – Grey (#7F7F7F)

Font – Gill Sans MT

1. Drop down menus with data sourced from the following:
   a. **MemberID** – MemberID from RECEIPT QUERY
   b. **Payment Method** – PaymentMethod from RECEIPT QUERY
   c. **ProductID** – ProductID from RECEIPT QUERY
2. Calculation for the following:
   a. **Report Subtotal -** =Sum([Item Total])

b. **Report Discount -** =[Report Subtotal]/100*[Report Discount (%)]
c. **Total -** =[Report Subtotal]-[Report Discount]

Members with medical conditions report



CUSTOMER RELATIONS (Gill Sans MT, Black, Size 28, Bold)

CONFIDENTIAL – MEMBERS WITH MEDICAL CONDITIONS (Gill Sans MT, Red, Size 16, Bold)

MemberID	(1a)	Gender	
Forename		Date of Birth	
Surname		Age	(2)

PAGE HEADER (Colour #88CBB5)

MedicalID	Medical Condition

MEDICALID HEADER (Colour #88CBB5)

(1b)	

List of medical conditions displayed here

DETAIL (Colour #D8D8D8, Alt. colour #CDDCAF)

CONTACT DETAILS (IN CASE OF EMERGENCY) (Gill Sans MT, Red, Size 16, Bold, Italics)

Address		Telephone No	
Town/City		Email Address	
County			
Postcode			

Please take care of them. These guests are your priority. (Gill Sans MT, Black, Size 24, Bold, Italics)

-*Management* (Gill Sans MT, Black, Size 14, Bold, Italics)

MEDICALID FOOTER (Colour #D8D8D8, Alt. colour #F2F2F2)

Tel: 028 9018 0444

Email: customerservice@beltel.co.uk

PAGE FOOTER (Colour: White)

Font – Gill Sans MT, Black, Size 16, Italics

Font – Gill Sans MT, Black, Size 14, Italics

Additional Information

Record source – Members with medical conditions QUERY

1. Drop down menus with data sourced from the following:
    a. **MemberID** – MemberID from Members with medical conditions QUERY
    b. **MedicalID** – MedicalID from Members with medical conditions QUERY
2. Calculation for **Age -** DateDiff("yyyy",[DOB],Date())

Section 3 – Application Development and Testing

Forms

MEMBER form

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
1.1	MEMBER Form	Valid Module Test	MemberID: 6 Title: Miss Forename: Sophie Surname: Hunt Gender: Female DOB: 27/01/1997 MembershipTypeID: 3 MembershipDateStarted: 09/09/2022 Address Line 1: 75 Hyde Rd Town/City: Paignton County: Devon Postcode: TQ4 5BP TelephoneNo: 08455 441231 Email: sophhunter96@yahoo.com	Input the valid member details and check that they are stored correctly in the MEMBER table	Data accepted and stored in the expected location
1.2	Surname - Presence Check (1)	Valid Unit Test	Hunt	Input a valid surname and check if it is stored correctly in the Surname field of the MEMBER table	Data accepted and stored in the expected location
		Invalid Unit Test		Input no data in that field and check if an error is detected	Error message appeared as expected
1.3	Forename - Length Check (2)	Valid Unit Test	Sophie	Input a valid forename (in this case, up to 20 characters) and ensure that it is stored correctly in the Forename field of the MEMBER table	Data accepted and stored in the expected location
		Invalid Unit Test	Wolfeschlegelsteinhausenbergerdorff	Input a long, invalid forename and ensure that the system doesn't store	As expected, refuses to store data beyond the limit

				more than specified character limit (in this case, beyond 20 characters)	
		Extreme Unit Test	Ashuqkeiqodleodoeqor	Input an extreme forename (in this case, exactly 20 characters) and check that it is stored correctly in the Forename field of the MEMBER table	Data accepted and stored in the expected location
1.4	Postcode - Format Check (3)	Valid Unit Test	TQ4 5BP	Input a valid postcode that respects the format (in this case, L?09\ 0LL) and check if it is stored correctly in the Postcode field of the MEMBER table	Data accepted and stored in the expected location
		Invalid Unit Test	5X0398H	Input an invalid postcode and ensure that the system rejects the data where it doesn't comply with the format	As expected, rejects data that doesn't conform to the format
1.5	Gender – Lookup (4)	Valid Unit Test	Male	Select a valid gender from a list of data items within the form and ensure it is stored correctly in the Gender field of the MEMBER table.	Data accepted and stored in the expected location
		Invalid Unit Test	m	Key in an invalid gender that's not in the list and check that the system rejects the data, and an error occurs	Data rejected and error message appeared as expected

MEMBER_MEDICAL subform

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
2.1	MedicalID – Lookup Table (5)	Valid Unit Test	2	Select a valid medical ID from a list of data items from the MEDICAL table and ensure it is stored correctly in the MedicalID field of the MEMBER_MEDICAL table	Data accepted and stored in the expected location
		Invalid Unit Test	101	Key in an invalid medical ID that's not in the list and check that the system rejects the data, and an error occurs	Data rejected and error message appeared as expected

MAIN ORDER form

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
3.1	MAIN ORDER form and ORDER subform	Valid Module Test	OrderID: 15 Order Date: 11/11/2022 MemberID: 10 Forename: Suzanne Surname: Reynolds Membership Type ID: 4 Telephone No: 02088 688028 Paid: No Payment Method: N/A Discount Applied (%): 50 StaffID: 2 Forename: Jordan Surname: Chapman Telephone No: 02894 258363 Email Address: chapmanJC0@gmail.com ProductID: 16, 11, 7, 10, 4 Quantity: 2, 1, 1, 2, 1 DishName: Garden salad, Tea, Cheese and aubergine lasagne, Semi-skimmed Milk, Pesto and black pepper soup Price: £4, £1, £14.95, £1, £13 DishType: Side, Drink, Main, Drink, Starter Calories: 260, 1, 608, 42, 356 VeganSuitable: Yes, Yes, No, No, No GlutenFree: Yes, Yes, Yes, No, Yes ContainsMilk: No, Yes, Yes, Yes, No Item Total: £8, £1, £14.95, £2, £13 Subtotal: £38.95 Discount Applied (-£): £19.48 Total Owed: £19.48	Input the valid order details and check that they are stored correctly in the ORDER and ORDER_PRODUCT table	Data accepted and stored in the expected location

3.2	MemberID – Autofill (6)	Valid Unit Test	8	Select a valid member ID from a list of data items from the MEMBER table and ensure that the relevant fields (e.g., membership type ID) are stored automatically	Data accepted and relevant fields entered straight away, and stored in the expected location
		Invalid Unit Test	105	Key in an invalid member ID that's not in the list and check that the relevant fields remain empty Also ensure that an error occurs when submitting the order	Relevant fields remain empty as expected; no data is stored, and error message appears
3.3	StaffID – Autofill	Valid Unit Test	2	Select a valid Staff ID from a list of data items from the STAFF table and ensure that the relevant fields (e.g., forename) are stored automatically	Data accepted and relevant fields entered straight away, and stored in the expected location
		Invalid Unit Test	79	Key in an invalid Staff ID that's not in the list and check that the relevant fields remain empty Also ensure that an error occurs when submitting the order	Relevant fields remain empty as expected; no data is stored, and error message appears
3.4	Subtotal – Calculation	Valid Unit Test	=[ORDER SUBFORM].[Form]![Subtotal] Subtotal (in ORDER subform): £20 Main Order Subtotal Should be £20	Select valid label names and apply them to the calculation – the main order subtotal should use the subtotal from the ORDER subform. Check that the correct result appears and is stored correctly as part of the MAIN ORDER form	Calculation successful – the data was accepted and stored in the expected location
3.5	Discount Applied - Calculation	Valid Unit Test	=[Main Order Subtotal]/100*[Discount (%)] Subtotal: £20 Discount: 20% Discount Applied should be £4	Select valid label names and apply them to the calculation – the main order subtotal should be divided by 100 before multiplying the discount. Check that the correct result appears	Calculation successful – the data was accepted and stored in the expected location

				and is stored correctly as part of the MAIN ORDER form	
3.6	Total Owed - Calculation	Valid Unit Test	=[Main Order Subtotal]-[Discount Applied] Subtotal: £20 Discount Applied: £4 Total Owed should be £16	Select valid label names and apply them to the calculation – the main order subtotal should be subtracted by the discount applied. Check that the correct result appears and is stored correctly as part of the MAIN ORDER form	Calculation successful – the data was accepted and stored in the expected location

ORDER subform

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
4.1	ProductID - Autofill	Valid Unit Test	16	Select a valid Product ID from a list of data items from the PRODUCT table and ensure that the relevant fields (e.g., dish name) are stored automatically	Data accepted and relevant fields entered straight away, and stored in the expected location
		Invalid Unit Test	420	Key in an invalid Product ID that's not in the list and check that the relevant fields remain empty Also ensure that an error occurs when submitting the order	Relevant fields remain empty as expected; no data is stored, and error message appears
4.2	Subtotal - Calculation	Valid Unit Test	=Sum([Item Total]) Item Total: £2, £10, £8 Subtotal should be £20	The sum of the item total should be calculated using the Item Total field from the ORDER SUBFORM query. Check that the correct result appears and is stored correctly as part of the ORDER subform	Calculation successful – the data was accepted and stored in the expected location

MEMBERSHIP TYPE form

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
5.1	Price Per Month – Type Check (7)	Valid Unit Test	9.99	Key valid (in this case, numeric) data and check that the system automatically formats the data as currency.	As expected, - data formatted as currency, with £ symbol appearing in front.
		Invalid Unit Test	Nine pounds ninety-nine pence	Key invalid (in this case, text) data and check that the system detects an error	Data rejected and error message appeared as expected
5.2	Discount (%) – Range Check (8)	Valid Unit Test	30	Input a valid discount amount (in this case, between and equal to 10 and 50) and ensure that it is stored correctly in the Discount field of the MEMBERSHIP TYPE table	Data accepted and stored in the expected location
		Invalid Unit Test	75	Input an invalid discount amount (in this case, below 10 and beyond 50) and ensure that the system rejects the data, and an error occurs	Data rejected and error message appeared as expected
		Extreme Unit Test	10 or 50	Input an extreme discount amount (in this case, exactly 10 or 50) and ensure that it is stored correctly in the Discount field of the MEMBERSHIP TYPE table	Data accepted and stored in the expected location
5.3	MembershipTypeID – Primary Key (9)	Valid Unit Test	5	When inputting new data, a unique primary key should be auto filled. Ensure that the system accepts the key and that it is stored correctly in the MembershipTypeID field of the MEMBERSHIP TYPE table.	As expected, a unique key has been assigned to each membership type and stored in the expected location
		Invalid Unit Test	2	Input an already used key (or any number) and ensure that the system rejects the data immediately, and an error occurs.	Data rejected and error message appeared as expected. The user cannot edit the field.

Additional Information

There are further validation checks in other parts of the system, not just the ones tested – these are:

1. **Presence checks**
   - All fields of the MEMBER and STAFF forms except Title
   - All fields of the MEDICAL, MEMBERSHIP TYPE and PRODUCT forms, and MEMBER_MEDICAL and ORDER_PRODUCT subforms
   - All fields of the ORDER form except Payment Method
   - All fields of the MAIN ORDER form except Paid and Payment Method
2. **Length checks**
   - All fields of the MEMBER form except MemberID, DOB, Membership Type ID and Membership Date Started
   - MEDICAL form – Medical Condition field
   - MEMBERSHIP TYPE form – Membership Type field
   - PRODUCT form – Dish Name and Dish Type fields
   - All fields of the STAFF form except StaffID and DOB
3. **Format checks**
   - MEMBER form – DOB, Membership Date Started and Telephone No fields
   - STAFF form – DOB and Telephone No fields
   - MAIN ORDER form – Order Date field
   - ORDER form – Date Ordered field
4. **Lookups**
   - PRODUCT form – Dish Type field
   - STAFF form – Pay Grade Code field
   - MAIN ORDER & ORDER form – Payment Method field
5. **Lookup tables**
   - ORDER_PRODUCT subform – OrderID and ProductID fields
   - MAIN ORDER & ORDER form – MemberID and StaffID (MAIN ORDER form only) fields
   - MEMBER form – Membership Type ID field
   - MEMBER_MEDICAL subform – MedicalID field
   - ORDER subform – ProductID field

6. **Autofills**
    - MEMBER_MEDICAL subform – MedicalID field
7. **Type checks**
    - MEMBER form – MemberID, DOB and Membership Date Started fields
    - MEDICAL form – MedicalID field
    - MEMBER_MEDICAL subform – MemberID and MedicalID fields
    - MEMBERSHIP TYPE form – Membership Type ID, Price Per Month and Discount (%) fields
    - PRODUCT form – ProductID, Calories, Price, VeganSuitable, GlutenFree and ContainsMilk fields
    - STAFF form – StaffID and DOB fields
    - ORDER form – All fields except PaymentMethod
    - ORDER_PRODUCT subform – All fields
8. **Range checks**
    - MEMBERSHIP TYPE form – Price Per Month field
    - PRODUCT form – Calories field
    - ORDER_PRODUCT subform – Quantity field
9. **Primary keys**
    - MEMBER form – MemberID field
    - MEDICAL form & MEMBER_MEDICAL subform – MedicalID field
    - MAIN ORDER & ORDER form, & ORDER_PRODUCT subform – OrderID field
    - ORDER subform & PRODUCT form – ProductID field
    - STAFF form – StaffID field

Queries

Best-selling products

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
6.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, PRODUCT and ORDER_PRODUCT).
6.2	Group by DishName	Unit Test	N/A	Each dish should be uniquely grouped, so that there is a price, quantity and total sales for one of each dish. Run the query and ensure that the system has returned the data as expected.	As expected, each dish was grouped into a unique row, with the price, quantity and total sales correctly assigned to one of each dish.
6.3	Number Purchased (Quantity) - Sum	Unit Test	DishName: Cheese and aubergine lasagne Number Purchased (before sum): 5, 2, 13, 12 Number Purchased should be 32	The sum of the no. purchased should be calculated using the Quantity field from the ORDER_PRODUCT table. Run the query and check that the no. purchased is accurate to the total sales for the product.	Calculation successful – as expected, the sum of the no. purchased is displayed correctly.
6.4	Total Sales - Field calculation	Unit Test	Sum([Price]*[Quantity]) Price: £14.95 Number Purchased: 39 Total Sales should be £583.05	This should be calculated by multiplying the price with the sum of the no. purchased. Run the query and ensure that the system has returned the data as expected.	Calculation successful – as expected, the total sales of each product is displayed correctly.
6.5	Sort by Total Sales in descending order	Unit Test	N/A	Run the query; ensure that the highest selling product is displayed first, and the lowest selling product last.	As expected, the system sorts the Total Sales field by descending order.

6.6	Total Sales - Criteria	Unit Test	N/A	Run the query; ensure that the system omits free products. In other words, products that have made £0.	As expected, the system only returns products that have made a profit.

DELETE ORDER

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
7.1	Query deletes the expected data	Unit Test	N/A	Run the query; the system should display a message asking for confirmation before deleting all the expected fields.	System displays the message and removes the expected fields from their respective tables (in this case, ORDER and ORDER_PRODUCT).
7.2	OrderDate - Criteria	Unit Test	N/A	Run the query; check that the system only deletes orders older than six months.	Some orders remain, the system expectedly removes orders older than six months.

INVOICE QUERY

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
8.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, MEMBERSHIP TYPE, MEMBER, ORDER, ORDER_PRODUCT and PRODUCT).
8.2	Paid - Criteria	Unit Test	N/A	Run the query; ensure that the system omits orders that have been paid for.	System returns only unpaid orders, as expected.
8.3	Item Total – Field calculation	Unit Test	[Quantity]*[Price] Quantity: 2	This should be calculated by multiplying the quantity ordered with the item price. Run the query and ensure that	Calculation successful – as expected, the item total

			Price: £4 Item Total should be £8	the system has returned the data as expected.	of each product is displayed correctly.

MAIN ORDER

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
9.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, ORDER, MEMBER, MEMBERSHIP TYPE and STAFF).

Members with medical conditions

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
10.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, MEMBER, MEMBER_MEDICAL and MEDICAL).
10.2	Sort by Forename in ascending order	Unit Test	N/A	Run the query; check that the system returns members in alphabetical order, with forenames beginning with A appearing first and those starting with Z last.	As expected, the system sorts the Forename field by ascending alphabetical order.
10.3	Age – Field calculation	Unit Test	DateDiff("yyyy",[DOB],Date()) Current date: 13/03/2023 DOB: 09/12/2000 Age should be 23 years old	Using the DateDiff formula, a difference should be calculated between the member's DOB and current date which is formatted in years. Run the query and ensure that the system has returned the data as expected.	Calculation successful – as expected, the age of each member is displayed correctly.
10.4	MedicalID - Criteria	Unit Test	N/A	Run the query; check that the system omits members who don't suffer from any medical conditions.	As expected, the system only returns members who've been diagnosed with a medical condition.

Monthly Sales

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
11.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables and queries (in this case, MEMBERSHIP TYPE, MEMBER and ORDER tables, and ORDER SUBFORM and Monthly Sales Discount queries).
11.2	Group by Order Year and Month Number	Unit Test	N/A	This should group the order year and month number together, which should enable the system to return the total for one of each month of a particular year. Run the query and ensure that the system has returned the data as expected.	As expected, each month of a particular year was grouped into a unique row, with the item total and discount applied correctly assigned.
11.3	Order Year – Field calculation	Unit Test	Year([OrderDate]) OrderDate: 07/11/2022 Order Year should be 2022	This should be calculated using the Year() function, taking the year of each order date and displaying it only. Run the query and check that the Order Year for each month is accurate.	Calculation successful – as expected, the year for each month an order was taken is displayed correctly.
11.4	Month Number – Field calculation	Unit Test	Month([OrderDate]) OrderDate: 07/11/2022 Month Number should be 11	This should be calculated using the Month() function, taking the month number of each order date and displaying it only. Run the query and check that the Month Number for each month is accurate.	Calculation successful – as expected, the month number for each month an order was taken is returned correctly e.g., 1 is displayed alongside January

11.5	Month – Field calculation	Unit Test	MonthName(Month([OrderDate])) OrderDate: 07/11/2022 Month should be November	The month should be calculated using the MonthName() function. Run the query and ensure it returns the month name based on the number of the order date.	Calculation successful – as expected, the name for each month that an order was taken is returned correctly e.g., 1 is translated to become January
11.6	Item Total - Sum	Unit Test	Item Total (before sum): £28, £57, £84, £74 Item Total should be £243	The sum of the item total should be calculated using the Item Total field from the ORDER SUBFORM query. Run the query and check that the Item Total for each month is accurate.	Calculation successful – as expected, the sum of the monthly item total is displayed correctly.
11.7	Discount Applied - Sum	Unit Test	Discount Applied (before sum): £16, £48, £76, £50 Discount Applied should be £190	The sum of the discount should be calculated using the Discount Applied field from the Monthly Sales Discount query. Run the query and check that the Discount Applied for each month is accurate – it should be lower than the item total.	Further inquiry needed – some field values are higher than their item total (i.e., before discount was applied). Unsure if this is an error or because of lost profits.
11.8	Order Year - Criteria	Unit Test	[Please specify order year] 2022	Run the query; check that the system displays a message prompting the user to input the monthly sales for a specific year. For example, inputting 2022 should return the monthly sales for 2022 only.	As expected, the message appears and returns the required fields subject to what the user has entered.

<u>Monthly Sales Discount</u>

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
12.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables and queries (in this case, MEMBERSHIP TYPE, MEMBER and ORDER tables, and ORDER SUBFORM query).
12.2	Group by OrderID	Unit Test	N/A	Each order should be uniquely grouped, with the item total assigned to the correct order. Run the query and ensure that the system has returned the data as expected.	As expected, each order was grouped into a unique row, with the item total correctly assigned.
12.3	Item Total - Sum	Unit Test	Item Total (before sum): £2, £10, £8 Item Total should be £20	The sum of the item total should be calculated using the Item Total field from the ORDER SUBFORM query. Run the query and check that the Item Total for each order is accurate.	Calculation successful – as expected, the sum for each order item total is displayed correctly.
12.4	Discount Applied – Field calculation	Unit Test	Sum([Item Total]/100*[Discount (%)]) Subtotal: £20 Discount: 20% Discount Applied should be £4	This should be calculated by dividing the sum of the item total by 100 and multiplying the calculated value with the discount. Run the query and ensure that the system has returned the data as expected.	Calculation successful – as expected, the discount applied for each order is displayed correctly.

Most popular payment method

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
13.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective table (in this case, ORDER).
13.2	Group by PaymentMethod	Unit Test	N/A	Each payment method should be (in this case) uniquely grouped into six, with the frequency assigned to the correct method. Run the query and ensure that the system has returned the data as expected.	As expected, each payment method was grouped into six with the frequency correctly assigned to each type of payment method.
13.3	Sort by CountOfPaymentMethod in descending order	Unit Test	N/A	Run the query; ensure that the most frequent payment method is displayed first, and the least frequent method last.	As expected, the system sorts the CountOfPaymentMethod field by descending order.
13.4	PaymentMethod - Count	Unit Test	PaymentMethod: PayPal, PayPal, PayPal CountOfPaymentMethod should be 3	Run the query; the system should correctly count the times each payment was used when paying for an order.	As expected, the system returns the frequency of each type of payment method correctly.
13.5	Paid - Criteria	Unit Test	N/A	Run the query; check that the system omits unpaid orders and those that haven’t selected a payment method.	As expected, the system returns only paid orders that have used a payment method.

ORDER ARCHIVE QUERY

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
14.1	Query appends the expected data	Unit Test	N/A	Run the query; the system should display a message asking for confirmation before appending all the expected fields.	System displays the message, returns the expected fields and appends the data into an ORDER ARCHIVE table.
14.2	OrderDate - Criteria	Unit Test	N/A	Run the query; check that the system only appends orders older than six months.	Some orders remain unchanged, the system expectedly appends orders older than six months to the ORDER ARCHIVE table.

ORDER SUBFORM

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
15.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, ORDER_PRODUCT and PRODUCT).
15.2	Item Total – Field calculation	Unit Test	[Quantity]*[Price] Quantity: 2 Price: £4 Item Total should be £8	This should be calculated by multiplying the quantity ordered with the item price. Run the query and ensure that the system has returned the data as expected.	Calculation successful – as expected, the item total of each product is displayed correctly.

RECEIPT QUERY

Test Number	**Attribute & Test**	**Test Type**	**Test Data**	**Method and Expected Outcome**	**Outcome**
16.1	Query returns the expected data	Unit Test	N/A	Run the query; the system should return all the expected fields.	System returns the expected fields and fetches them from their respective tables (in this case, MEMBERSHIP TYPE, MEMBER, ORDER, ORDER_PRODUCT and PRODUCT).
16.2	Paid - Criteria	Unit Test	N/A	Run the query; ensure that the system omits orders that haven't been paid for.	System returns only paid orders, as expected.
16.3	Item Total – Field calculation	Unit Test	[Quantity]*[Price] Quantity: 2 Price: £4 Item Total should be £8	This should be calculated by multiplying the quantity ordered with the item price. Run the query and ensure that the system has returned the data as expected.	Calculation successful – as expected, the item total of each product is displayed correctly.

Reports

Best-selling products

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
17.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and no fields are cut off.
17.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the dish name, price of each dish, quantity ordered and total sales from each dish)	Report expectedly shows all the data required for it to be beneficial and fetches them from the relevant source.
17.3	Sort by Total Sales in descending order	Unit Test	N/A	Ensure that the report shows the highest selling product first, and the lowest selling product last.	As expected, the report sorts Total Sales by descending order.

Invoice Report

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
18.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and no fields are cut off.
18.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the member and order details)	Report expectedly shows all the data required for it to be beneficial and fetches them from the relevant source.
18.3	Group by OrderID (Invoice)	Unit Test	N/A	The report should display all purchases made by a customer on the same order/invoice	As expected, all products are grouped together on the same order
18.4	Subtotal - Calculation	Unit Test	=Sum([Item Total]) Item Total: £1, £8, £12 Subtotal should be £21	The report subtotal should be calculated using the sum of the Item Total field from the INVOICE query. View the report and check that the subtotal for each order is accurate.	Calculation successful – as expected, the subtotal for each order is displayed correctly.
18.5	Discount Applied - Calculation	Unit Test	=[Report Subtotal]/100*[Report Discount (%)] Subtotal: £20 Discount: 20% Discount Applied should be £4	The report discount should be calculated by dividing the report subtotal by 100 then multiplying with the discount applied for each order. View the report and check that the discount applied for each order is accurate.	Calculation successful – as expected, the discount applied for each order is displayed correctly.
18.6	Total Owed - Calculation	Unit Test	=[Report Subtotal]-[Report Discount] Subtotal: £20 Report Discount: £4	The total owed for each order should be calculated by subtracting the discount applied from the subtotal.	Calculation successful – as expected, the total owed for each order is displayed correctly.

			Total Owed should be £16	View the report and check that the total owed for each order is accurate.	

Members with medical conditions

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
19.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and no fields are cut off.
19.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the member, medical and contact details)	Report expectedly shows all the data required for it to be beneficial and fetches them from the relevant source.
19.3	Group by MedicalID (Member)	Unit Test	N/A	The report should display all medical conditions that each member suffers from on the same page.	Fixes required – member details only appear once whereas medical and contact details are unnecessarily repeated on the one page. Will have to look further at how it's grouped.

Monthly Sales

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
20.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and stays on one page.
20.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the total sales for each month of the year and the grand total)	Fix required - report fails to show all the data required for it to be beneficial. The grand total is missing.
20.3	Group by Order Year	Unit Test	N/A	The report should display all monthly sales for each year on the same page.	As expected, all monthly sales are grouped together on each year.
20.4	Sort by Month Number by ascending order	Unit Test	N/A	The report should ensure that each month of the year is displayed in chronological order.	As expected, the reports sort each month by chronological order.
20.5	Total Sales (after discounts) – calculation	Unit Test	=[Total Sales BD]-[Discount Applied] Total Sales BD: £500 Discount Applied: £50 Total Sales (after discounts) should be £450	This should be calculated by subtracting the discount applied from the total sales (before discounts). View the report and check that the total sales (after discounts) for each month is accurate.	Further inquiry needed – the calculation is accurate, but some field values are negative. Unsure if this is an error or because of lost profits.
20.6	Grand Total - calculation	Unit Test	=Sum([Total Sales AD]) Total Sales AD: £250, £378, £640 Grand Total should be £1268	The grand total should be calculated using the SUM formula to total all monthly sales after the discount was applied. View the report and check that the grand total for each year is accurate.	Calculation failed – the system displays a message asking to key a parameter, which shouldn't happen.

					Otherwise, the grand total appears blank.
20.7	Order Year - Parameter	Unit Test	2022	Before the report is presented, the system should display a message prompting the user to input the monthly sales for a specific year. For example, inputting 2022 should return the monthly sales for 2022 only.	As expected, the message appears and presents the data required subject to what the user has entered.

Most popular payment method

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
21.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and no fields are cut off.
21.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the member, medical and contact details)	Report expectedly shows all the data required for it to be beneficial and fetches them from the relevant source.
21.3	Sort by CountOfPaymentMethod by descending order	Unit Test	N/A	View the report; ensure that the most frequent payment method is displayed first, and the least frequent method last.	As expected, the system sorts the CountOfPaymentMethod field by descending order.

Receipt report

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
22.1	Report layout	Unit Test	N/A	The report should present all the data in a professional manner – it should follow the brand's colour scheme, be intuitive to read and understand, and should all be kept in one page.	As expected, data is laid out professionally, follows the colour scheme and no fields are cut off.
22.2	All data required	Unit Test	N/A	The report should display all the necessary data to ensure that it is useful (in this case, the member and order details)	Report expectedly shows all the data required for it to be beneficial and fetches them from the relevant source.
22.3	Group by OrderID (Invoice)	Unit Test	N/A	The report should display all purchases made by a customer on the same order/receipt.	As expected, all products are grouped together on the same order.
22.4	Subtotal - Calculation	Unit Test	=Sum([Item Total]) Item Total: £1, £7, £12 Subtotal should be £20	The report subtotal should be calculated using the sum of the Item Total field from the RECEIPT query. View the report and check that the subtotal for each order is accurate.	Calculation successful – as expected, the subtotal for each order is displayed correctly.
22.5	Discount Applied - Calculation	Unit Test	=[Report Subtotal]/100*[Report Discount (%)] Subtotal: £20 Discount: 20% Discount Applied should be £4	The report discount should be calculated by dividing the report subtotal by 100 then multiplying with the discount applied for each order. View the report and check that the discount applied for each order is accurate.	Calculation successful – as expected, the discount applied for each order is displayed correctly.
22.6	Total Owed - Calculation	Unit Test	=[Report Subtotal]-[Report Discount] Subtotal: £20 Report Discount: £4	The total owed for each order should be calculated by subtracting the discount applied from the subtotal.	Calculation successful – as expected, the total owed for each order is displayed correctly.

			Total Owed should be £16	View the report and check that the total owed for each order is accurate.	

Menu System (Switchboard)

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
23.1	Open menu system automatically	Unit Test	N/A	When opening the Beltel database in Microsoft Access, the switchboard should be displayed immediately.	Fix required – switchboard fails to open as expected when accessing the database. Will have to look through program options to enable this feature.
23.2	Button – Go to Member Menu Switchboard	Unit Test	N/A	Access the switchboard form and in the main menu click the "Members" button. It should be hyperlinked to Member Menu switchboard.	As expected, the "Members" button takes you to the Member Menu switchboard.
23.3	Button - Open MEMBER Form in Edit Mode	Unit Test	N/A	Access the switchboard form, navigate to the member menu and click the "View Member Details" button. It is linked to the MEMBER form and should show a record with its fields input.	As expected, the "View Member Details" button opens the MEMBER form, displaying an existing record.
23.4	Button – Open MEMBER form in Add Mode	Unit Test	N/A	Access the switchboard form, navigate to the member menu and click the "Add New Member" button. It is linked to the MEMBER form and should show a blank record.	As expected, the "Add New Member" button opens the MEMBER form, displaying an empty record ready for data input.
23.5	Button – Exit Application	Unit Test	N/A	Access the switchboard form and click the "Quit" button. It should exit the database.	The "Quit" button functions as expected to exit the database.
23.6	Button – Run ORDER ARCHIVE MACRO	Unit Test	N/A	Access the switchboard form, navigate to the order menu and click the "Archive Old Orders" button. It is linked	As expected, the "Archive Old Orders" button performs the macro.

				to the ORDER ARCHIVE MACRO and should execute the macro.	
23.7	Button - Open Members with medical conditions report	Unit Test	N/A	Access the switchboard form, navigate to the member menu and click the “View members with medical conditions” button. It is linked to the Members with medical conditions report and should open it.	As expected, the “View members with medical conditions” button opens the report.

Additional Information

There are further buttons within the menu system, not just the ones tested. More tests can be applied to:

- Test 23.2 - Go to Main/Staff Menu/Order Menu/Product Menu Switchboard
- Test 23.3 - Open STAFF/MAIN ORDER/MEDICAL/MEMBERSHIP TYPE/PRODUCT form in Edit Mode
- Test 23.4 - Open STAFF/MAIN ORDER/MEDICAL/MEMBERSHIP TYPE/PRODUCT form in Add Mode
- Test 23.7 - Open Monthly Sales/Invoice/Receipt/Most popular payment methods/Best-selling products report

Macros

ORDER ARCHIVE MACRO

Test Number	Attribute & Test	Test Type	Test Data	Method and Expected Outcome	Outcome
24.1	Warning message appears	Unit Test	N/A	Run the macro; the system should display a message asking for confirmation before executing the macro.	Message box appears as expected. The user selects “OK” to proceed.
24.2	Run ORDER ARCHIVE QUERY	Unit Test	N/A	The system should now proceed by running the ORDER ARCHIVE QUERY. It should also display another message asking for confirmation.	As expected, the message box appears, and the system executes the ORDER ARCHIVE QUERY once the user selects “Yes”.
24.3	Run DELETE ORDER query	Unit Test	N/A	After running the ORDER ARCHIVE QUERY, the system should run the DELETE ORDER query. It should also display another message asking for confirmation.	As expected, the message box appears, and the system executes the DELETE ORDER query once the user selects “Yes”.
24.4	Information message appears	Unit Test	N/A	After the system runs the DELETE ORDER query, it should display a message notifying the user that the process has been completed.	Message box appears as expected.

User requirements

- **(Refer to test 1.1) Adding/modifying member details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot print receipts/invoices without knowing who ordered.
- **(Refer to test 1.1 or 3.1) Adding/modifying product details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot place orders when you don't have products to order from.
- **(Refer to test 3.1) Adding/modifying order details** – The backbone of the ordering system. The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot keep track of, for example, unpaid orders without a table to store them.
    - **(Refer to test 9.1) In addition, a query to view/store main order details must be created** – it provides the control source necessary for the MAIN ORDER form, otherwise it is redundant. As such, a list of orders must be returned no later than 10 seconds after the query is run.
    - **(Refer to test 15.1) There should also be a query for the order subform** – it provides the control source necessary for the ORDER SUBFORM and is also vital for some fields in the MAIN ORDER form, such as the subtotal. A list of products purchased for each order must be returned no later than 10 seconds after the query is run.
- **(Refer to test 1.1 or 3.1) Adding/modifying medical details** – Vital for knowing who needs special care from staff. The form will be an intuitive way to input the data and must be processed in less than 5 seconds. You don't want a member with a peanut allergy allowed to order a product containing peanuts.
- **(Refer to tests 3.5, 3.6, 4.2, 6.3, 6.4, 8.3, 10.3, 11.6, 11.7, 12.3, 12.4, 13.4, 15.2, 16.3, 18.4 - 18.6, 20.5, 20.6 and 22.4 - 22.6) Calculating totals** – Necessary for orders/sales. Without a subtotal, it's hard to tell how much Beltel has made from an order. The calculations must be accurate, otherwise problems will arise if false profits/losses are recorded.
- **(Refer to tests 3.4 and 11.3 – 11.5) Calculating miscellaneous fields** – Also necessary for orders/sales. Without an order year, for example, it will be very difficult to calculate the monthly sales for each year. The calculations must be accurate, otherwise problems will arise if the wrong fields are recorded.
- **(Refer to test 1.1 or 3.1) Adding/modifying membership details** – Necessary when applying discounts to orders, otherwise members will be inconvenienced. The form will be an intuitive method to input the data and must be processed in less than 5 seconds.
- **(Refer to test 1.1 or 3.1) Adding/modifying staff details** – The form will be an intuitive method to input the data and must be processed in less than 5 seconds. You cannot place orders when you don't know who took the order.
- **(Refer to tests 18.1, 18.2, 22.1 and 22.2) Producing a receipt/invoice** – Vital for knowing what a member has ordered, and how much they paid/owe. Management can easily contact members with an invoice. The report must be produced in less than 10 seconds and presented in a professional manner.
- **(Refer to tests 8.1 and 16.1) Creating queries (searching) for invoices/receipts** – they provide the control source for their respective reports, otherwise they are inessential. As such, a list of paid/unpaid orders must be returned no later than 10 seconds after the query is run.

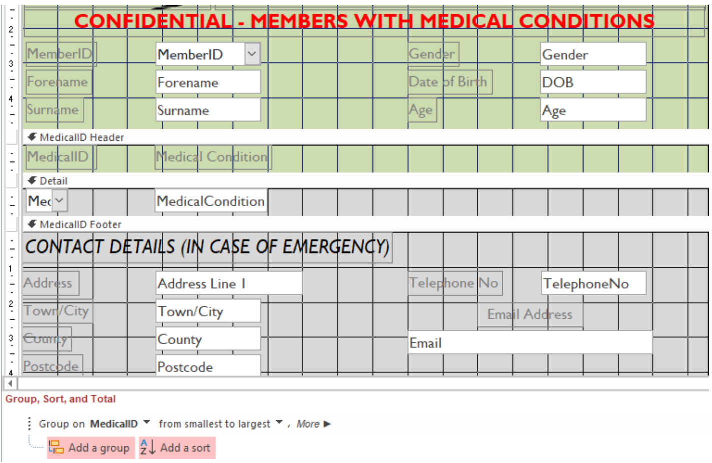
- **(Refer to test 10.1) Creating a query (searching) for members with medical conditions** – it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of members suffering with medical conditions must be returned no later than 10 seconds after the query is run.
- **(Refer to test 11.1) Creating a query for monthly sales** – it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of profits from each month of the year (with the discount applied) must be returned no later than 10 seconds after the query is run.
    - **(Refer to test 12.1) In addition, a query must be added that applies the discounts for all orders which then must be grouped and linked to the query above.** A list of orders (with the discount applied) must be returned no later than 10 seconds after the query is run. They rely on each other, and hence vital for management when evaluating losses due to discounts.
- **(Refer to test 6.1) Creating a query (searching) for the best-selling products** - it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of the highest selling products at the restaurant must be returned no later than 10 seconds after the query is run.
- **(Refer to test 13.1) Creating a query (searching) for the most popular payment method** - it provides the control source necessary for its corresponding report, otherwise it is left inessential. As such, a list of payment methods (starting by the most frequent) must be returned no later than 10 seconds after the query is run.
- **(Refer to tests 20.1 and 20.2) Monthly sales report system** – this assists management when evaluating the restaurant's performance. It must be produced in less than 10 seconds and presented in a professional manner. All it does is give insight into Beltel's performance; the system can function without it.
- **(Refer to tests 17.1 and 17.2) Best-selling products report system** – this assists management when evaluating the restaurant's menu. It must omit free products (since they will not make a profit), be produced in less than 10 seconds and should be presented in a professional manner. However, all it does is give insight into Beltel's performance; the system can function without it.
- **(Refer to tests 19.1 and 19.2) Members with medical conditions report** – this notifies management of members suffering from medical conditions; however, this can be discovered by other means, not just through a report. Nevertheless, once it is developed, it must be produced in less than 10 seconds and presented in a professional manner.
- **(Refer to tests 23.1 – 23.7) Developing a switchboard user interface** – this provides a more user-friendly experience when interacting with the database. Navigation should be quick; load times should take less than 4 seconds along with an intuitive layout. Nevertheless, interaction is possible without the switchboard at the expense of intuitiveness.
- **(Refer to test 24.1 – 24.4) Archiving orders** – orders older than six months, for example, will be archived to increase efficiency and decrease storage space. The macro required for this operation must process the data in less than 5 seconds, with message boxes to notify the user of its status. Development can begin after the first release since there'll be no orders to archive, although it must be completed before the deadline.

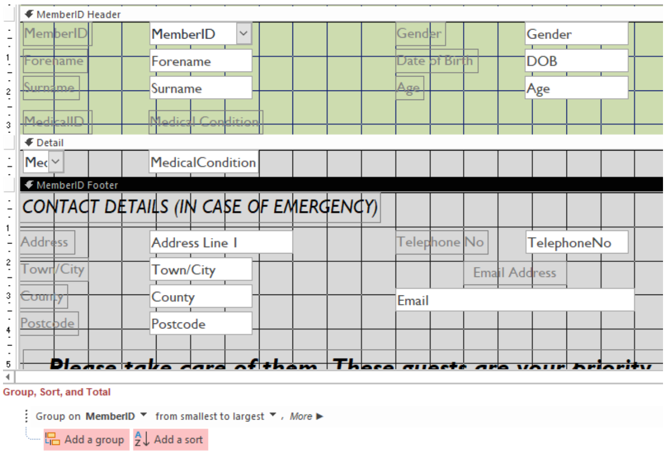
    - **(Refer to test 7.1 and 14.1) Two queries are required for this macro** – one must append the data required to a new table, while the other must delete the specific data from their old tables. Both must be performed no more than 5 seconds.
- **(Not yet developed) Preventing members with allergies from ordering flagged products** – for example, a staff member should receive an error message no later than 3 seconds after a member with a milk allergy attempts to order a product containing milk. However, this can be achieved by other means, such as highlighting allergens on the menu.
- **(Not yet developed) Show a splash screen on start-up** – this should be presented professionally to the user as the system starts up. It must show the Beltel logo and follow the brand's colour scheme. All it does is notify the user that the system is loading; it can function without it.

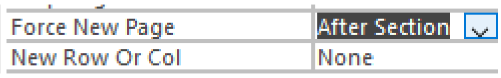
Corrective action

Test 19.3

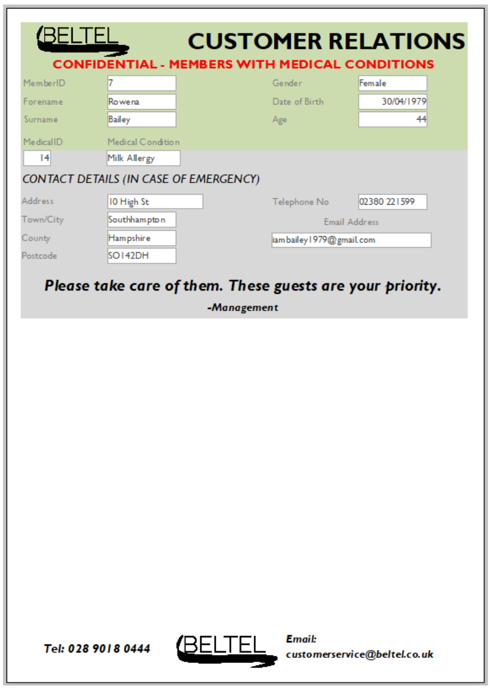
The members with medical conditions report didn’t display all the data correctly. Member details only appeared once, and medical and contact details were being repeated; they should only be displayed once for each member. This was fixed by changing the grouping to be by MemberID rather than MedicalID, moving member details from the page header to the MemberID header and by forcing a new page after the MemberID footer.







Force New Page	After Section
New Row Or Col	None



BELTEL CUSTOMER RELATIONS

CONFIDENTIAL - MEMBERS WITH MEDICAL CONDITIONS

MemberID	7	Gender	Female
Forename	Rowena	Date of Birth	30/04/1979
Surname	Bailey	Age	44

MedicalID	Medical Condition
14	Milk Allergy

CONTACT DETAILS (IN CASE OF EMERGENCY)

Address	10 High St	Telephone No	02380 221599
Town/City	Southhampton	Email Address	
County	Hampshire		iambailey1979@gmail.com
Postcode	SO142DH		

Please take care of them. These guests are your priority.

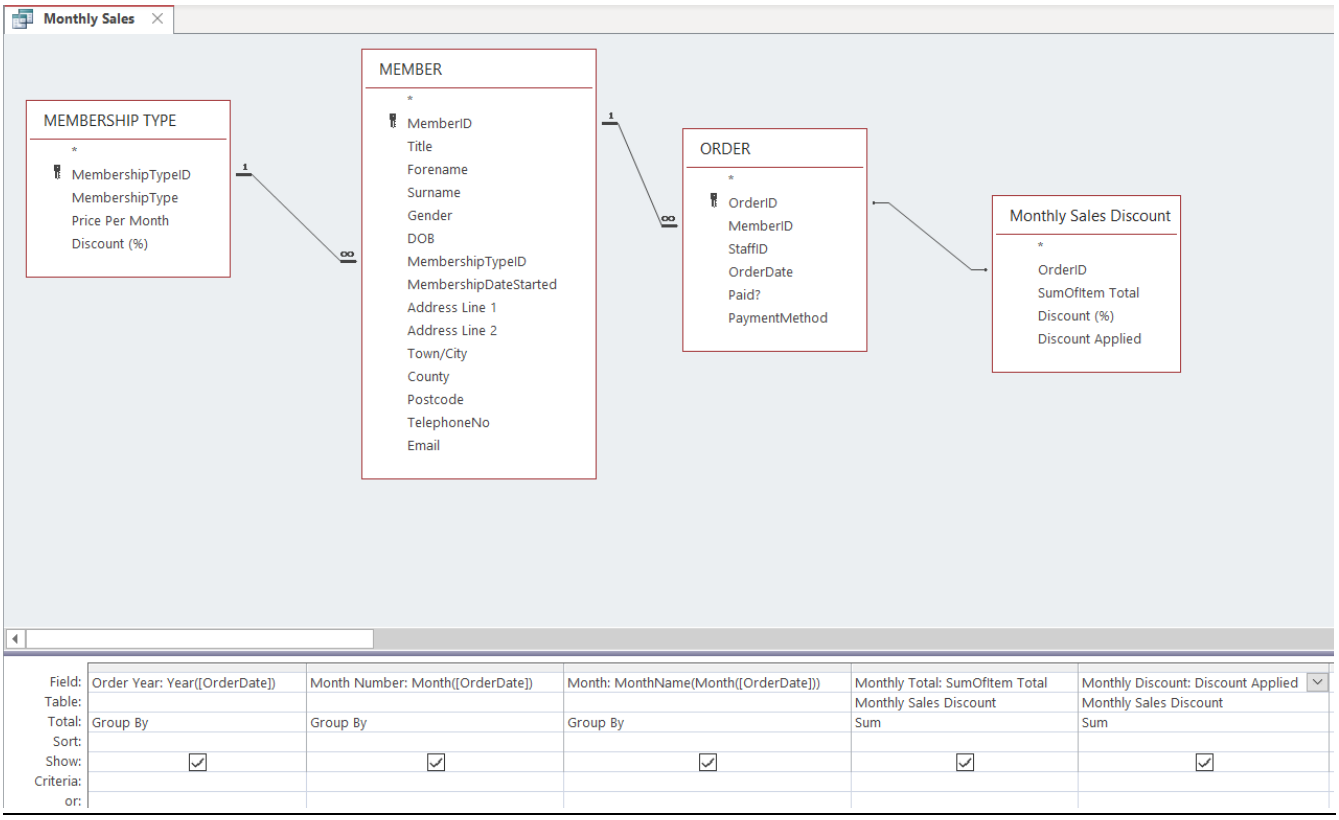
-Management

Tel: 028 9018 0444 BELTEL Email: customerservice@beltel.co.uk

Member, medical and contact details are now displayed for each member

Test 11.7/20.2/20.5/20.6

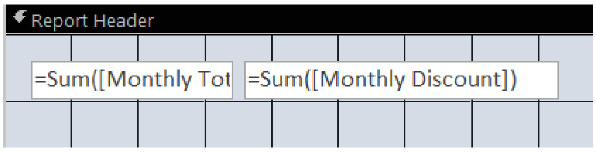
The monthly discounts were returning inaccurate results. This affected the values for the monthly sales report as it erroneously listed negative profits for some months. This was fixed by redesigning the query – the item total from the ORDER SUBFORM query was providing the wrong information and thus was replaced by the sum of item total from the Monthly Sales Discount query.



Field:	Order Year: Year([OrderDate])	Month Number: Month([OrderDate])	Month: MonthName(Month([OrderDate]))	Monthly Total: SumOfItem Total	Monthly Discount: Discount Applied
Table:				Monthly Sales Discount	Monthly Sales Discount
Total:	Group By	Group By	Group By	Sum	Sum
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

The redesigned Monthly Sales query

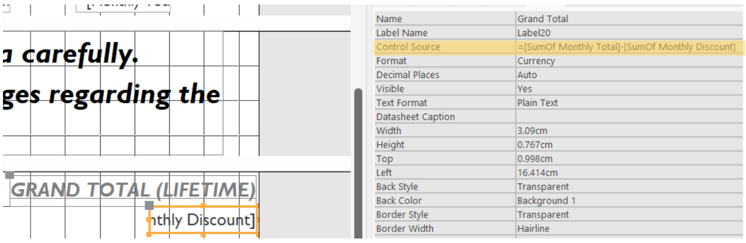
The monthly sales report also had to be fixed – first, the query had to be saved so that the report would return the new totals. New calculations were then created for the monthly and discount totals which were then subtracted from each other to give the grand total.



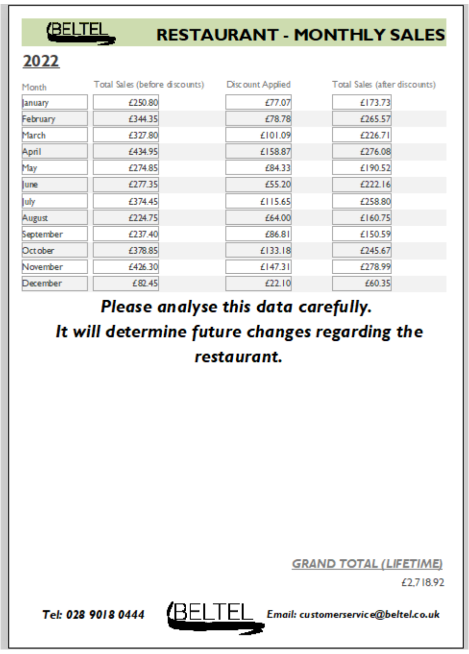
=Sum([Monthly Tot

=Sum([Monthly Discount])

The new calculations required for the grand total



Name	Grand Total
Label Name	Label20
Control Source	=[SumOf Monthly Total]-[SumOf Monthly Discount]
Format	Currency
Decimal Places	Auto
Visible	Yes
Text Format	Plain Text
Datasheet Caption	
Width	3.09cm
Height	0.767cm
Top	0.998cm
Left	16.414cm
Back Style	Transparent
Back Color	Background 1
Border Style	Transparent
Border Width	Hairline



BELTEL RESTAURANT - MONTHLY SALES

2022

Month	Total Sales (before discounts)	Discount Applied	Total Sales (after discounts)
January	£250.80	£77.07	£173.73
February	£344.35	£78.78	£265.57
March	£327.80	£101.09	£226.71
April	£434.95	£158.87	£276.08
May	£274.85	£84.33	£190.52
June	£277.35	£55.20	£222.16
July	£374.45	£115.65	£258.80
August	£224.75	£64.00	£160.75
September	£237.40	£86.81	£150.59
October	£378.85	£133.18	£245.67
November	£426.30	£147.31	£278.99
December	£82.45	£22.10	£60.35

Please analyse this data carefully.
It will determine future changes regarding the restaurant.

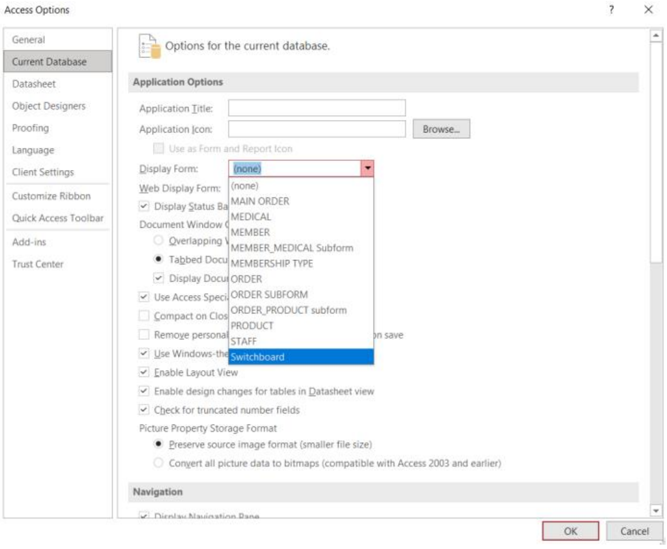
GRAND TOTAL (LIFETIME)
£2,718.92

Tel: 028 9018 0444 BELTEL Email: customerservice@beltel.co.uk

The report now displays the correct calculations

Test 23.1

When opening the Beltel database in Microsoft Access, the switchboard failed to be displayed immediately. This was a simple fix – in the database, select File > Options > Current database > Display Form and select Switchboard from the dropdown menu. Close and reopen the database to retest.



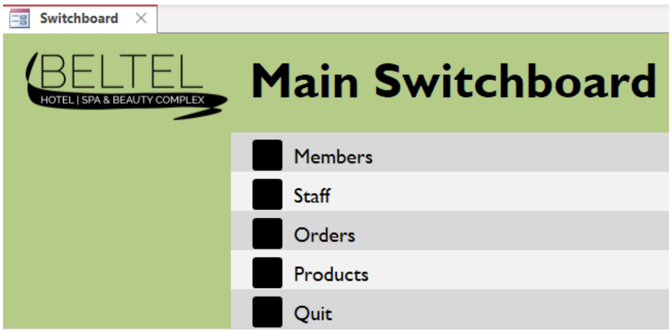
Access Options

Options for the current database.

Application Options

Application Title:
Application Icon:
Display Form: (none)

☐ Use as Form and Report Icon
Web Display Form:
☑ Display Status Ba
Document Window O
○ Overlapping W
● Tabbed Docu
☑ Display Docu
☑ Use Access Speci
☐ Compact on Clos
☐ Remove personal
☑ Use Windows-the
☑ Enable Layout View
☑ Enable design changes for tables in Datasheet view
☑ Check for truncated number fields
Picture Property Storage Format
● Preserve source image format (smaller file size)
○ Convert all picture data to bitmaps (compatible with Access 2003 and earlier)
☑ Display Navigation Pane



The switchboard now opens on start-up

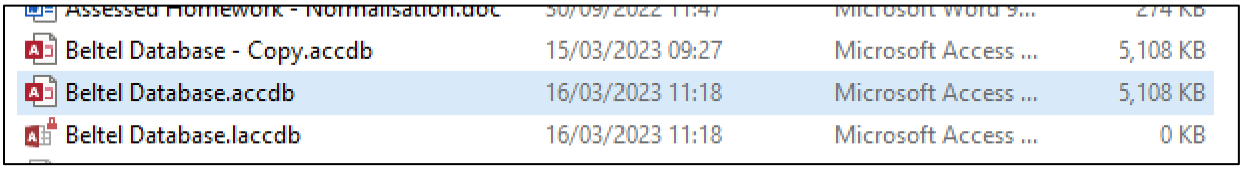
System testing

Below are examples of scenarios that will require extensive testing of various aspects of the system:

1. **Processing an order for a new customer**

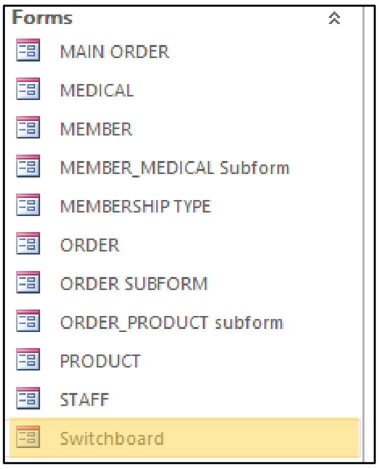
   Beltel has seen the arrival of a new member, Scott Reid. On his first day, he decides to go to the restaurant and orders the pesto and black pepper soup, mushroom and peppercorn stew, peach and almond buns, and semi-skimmed milk. His arrival being so recent, however, Beltel haven't stored his details into the database. As such, the waiter will have to do the following:

   - Open Microsoft Access and select the file "Beltel Database.accdb"



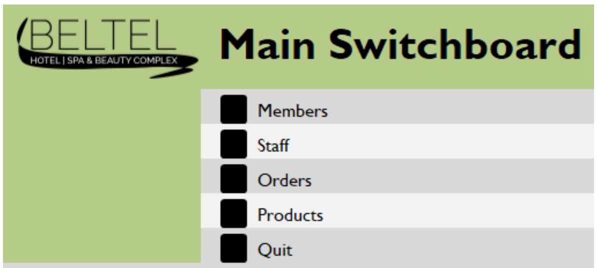
Assessed Homework - Normalisation.doc	30/09/2022 11:47	Microsoft Word 9...	274 KB
Beltel Database - Copy.accdb	15/03/2023 09:27	Microsoft Access ...	5,108 KB
Beltel Database.accdb	16/03/2023 11:18	Microsoft Access ...	5,108 KB
Beltel Database.laccdb	16/03/2023 11:18	Microsoft Access ...	0 KB

   - The database should load into the menu system. If it doesn't, the staff member can select "Switchboard" under the Forms section.

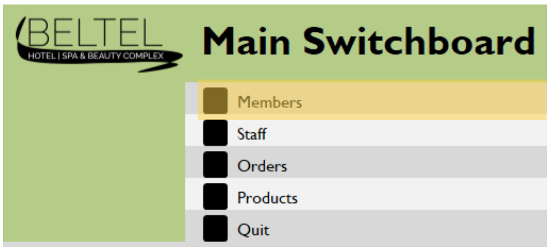


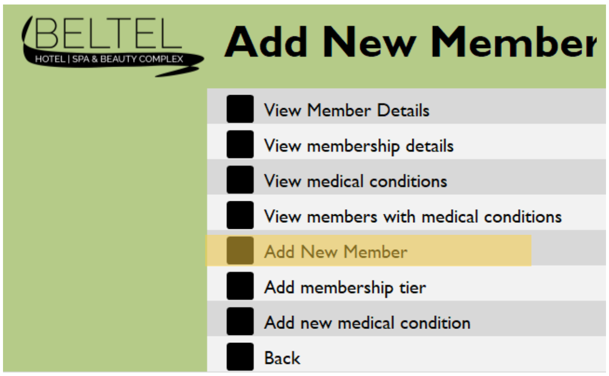
Forms

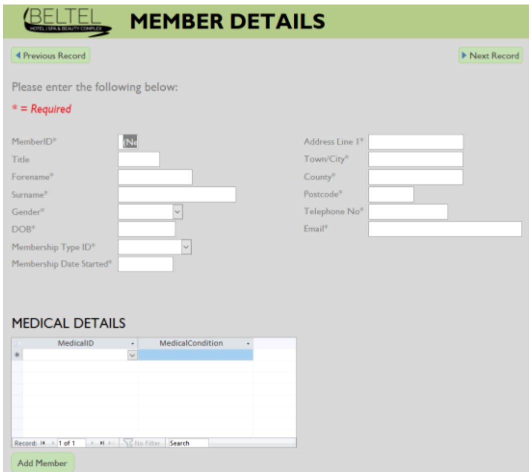
- MAIN ORDER
- MEDICAL
- MEMBER
- MEMBER_MEDICAL Subform
- MEMBERSHIP TYPE
- ORDER
- ORDER SUBFORM
- ORDER_PRODUCT subform
- PRODUCT
- STAFF
- Switchboard



- Navigate through the menu system. First by selecting “Members”, then in the Member Menu selecting “Add New Member” which should open the MEMBER form displaying an empty record ready for data input.







Please enter the following below:

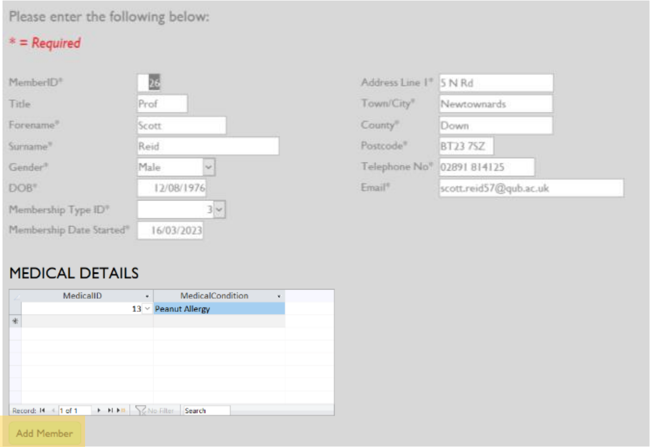
* = Required

MemberID*	(Ne	Address Line 1*	
Title		Town/City*	
Forename*		County*	
Surname*		Postcode*	
Gender*		Telephone No*	
DOB*		Email*	
Membership Type ID*			
Membership Date Started*			

MEDICAL DETAILS

MedicalID	MedicalCondition

- Enter all the details required, then select “Add Member”. This should store Scott Reid’s details into the MEMBER table.



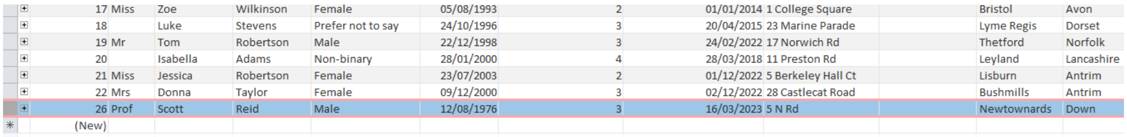
Please enter the following below:

* = Required

MemberID*	26	Address Line 1*	5 N Rd
Title	Prof	Town/City*	Newtownards
Forename*	Scott	County*	Down
Surname*	Reid	Postcode*	BT23 7SZ
Gender*	Male	Telephone No*	02891 814125
DOB*	12/08/1976	Email*	scott.reid57@qub.ac.uk
Membership Type ID*	3		
Membership Date Started*	16/03/2023		

MEDICAL DETAILS

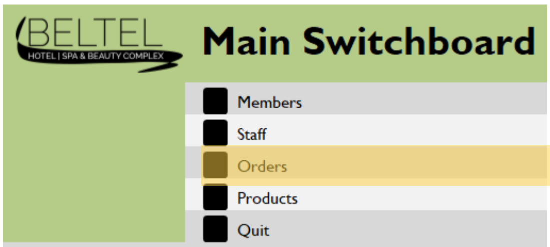
MedicalID	MedicalCondition
13	Peanut Allergy

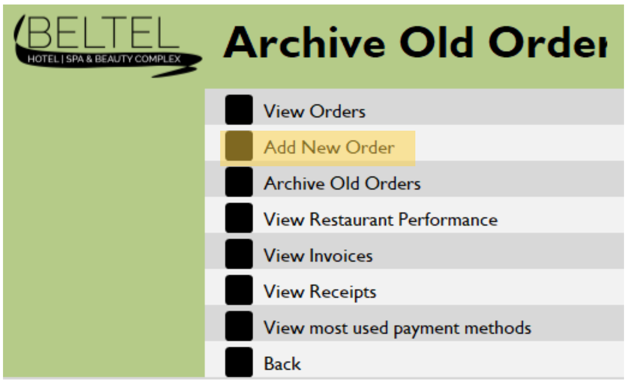


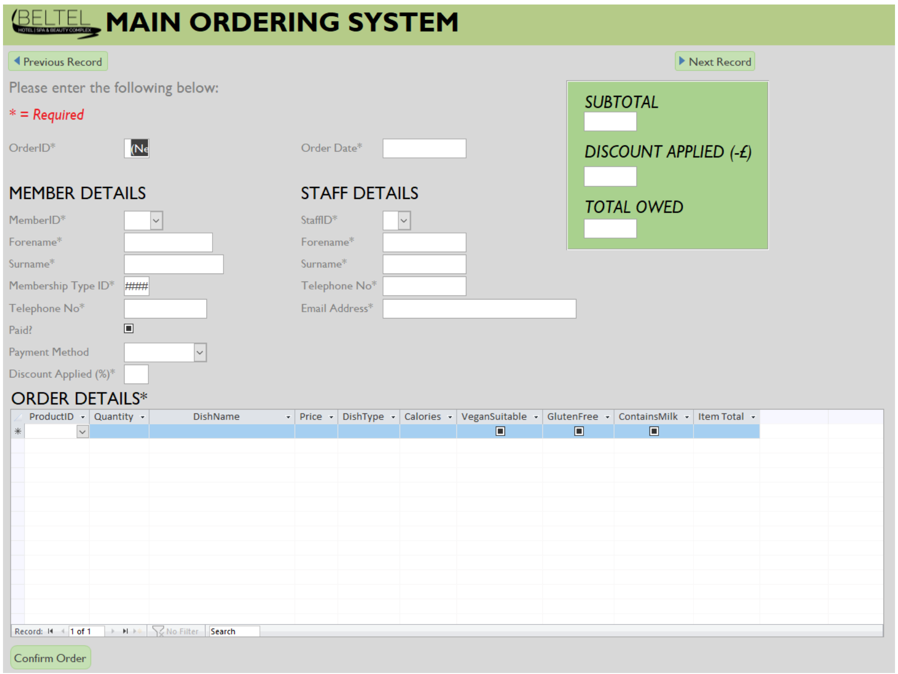
17	Miss	Zoe	Wilkinson	Female	05/08/1993	2	01/01/2014	1 College Square		Bristol	Avon
18		Luke	Stevens	Prefer not to say	24/10/1996	3	20/04/2015	23 Marine Parade		Lyme Regis	Dorset
19	Mr	Tom	Robertson	Male	22/12/1998	3	24/02/2022	17 Norwich Rd		Thetford	Norfolk
20		Isabella	Adams	Non-binary	28/01/2000	4	28/03/2018	11 Preston Rd		Leyland	Lancashire
21	Miss	Jessica	Robertson	Female	23/07/2003	2	01/12/2022	5 Berkeley Hall Ct		Lisburn	Antrim
22	Mrs	Donna	Taylor	Female	09/12/2000	3	02/12/2022	28 Castlecat Road		Bushmills	Antrim
26	Prof	Scott	Reid	Male	12/08/1976	3	16/03/2023	5 N Rd		Newtownards	Down
(New)											

Professor Scott Reid is now stored in the MEMBER table.

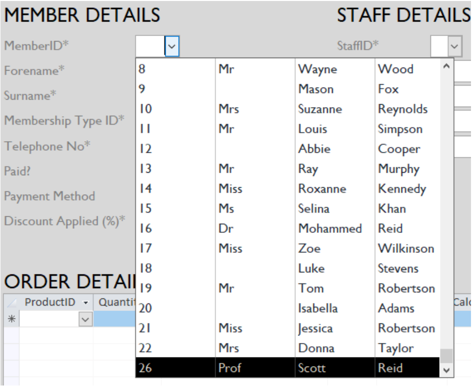
- Return to the menu system – select “Orders” then “Add New Order” which should open the MAIN ORDER form displaying an empty record ready for data input.



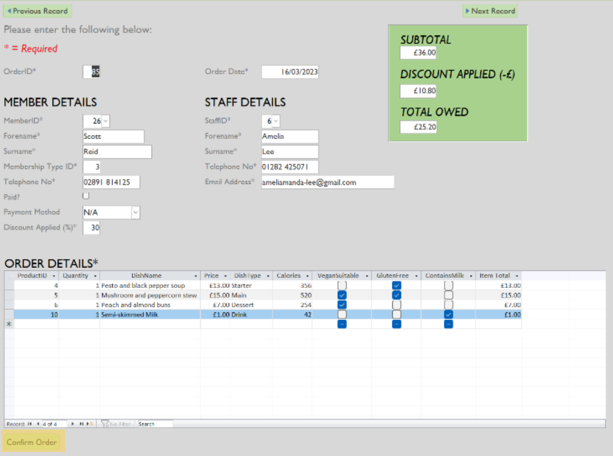




- Professor Reid’s name should appear on the drop-down menu. When selected, his member details will be automatically filled in. The staff member will have to simply input their own details and what Professor Reid has ordered. Calculations will be handled by the system.
  Once finished, select “Confirm Order”. This should store Professor Reid’s order into the ORDER and ORDER_PRODUCT table, and (in this case) the INVOICE query.



8	Mr	Wayne	Wood
9		Mason	Fox
10	Mrs	Suzanne	Reynolds
11	Mr	Louis	Simpson
12		Abbie	Cooper
13	Mr	Ray	Murphy
14	Miss	Roxanne	Kennedy
15	Ms	Selina	Khan
16	Dr	Mohammed	Reid
17	Miss	Zoe	Wilkinson
18		Luke	Stevens
19	Mr	Tom	Robertson
20		Isabella	Adams
21	Miss	Jessica	Robertson
22	Mrs	Donna	Taylor
26	Prof	Scott	Reid



OrderID* 35
Order Date* 16/03/2023

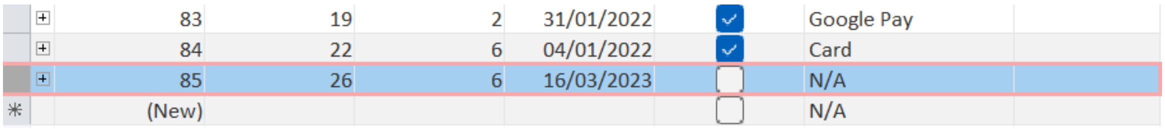
SUBTOTAL £36.00
DISCOUNT APPLIED (-£) £10.80
TOTAL OWED £25.20

MEMBER DETAILS
MemberID* 26
Forename* Scott
Surname* Reid
Membership Type ID* 3
Telephone No* 02891 814125
Paid? ☐
Payment Method N/A
Discount Applied (%)* 30

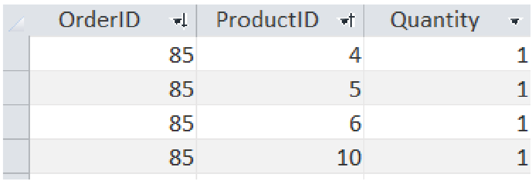
STAFF DETAILS
StaffID* 6
Forename* Amelia
Surname* Lee
Telephone No* 01282 425071
Email Address* ameliamanda-lee@gmail.com

ORDER DETAILS*

ProductID	Quantity	DishName	Price	DishType	Calories	VeganSuitable	GlutenFree	ContainsMilk	Item Total
4	1	Pesto and black pepper soup	£13.00	Starter	356	☐	☑	☐	£13.00
5	1	Mushroom and peppercorn stew	£15.00	Main	520	☑	☑	☐	£15.00
6	1	Peach and almond buns	£7.00	Dessert	254	☑	☐	☐	£7.00
10	1	Semi-skimmed Milk	£1.00	Drink	42	☐	☐	☑	£1.00

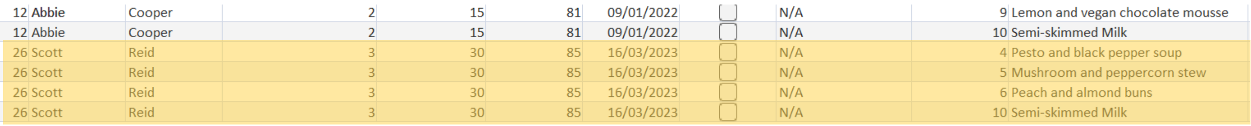


⊞	83	19	2	31/01/2022	☑	Google Pay	
⊞	84	22	6	04/01/2022	☑	Card	
⊞	85	26	6	16/03/2023	☐	N/A	
✳	(New)				☐	N/A	



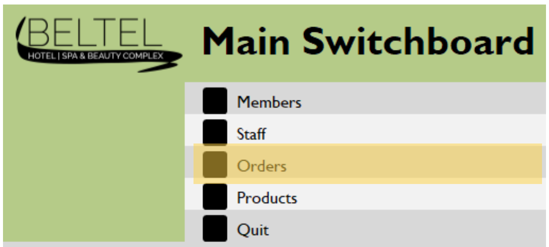
OrderID	ProductID	Quantity
85	4	1
85	5	1
85	6	1
85	10	1

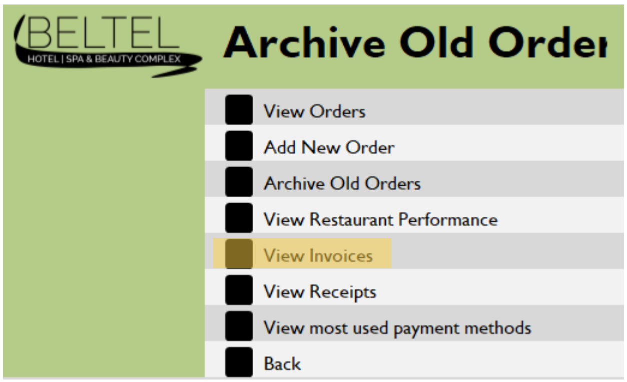
Professor Reid's order is now stored in the ORDER and ORDER_PRODUCT table, and the INVOICE query.

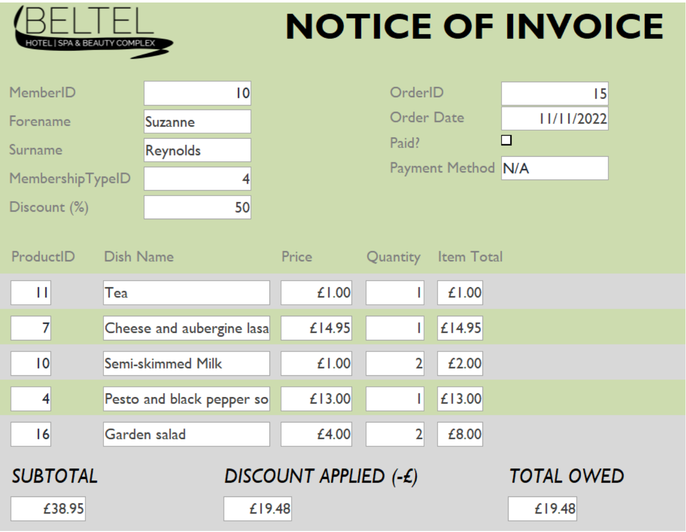


12	Abbie	Cooper	2	15	81	09/01/2022	☐	N/A	9	Lemon and vegan chocolate mousse
12	Abbie	Cooper	2	15	81	09/01/2022	☐	N/A	10	Semi-skimmed Milk
26	Scott	Reid	3	30	85	16/03/2023	☐	N/A	4	Pesto and black pepper soup
26	Scott	Reid	3	30	85	16/03/2023	☐	N/A	5	Mushroom and peppercorn stew
26	Scott	Reid	3	30	85	16/03/2023	☐	N/A	6	Peach and almond buns
26	Scott	Reid	3	30	85	16/03/2023	☐	N/A	10	Semi-skimmed Milk

- Professor Reid hasn't paid for his order; he will need an invoice. As such, return to the menu system - select "Orders" then "View Invoices" which should open the invoice report displaying the member and order details, and how much they owe.







BELTEL
HOTEL | SPA & BEAUTY COMPLEX

NOTICE OF INVOICE

MemberID 10
Forename Suzanne
Surname Reynolds
MembershipTypeID 4
Discount (%) 50

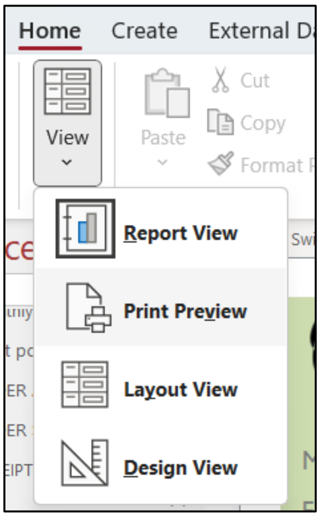
OrderID 15
Order Date 11/11/2022
Paid? ☐
Payment Method N/A

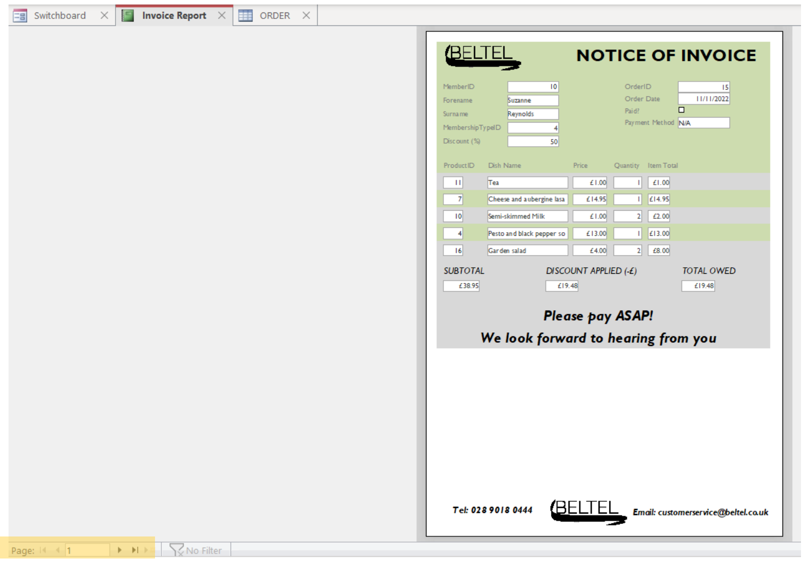
ProductID	Dish Name	Price	Quantity	Item Total
11	Tea	£1.00	1	£1.00
7	Cheese and aubergine lasa	£14.95	1	£14.95
10	Semi-skimmed Milk	£1.00	2	£2.00
4	Pesto and black pepper so	£13.00	1	£13.00
16	Garden salad	£4.00	2	£8.00

SUBTOTAL	DISCOUNT APPLIED (-£)	TOTAL OWED
£38.95	£19.48	£19.48

The invoice for Suzanne Reynolds.

- To view Professor Reid's invoice, select the "Home" tab followed by "View" then "Print Preview". At the bottom of the window is where you can view the different invoices. Since his is the latest invoice, select "Last Page" (the ▶| symbol) and Reid's invoice should appear.





BELTEL NOTICE OF INVOICE

MemberID	10	OrderID	15
Forename	Suzanne	Order Date	11/11/2022
Surname	Reynolds	Paid?	□
MembershipTypeID	4	Payment Method	N/A
Discount (%)	50		

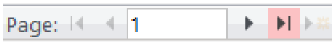
ProductID	Dish Name	Price	Quantity	Item Total
11	Tea	£1.00	1	£1.00
7	Cheese and aubergine lasa	£14.95	1	£14.95
10	Semi-skimmed Milk	£1.00	2	£2.00
4	Pesto and black pepper so	£13.00	1	£13.00
16	Garden salad	£4.00	2	£8.00

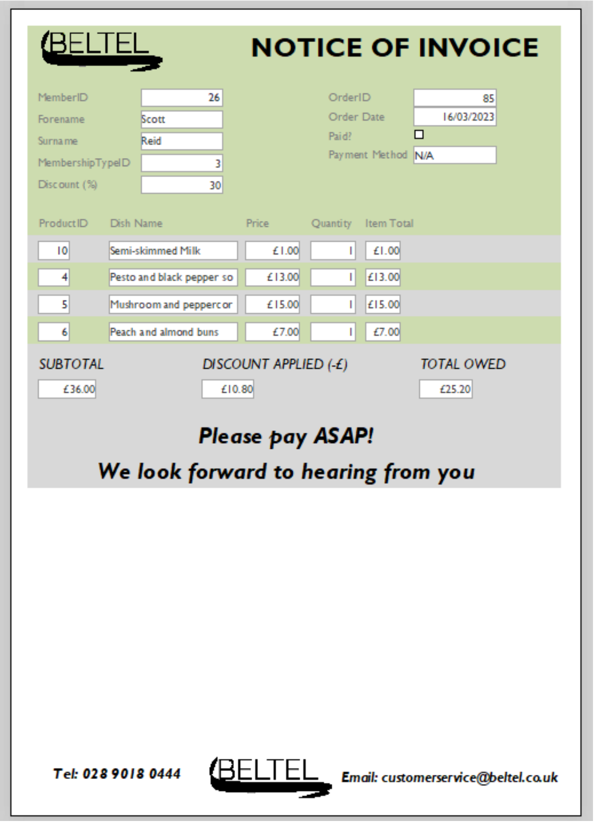
SUBTOTAL	DISCOUNT APPLIED (-£)	TOTAL OWED
£38.95	£19.48	£19.48

Please pay ASAP!

We look forward to hearing from you

Tel: 028 9018 0444 BELTEL Email: customerservice@beltel.co.uk





BELTEL NOTICE OF INVOICE

MemberID	26	OrderID	85
Forename	Scott	Order Date	16/03/2023
Surname	Reid	Paid?	□
MembershipTypeID	3	Payment Method	N/A
Discount (%)	30		

ProductID	Dish Name	Price	Quantity	Item Total
10	Semi-skimmed Milk	£1.00	1	£1.00
4	Pesto and black pepper so	£13.00	1	£13.00
5	Mushroom and peppercor	£15.00	1	£15.00
6	Peach and almond buns	£7.00	1	£7.00

SUBTOTAL	DISCOUNT APPLIED (-£)	TOTAL OWED
£36.00	£10.80	£25.20

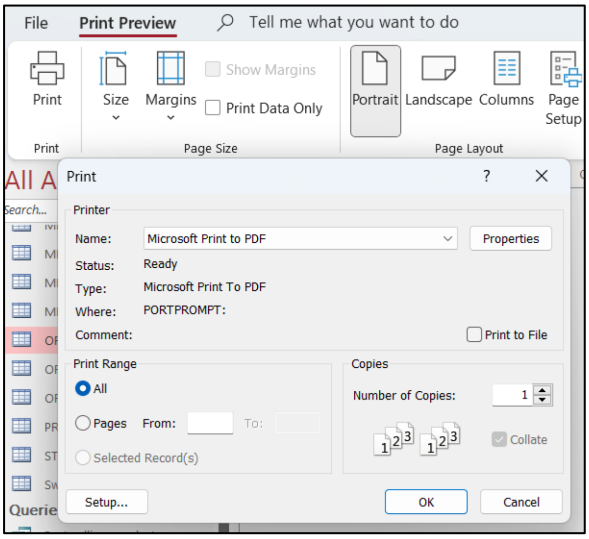
Please pay ASAP!

We look forward to hearing from you

Tel: 028 9018 0444 BELTEL Email: customerservice@beltel.co.uk

Professor Reid's invoice.

- Finally, the staff member can print the invoice by selecting the “Print Preview” tab then “Print”.



Printer

Name: Microsoft Print to PDF

Status: Ready

Type: Microsoft Print To PDF

Where: PORTPROMPT:

Comment:

□ Print to File

Print Range

● All

○ Pages From: To:

○ Selected Record(s)

Copies

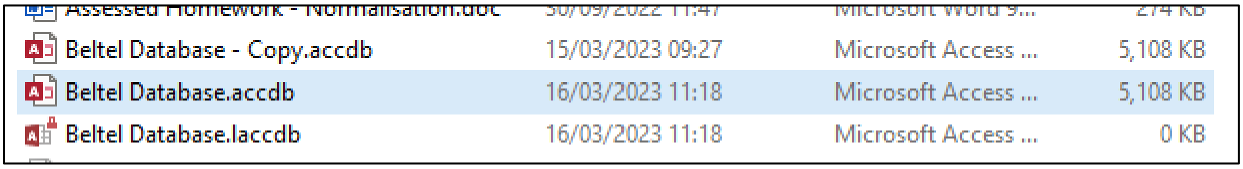
Number of Copies: 1

☑ Collate

2. Archiving orders

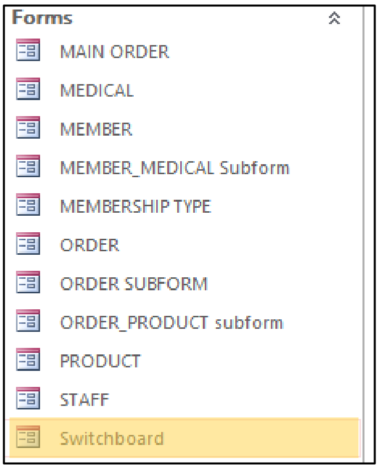
Beltel’s new system has been running without a hitch for a few months. However, management are concerned about the growing backlog of old orders; they fear it is slowing down the system and occupying unnecessary space on their servers. They don’t want them deleted, however, since they still must resolve outstanding invoices. Hence, it is time to archive by doing the following:

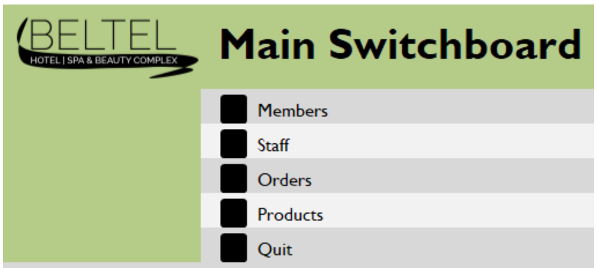
- Open Microsoft Access and select the file “Beltel Database.accdb”



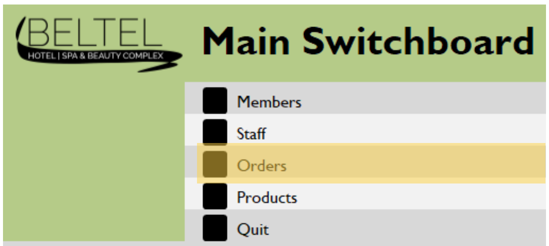
Assessed Homework - Normalisation.doc	30/09/2022 11:47	Microsoft Word 9...	274 KB
Beltel Database - Copy.accdb	15/03/2023 09:27	Microsoft Access ...	5,108 KB
Beltel Database.accdb	16/03/2023 11:18	Microsoft Access ...	5,108 KB
Beltel Database.laccdb	16/03/2023 11:18	Microsoft Access ...	0 KB

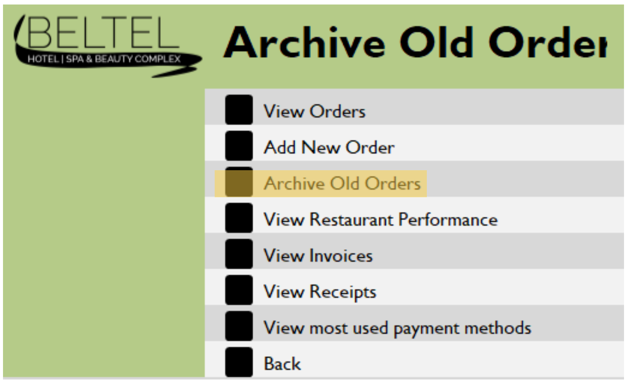
- The database should load into the menu system. If it doesn’t, the staff member can select “Switchboard” under the Forms section.



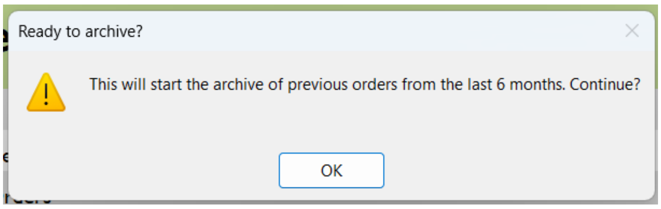


- Navigate through the menu system. First by selecting “Orders”, then in the Order Menu selecting “Archive Old Orders” which should execute the ORDER ARCHIVE MACRO.



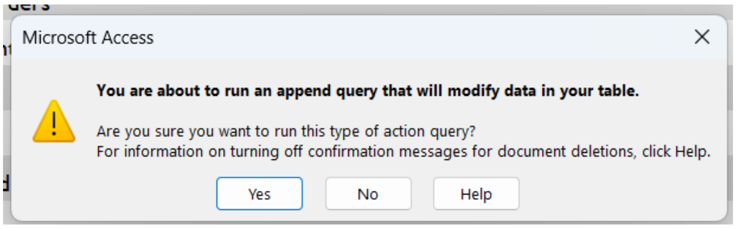


- The staff member will be prompted to confirm the archive process. Select “OK”.


Ready to archive?

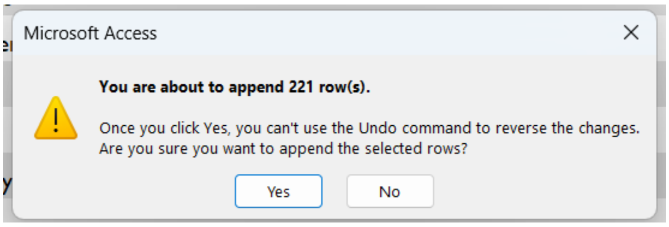
This will start the archive of previous orders from the last 6 months. Continue?

- During this process, the system will append then delete orders older than six months. The staff member will be prompted to confirm these actions. Select “Yes”.


Microsoft Access

You are about to run an append query that will modify data in your table.

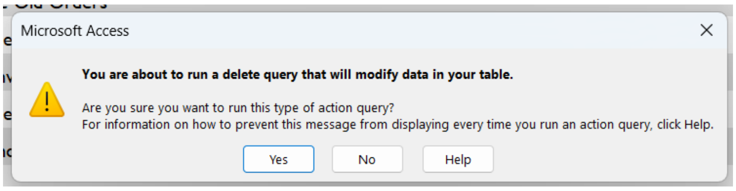
Are you sure you want to run this type of action query?
For information on turning off confirmation messages for document deletions, click Help.


Microsoft Access

You are about to append 221 row(s).

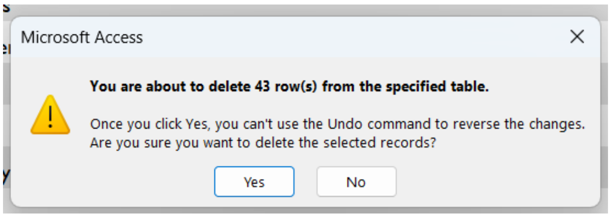
Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to append the selected rows?

ORDER ARCHIVE QUERY


Microsoft Access

You are about to run a delete query that will modify data in your table.

Are you sure you want to run this type of action query?
For information on how to prevent this message from displaying every time you run an action query, click Help.

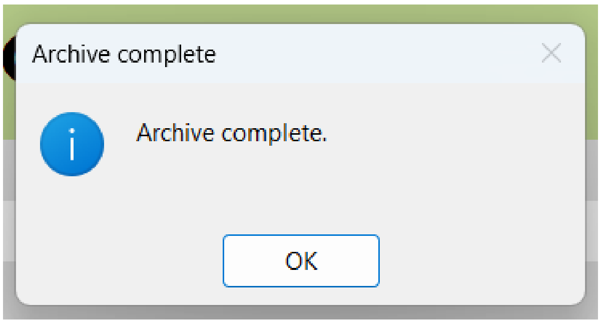

Microsoft Access

You are about to delete 43 row(s) from the specified table.

Once you click Yes, you can't use the Undo command to reverse the changes.
Are you sure you want to delete the selected records?

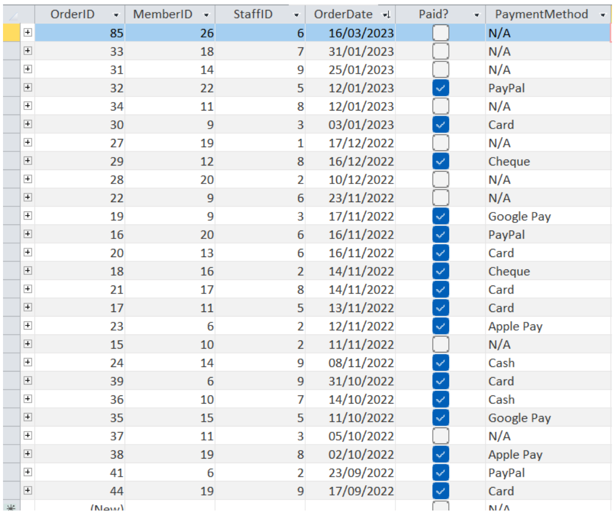
DELETE ORDER query

- When the order archive process is finished, the system will notify the staff member. To acknowledge, select “OK”.



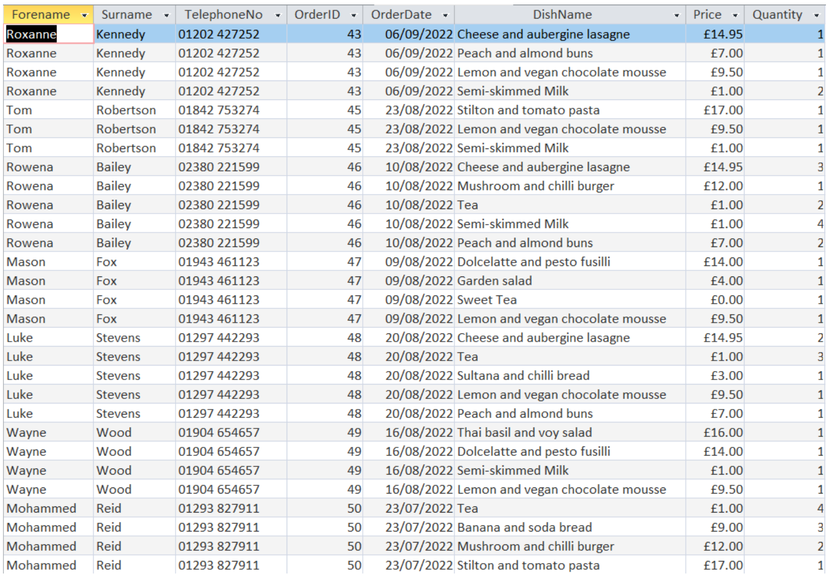
Archive complete

Archive complete.



OrderID	MemberID	StaffID	OrderDate	Paid?	PaymentMethod
85	26	6	16/03/2023	☐	N/A
33	18	7	31/01/2023	☐	N/A
31	14	9	25/01/2023	☐	N/A
32	22	5	12/01/2023	☑	PayPal
34	11	8	12/01/2023	☐	N/A
30	9	3	03/01/2023	☑	Card
27	19	1	17/12/2022	☐	N/A
29	12	8	16/12/2022	☑	Cheque
28	20	2	10/12/2022	☐	N/A
22	9	6	23/11/2022	☐	N/A
19	9	3	17/11/2022	☑	Google Pay
16	20	6	16/11/2022	☑	PayPal
20	13	6	16/11/2022	☑	Card
18	16	2	14/11/2022	☑	Cheque
21	17	8	14/11/2022	☑	Card
17	11	5	13/11/2022	☑	Card
23	6	2	12/11/2022	☑	Apple Pay
15	10	2	11/11/2022	☐	N/A
24	14	9	08/11/2022	☑	Cash
39	6	9	31/10/2022	☑	Card
36	10	7	14/10/2022	☑	Cash
35	15	5	11/10/2022	☑	Google Pay
37	11	3	05/10/2022	☐	N/A
38	19	8	02/10/2022	☑	Apple Pay
41	6	2	23/09/2022	☑	PayPal
44	19	9	17/09/2022	☑	Card
(New)				☐	N/A

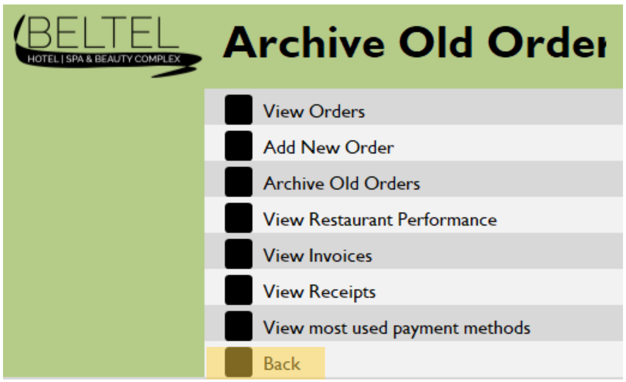
Orders older than six months have been deleted from the ORDER table.

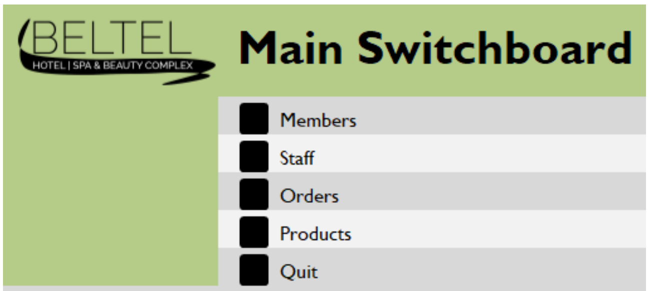


Forename	Surname	TelephoneNo	OrderID	OrderDate	DishName	Price	Quantity
Roxanne	Kennedy	01202 427252	43	06/09/2022	Cheese and aubergine lasagne	£14.95	1
Roxanne	Kennedy	01202 427252	43	06/09/2022	Peach and almond buns	£7.00	1
Roxanne	Kennedy	01202 427252	43	06/09/2022	Lemon and vegan chocolate mousse	£9.50	1
Roxanne	Kennedy	01202 427252	43	06/09/2022	Semi-skimmed Milk	£1.00	2
Tom	Robertson	01842 753274	45	23/08/2022	Stilton and tomato pasta	£17.00	1
Tom	Robertson	01842 753274	45	23/08/2022	Lemon and vegan chocolate mousse	£9.50	1
Tom	Robertson	01842 753274	45	23/08/2022	Semi-skimmed Milk	£1.00	1
Rowena	Bailey	02380 221599	46	10/08/2022	Cheese and aubergine lasagne	£14.95	3
Rowena	Bailey	02380 221599	46	10/08/2022	Mushroom and chilli burger	£12.00	1
Rowena	Bailey	02380 221599	46	10/08/2022	Tea	£1.00	2
Rowena	Bailey	02380 221599	46	10/08/2022	Semi-skimmed Milk	£1.00	4
Rowena	Bailey	02380 221599	46	10/08/2022	Peach and almond buns	£7.00	2
Mason	Fox	01943 461123	47	09/08/2022	Dolcelatte and pesto fusilli	£14.00	1
Mason	Fox	01943 461123	47	09/08/2022	Garden salad	£4.00	1
Mason	Fox	01943 461123	47	09/08/2022	Sweet Tea	£0.00	1
Mason	Fox	01943 461123	47	09/08/2022	Lemon and vegan chocolate mousse	£9.50	1
Luke	Stevens	01297 442293	48	20/08/2022	Cheese and aubergine lasagne	£14.95	2
Luke	Stevens	01297 442293	48	20/08/2022	Tea	£1.00	3
Luke	Stevens	01297 442293	48	20/08/2022	Sultana and chilli bread	£3.00	1
Luke	Stevens	01297 442293	48	20/08/2022	Lemon and vegan chocolate mousse	£9.50	1
Luke	Stevens	01297 442293	48	20/08/2022	Peach and almond buns	£7.00	1
Wayne	Wood	01904 654657	49	16/08/2022	Thai basil and voy salad	£16.00	1
Wayne	Wood	01904 654657	49	16/08/2022	Dolcelatte and pesto fusilli	£14.00	1
Wayne	Wood	01904 654657	49	16/08/2022	Semi-skimmed Milk	£1.00	1
Wayne	Wood	01904 654657	49	16/08/2022	Lemon and vegan chocolate mousse	£9.50	1
Mohammed	Reid	01293 827911	50	23/07/2022	Tea	£1.00	4
Mohammed	Reid	01293 827911	50	23/07/2022	Banana and soda bread	£9.00	3
Mohammed	Reid	01293 827911	50	23/07/2022	Mushroom and chilli burger	£12.00	2
Mohammed	Reid	01293 827911	50	23/07/2022	Stilton and tomato pasta	£17.00	1

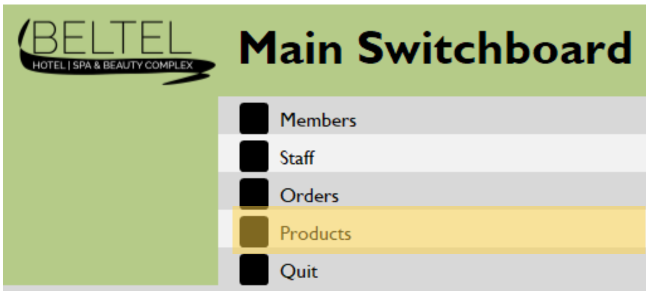
They have been appended to the ORDER ARCHIVE table.

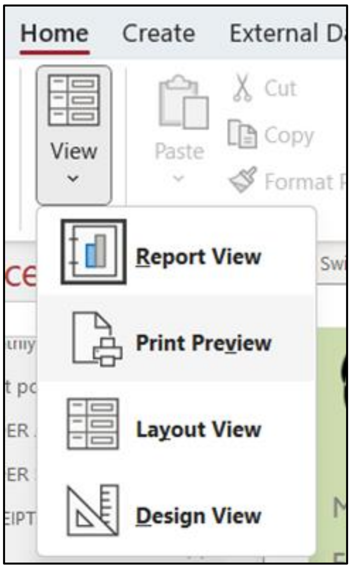
- As a result, some of our reports should now return different results. Going back to the menu system, the staff member should select “Back” which should return them to the Main Menu.

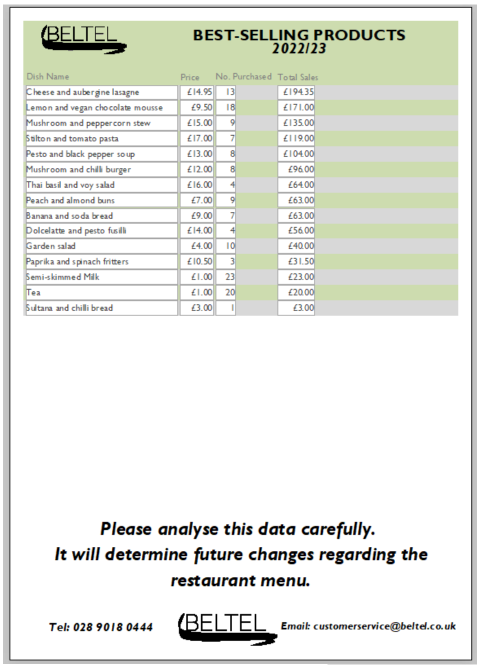




- On the main menu, select “Products” then in the Product Menu select “View best-selling products”. This should open the best-selling products report detailing the total sales for each product. To view the report properly, select the “Home” tab followed by “View” then “Print Preview”.







BELTEL

BEST-SELLING PRODUCTS
2022/23

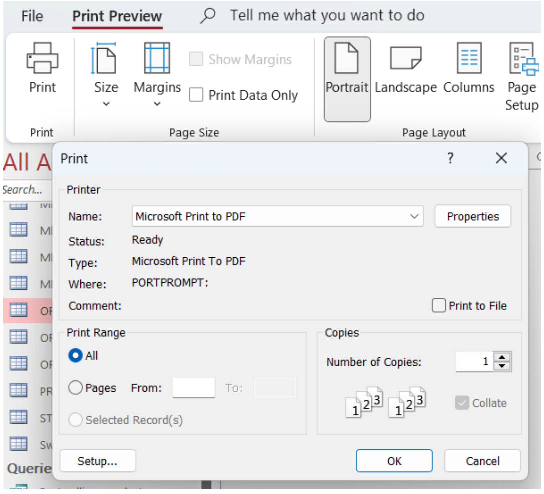
Dish Name	Price	No. Purchased	Total Sales
Cheese and aubergine lasagne	£14.95	13	£194.35
Lemon and vegan chocolate mousse	£9.50	18	£171.00
Mushroom and peppercorn stew	£15.00	9	£135.00
Stilton and tomato pasta	£17.00	7	£119.00
Pesto and black pepper soup	£13.00	8	£104.00
Mushroom and chilli burger	£12.00	8	£96.00
Thai basil and voy salad	£16.00	4	£64.00
Peach and almond buns	£7.00	9	£63.00
Banana and soda bread	£9.00	7	£63.00
Dolcelatte and pesto fusilli	£14.00	4	£56.00
Garden salad	£4.00	10	£40.00
Paprika and spinach fritters	£10.50	3	£31.50
Semi-skimmed Milk	£1.00	23	£23.00
Tea	£1.00	20	£20.00
Sultana and chilli bread	£3.00	1	£3.00

Please analyse this data carefully.
It will determine future changes regarding the restaurant menu.

Tel: 028 9018 0444 BELTEL ***Email: customerservice@beltel.co.uk***

Our new best-selling products

- Finally, the staff member can print the report by selecting the “Print Preview” tab then “Print”.



Print

Printer

Name: Microsoft Print to PDF

Status: Ready

Type: Microsoft Print To PDF

Where: PORTPROMPT:

Comment:

□ Print to File

Print Range

● All

○ Pages From: To:

○ Selected Record(s)

Copies

Number of Copies: 1

☑ Collate

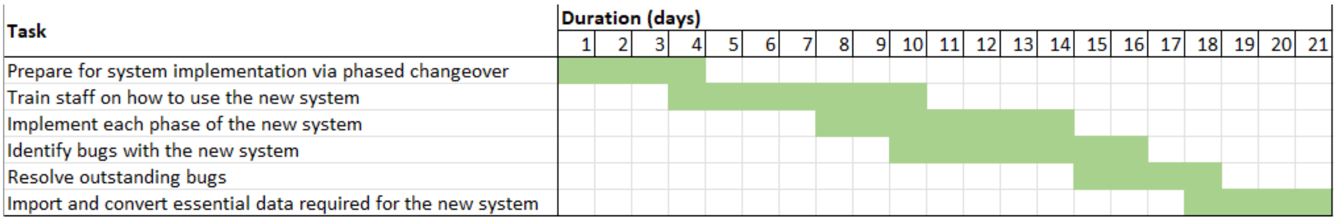
Setup... OK Cancel

Strategy for system implementation

Implementation Plan

Beltel's new system will undertake a phased changeover. The new system is implemented separately at different times. If each phase is successful (with only minor bugs) then the next phase is started. By the final phase, the most serious problems will have been discovered and fixed; the old system is deprecated.

Below is a Gantt chart illustrating how each task will be scheduled during the changeover period:



Task	Duration (days)																				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Prepare for system implementation via phased changeover	🟩	🟩	🟩	🟩																	
Train staff on how to use the new system				🟩	🟩	🟩	🟩	🟩	🟩	🟩											
Implement each phase of the new system								🟩	🟩	🟩	🟩	🟩	🟩	🟩							
Identify bugs with the new system										🟩	🟩	🟩	🟩	🟩	🟩	🟩					
Resolve outstanding bugs															🟩	🟩	🟩	🟩			
Import and convert essential data required for the new system																		🟩	🟩	🟩	🟩

Over a week, the staff of Beltel will be trained on how to use the new system effectively. They must be able to use most if not all aspects of the system, from printing receipts to adding product details. Various training methods are planned to ease the impact of disruption during this important time. Staff will be trained with the user guide and in-house training courses. If one is unable to attend a course, computer-based training is available to teach themselves flexibly.

Furthermore, the developers will remain in contact with Beltel throughout the changeover period. This allows them to provide support in situations where management cannot, such as explaining the order archive process. During days 15 to 18, developers will also be on hand to resolve any outstanding bugs that staff may have encountered when using the new system for the first time.

System changeover

There are various changeover methods available once the new system is functional. Below are evaluations of the following:

Direct Changeover

Beltel will stop using the old system and switch over to the new one the next day; the old system is made redundant immediately.

Advantages

- Only one system is operational at any time. As a result, no time is wasted running the extra system.
- It's also inexpensive if the new system works flawlessly, as Beltel staff will not have to work to manage the two systems simultaneously. All staff can use the one system and as such there is no duplication of effort.

Disadvantages

- Impractical for many organisations. Beltel's operations could be severely disrupted if bugs or other problems arise in the new system. For example, not being able to process new orders otherwise the restaurant would be forced to close until the problem is fixed; this would result in lost profits.

- It may also be impossible to revert to the old system if a problem occurs. Beltel will be forced to record lost data manually or simply live with it. Even if it's small, it will still have recuperations for them.

Parallel Changeover

Beltel's staff will work on the old and new systems simultaneously. As a result, receptionists will duplicate member details on both systems. Once the new system has fulfilled all user requirements, the old system can be made redundant.

Advantages

- The accuracy of the new system can be tested by comparing the output with that of the old system.
- If bugs occur, they can simply rollback onto the old system until they've been resolved.

Disadvantages

- Because of data duplication, extra effort and the resulting strain is put upon staff.
- It can also be costly because extra staff will need to be employed, or existing staff might require working overtime.

Pilot Changeover

A selected group of end users (in this case, management) access the new system first and resolve any errors before making the system available to all users (in this case, staff members).

Advantages

- If management encounters a bug, the changeover can be halted at any time.
- Easy to compare the old and new systems as they're running simultaneously.
- Low risk - if the changeover fails then not too much has been lost.
- Easy to train management by letting them learn new skills during the changeover phase.

Disadvantages

- It can be time consuming to completely replace the old system.
- A pilot changeover may not present immediate problems that a full-scale implementation (e.g., direct) would. Initially, the new system can work well as a small-scale pilot but could suffer difficulties (and disrupt Beltel's operations) when it is scaled up to a full operating system with higher volumes of data to be processed.

Phased Changeover

Occurs in stages whereby Beltel's new system is implemented separately at different times. Data such as order details will be shared between the old and new systems for a time. In this case, Beltel may move half of the customer service department weekly onto the new system, allowing bugs to be resolved at a manageable rate. By the time most departments have been transferred to the new system, the most serious problems will have been discovered and fixed.

Advantages

- Suitable for systems that operate as clearly distinct modules.
- Very structured – each phase can be fully evaluated before moving onto the next one.
- A low risk, well-planned and controlled introduction of the new system.
- Easy to train staff as they can learn new skills on each phase as it is implemented.

Disadvantages

- Slower than direct implementation.
- Although each phase is easy to evaluate, all phases must be complete to evaluate the whole changeover.

What is the most appropriate changeover and why?

Beltel's new system is best suited for a phased changeover – despite the long changeover period and increased implementation cost, the low risk compared to a direct changeover, as well as the fact that staff can be trained as it is implemented helps to make up for time that otherwise would've been lost if using a direct/parallel changeover. There is too much to lose using a direct changeover, the pilot changeover will only present problems after the system is fully implemented and the duplication of data required during a parallel changeover is simply not ideal when processing orders. Rest assured, even if something goes wrong, it won't fully disrupt Beltel's operations.

Training

To perform their jobs capably, the staff of Beltel must receive adequate training on how to use the new system effectively. They must be able to use most if not all aspects of the system, from printing receipts to adding product details. Various training methods are being offered, such as:

User guides

A very useful first line of support, this is provided to all staff who are going to be using the new system. Included are numerous step-by-step instructions for different tasks such as installing the system, adding medical details, archiving orders etc. In addition, during start-up the splash screen will display a range of facts which will notify users of features they may not have used before; this is part of pre-emptive error prevention. The guides will be emailed to staff as a PDF document to be downloaded but booklets are available if they so wish.

Training courses

This involves Beltel management, with the assistance of the development team, providing in-house training for all staff members. This should be started before the changeover process begins or, in the case of phased changeover, before each phase of the new system is implemented. The courses will include classes whereby management teaches their staff on how to utilize everything the system has to offer over a certain period. This can be navigating the switchboard or what not to do e.g., entering data into a query. It also provides staff the opportunity to ask questions about specific features e.g., archiving orders or if they are unsure, they can submit an email to the developers.

This has two main advantages in that these courses exactly match Beltel's requirements, and that staff aren't required to go offsite. However, the expertise needed may not be available internally, thus outsourcing is needed which can increase costs.

Computer based training

Instead of receiving training by management, Beltel staff can be trained using interactive software either stored on an optical disc or web server. This form of training will include multimedia elements such as sound, animation and video, and optional supervision to help the user whenever they're unsure. For example, they will be prompted to watch video tutorials and follow the step-by-step instructions as required. This allows them to rewind and replay material, but there is no opportunity for questions. Furthermore, multiple staff can be trained simultaneously at their own pace, and it may also be cheaper than sending staff on training courses. However, the user may just skip through the tutorials if they aren't forced to repeat the steps shown; this can jeopardise not only the system's effectiveness but also their own performance.

Problems/Maintenance

It is important to prepare for any problems during or after the implementation period. Below are some examples:

- New members might join Beltel during the changeover period. As a result, staff members must record their details manually on the old/new system depending on the implementation stage. If they are yet to be trained, it is recommended that they use the old system for the time being, although that increases the effort required during data conversion.
- Beltel changing their business requirements during/after the implementation period – for example, they may want to expand the ordering system to include spa products. While undoubtedly a headache, the developers will try to apply these additions during the changeover, otherwise they'll have to perform these changes after the new system has been implemented. As a result, time and money are wasted.
- The new system has yet to meet all its user requirements – for example, the calculations for sales are inaccurate, or the order archive process isn't working as intended. Consequently, the new system is simply not ready. Therefore, the phased changeover will have to be paused, Beltel will have to retain the old system for longer, and money will have to be spent resolving any outstanding issues.

Throughout the system's lifetime, maintenance must be performed to ensure that it continues to run efficiently and that it adapts to new requirements. In this case, corrective, adaptive and perfective maintenance will all be executed on the system at one stage and are detailed below:

- **Corrective** – short term maintenance; this corrects bugs or remedies aspects of the system that don't meet the proper requirements. For example, Beltel's ordering system had inaccurate calculations that weren't detected during the changeover period and remain in the system. Over a few weeks, the developers will identify and resolve unmet requirements, bringing the system up to standard.
- **Perfective** – this is applied a few months after system implementation and involves improving the performance of the system by removing inefficiencies. For example, the growing backlog of old orders could slow down the system and store unnecessary space on Beltel's servers. The creation of an order archive macro will optimise the system by appending and removing

orders older than six months. This decreases the hotel’s server storage and maintains the efficiency of the system during its lifecycle.

- **Adaptive –** long term maintenance; this adds new functionality to the system due to changing business requirements or outside factors. Beltel may require this when they decide that the current ordering system (which is developed specifically for the restaurant) no longer fulfils its purpose because they want to branch out by allowing members to order spa products. As such, an expanded and enhanced ordering system must be commissioned to fulfil the restaurant and spa shop, and new reports will have to be created for sales regarding the spa. Furthermore, adaptive maintenance may also be required to take advantage of new hardware (e.g., faster EPOS) or to abide by new legal legalisation (e.g., the storage of member details, what should be stored, and how long for).

Data conversion

This is the final task during the implementation period (days 18 to 21) whereby essential details must be converted from a paper-based format to the new system’s digital format. As this is a phased changeover, however, some orders may have been placed before that aspect was implemented for the new system. This means that member details (especially new members) may’ve been stored on paper to ensure consistency. As a result, the personnel responsible for converting data (in this case, receptionists and ICT support) must key the missing data using the member form provided with the new system.

Likewise, any other essential details stored on paper such as order, product, medical, staff and membership details must be imported into the new system by management using the relevant forms provided.

Section 4 – Evaluation

Aim of the system

This new system was designed to replace the obsolete paper-based system that Beltel had been using. For example, the faulty restaurant POS system meant that staff often resorted to written tickets. This increased the risk of data being lost or misplaced, resulting in inconvenience for not only management but also the member. In addition, there was difficulty keeping track of members' membership plans; someone could stay onsite longer than intended. If a member declared a medical condition after joining the complex, time was wasted searching for their details in the journal.

The new system was intended to fix that. Data is now stored digitally in an organised database, complete with a state-of-the-art ordering system - staff members can select what products have been ordered, and the system will automatically calculate any totals depending on the quantity ordered as well as applying a discount varying on the membership tier. Likewise, membership plans should be easier to track; management should know when someone's overstayed their welcome. Details such as medical conditions can be edited quickly using the relevant forms. Overall, the brand-new menu system should provide an intuitive experience for novice users when navigating the system.

User requirements

Essential requirements

- **Adding/modifying member details** – A member form was created which enabled staff members to process a new member's details using the provided fields. Lookups are utilised when entering a gender or membership type ID. Likewise, the subform utilises a lookup table which they can use to quickly select a list of medical conditions from a drop-down menu. Each field that is required is marked with a * symbol so that they know what field must be entered. It passed the tests needed - the system reliably detected whenever a required field was not entered and displayed an error message. It processed the record in less than 5 seconds, storing it in the dedicated MEMBER and MEMBER_MEDICAL tables. The form was well presented, with navigation buttons to allow staff members to easily view other members' details.
- **Adding/modifying product details** – A product form was created for suppliers to enter their product's details using the provided fields. Range checks are utilised to ensure that a product doesn't contain excessive calories and lookups are utilised to quickly select a list of dish types from a drop-down menu. Each field that is required is marked with a * symbol so that they know what field must be entered. It passed the tests needed - the system reliably detected whenever a required field was not entered and displayed an error message. When a value not part of the list was entered, the system detected an error. It processed the record in less than 5 seconds, storing it in the dedicated PRODUCT table. The form was well presented, with navigation buttons to allow staff members/suppliers to easily view other products.
- **Adding/modifying order details** – A main order form was created for staff members to place an order on the customer's behalf. With lookup tables, they can quickly select members and their own staff details from a drop-down menu. Likewise, the order subform allows staff to easily select what products have been ordered using a drop-down menu. Once chosen, the system automatically enters these details from the relevant tables. Calculations such as subtotal, discount applied and total owed were tested and returned accurate results. In addition, each field that is required is marked with a * symbol so that they know what field

must be entered. When tested, the system reliably detected whenever a required field was not entered and displayed an error message. When a value not part of the list was entered, the system detected an error. It processed the record in less than 5 seconds, storing it in the dedicated ORDER and ORDER_PRODUCT tables. The form was well presented, with navigation buttons to allow staff members to easily view other orders.

  - **A query to view/store main order details** – A main order query was created to function as the control source for the main order form. It passed the tests required – the system returned an accurate list of orders in less than 10 seconds after it was run.
  - **A query for the order subform** – An order subform query was created to function as the control source for the order subform. The system calculated the item total field which was necessary to calculate the subtotal and did so accurately. Likewise, the system returned an accurate list of products purchased for each order in less than 10 seconds after it was run.

- **Adding/modifying medical details** – A medical form was created for staff members to declare a customer's medical condition on their behalf. A length check is utilised to ensure that excessively long names aren't stored, otherwise they'd be difficult to understand. Each field that is required is marked with a * symbol so that they know what field must be entered. It passed the tests needed - the system reliably detected whenever a required field was not entered and displayed an error message. When the maximum number of characters was reached, no more data could be entered. Likewise, it processed the record in less than 5 seconds, storing it in the dedicated MEDICAL table. The form was well presented, with navigation buttons to allow staff members to easily view other medical conditions.
- **Calculating totals** – With assistance from Access' Totals feature, formulas were produced for the system to automatically calculate totals. Using the main order form as an example, the formula for the total owed field required the subtotal to be subtracted from the discount applied. Overall, most of the calculations were accurate, although there was a problem regarding the monthly sales query that was later fixed.
- **Calculating miscellaneous fields** – Unique formulas were produced to retrieve specific data. Using the main order form as an example, the formula for the subtotal field required the retrieval of the sum of the item total field from the order subform. It passed all testing - all types of these calculations returned by the system were accurate.
- **Adding/modifying membership details** – A membership type form was created for management to store different membership plans. A length check is utilised to ensure that excessively long names aren't stored, otherwise they'd hoard unnecessary space on Beltel's servers. Range checks are also used for prices and discounts to prevent radical changes without consent. Each field that is required is marked with a * symbol so that they know what field must be entered. It passed the tests needed - the system reliably detected whenever a required field was not entered and displayed an error message. When the maximum number of characters was reached, no more data could be entered. When management keyed in data beyond the limit, the system detected the error. It processed the record in less than 5 seconds, storing it in the dedicated MEMBERSHIP TYPE table. The form was well presented, with navigation buttons to allow management to easily view other membership tiers.
- **Adding/modifying staff details** – A staff form was created for management to capture a staff member's details whenever they've been employed using the provided fields. A format check is utilised when entering a member's DOB; the system is formatted to accept DD/MM/YYYY. A lookup is also used which allows them to quickly select a list of pay grade codes from a drop-

down menu. Each field that is required is marked with a * symbol so that they know what field must be entered. It passed the tests needed - the system reliably detected whenever a required field was not entered and displayed an error message. If a date was entered incorrectly, the system applied autocorrect or rejected it. When a value not part of the list was entered, the system detected an error. It processed the record in less than 5 seconds, storing it in the dedicated STAFF table. The form was well presented, with navigation buttons to allow management to easily view other members' details.

- **Producing a receipt/invoice** – A copy of every order placed was produced as a report which staff would print and send to the member. It passed the tests needed – it retrieved the correct data sources and presented all the essential information. The report was professional and intuitive, following the brand's colour scheme with an easy-to-read layout. Each order was grouped accurately and calculations such as subtotal, discount applied and total owed were tested and returned accurate results. It also managed to be produced in less than 10 seconds.
- **Creating queries (searching) for invoices/receipts** – An invoice and receipt query were created to function as the control sources for the invoice and receipt reports. Both utilised criteria to return only paid/unpaid orders and a calculation to produce the item total. It passed the tests required – the system returned an accurate list of paid/unpaid orders with reliable totals in less than 10 seconds after it was run.
- **Creating a query (searching) for members with medical conditions** – A members with medical conditions query was created to function as the control source for its corresponding report. It was sorted in ascending alphabetical order and used the DateDiff formula to calculate the member's age. It passed the tests required – the system returned an accurate list of members suffering from medical conditions with ages reliably calculated in less than 10 seconds after it was run.
- **Creating a query for monthly sales** – A monthly sales query was created to function as the control source for its corresponding report. Numerous calculations were applied such as taking the sum of every order in January to give the monthly total and using the MonthName formula to convert the month number of each order to its named equivalent. Unfortunately, the calculations were returning erroneous totals because it was retrieving data from the wrong source. The query had to be redesigned and now displays an accurately calculated list of profits from each month of the year; it does so in less than 10 seconds after it was run. Users can easily filter by a particular year using the parameter displayed once the query is run.
    - **In addition, a query must be added that applies the discounts for all orders which then must be grouped and linked to the query above.** – A monthly sales discount query was created and linked to provide the monthly total and monthly discount fields. Calculations were applied such as taking the sum of each item total to give the order total and applying the discount by dividing the order total by 100 before multiplying by the discount amount. It passed the tests required – in less than 10 seconds, the system returned a list of orders with accurately calculated discounts and the system linked the two queries.
- **Creating a query (searching) for the best-selling products** – A best-selling products query was created to function as the control source for its corresponding report. Calculations were applied such as summing the quantity to provide the number ordered for each product. It also utilised criteria to omit free products and was sorted in descending order so that the highest selling product appeared first. It passed the tests required – the system returned an accurate

list of the highest selling products at the restaurant with quantity reliably calculated in less than 10 seconds after it was run.

- **Creating a query (searching) for the most popular payment method** – A most popular payment methods query was created to function as the control source for its corresponding report. The different payment methods were totalled to provide the frequency of each using the COUNT function; it was sorted in descending order so that the most frequent method appeared first. Criteria was also used to omit any "N/A" orders. It passed the tests required – the system returned a sorted, accurate list of payment methods in less than 10 seconds after it was run.

Non-essential requirements

- **Monthly sales report system** – A copy of the monthly sales query was produced as a report which the owner/senior manager would print and send to management. Unfortunately, it failed some tests – while it retrieved the correct data source, it produced erroneous profits because of the faults within that query. It also didn't present all the essential information as the grand total calculation failed to be displayed due to a formula error. These problems were rectified when the query and calculation were redesigned. Fortunately, the report was professional and intuitive, following the brand's colour scheme with an easy-to-read layout. Each year was grouped and presented accurately subject to what the user keyed into the parameter, and calculations such as total sales (after discounts) were tested and returned accurate results. It also managed to be produced in less than 10 seconds.
  However, that is not to say that this requirement was fully met – it was planned that the grand total calculate the total sales for each year. This was found to be more complex to achieve in Access than previously thought and, due to time constraints, a compromise was needed. As such, a much simpler calculation was implemented that totals the sales in a lifetime.
- **Best-selling products report system** – A copy of the best-selling products query was produced as a report which the owner/senior manager would print and send to management. It passed the tests needed – it retrieved the correct data source and presented all the essential information. The report was professional and intuitive, following the brand's colour scheme with an easy-to-read layout. It was automatically sorted by total sales in descending order and omitted free products (since they will not make a profit). It also managed to be produced in less than 10 seconds.
- **Members with medical conditions report** – A copy of the members with medical conditions query was produced as a report which the owner/senior manager would print and send to management. Unfortunately, it failed some tests – the report failed to present all the essential information because it was grouped by MedicalID rather than MemberID and the member details were incorrectly stored in the page header. Once it was fixed, it met its requirements - it retrieved the correct data source and now presented all the essential information. The report was professional and intuitive, following the brand's colour scheme with an easy-to-read layout. Each member was grouped accurately, and it managed to be produced in less than 10 seconds.
- **Developing a switchboard user interface** – With assistance from the Switchboard Manager, a menu system complete with hyperlinked buttons and submenus were created to aid novice users when navigating the system. Each button has been labelled clearly describing to the user what they do e.g., "View Member Details" - this opens the MEMBER form in edit mode. It passed the tests required – each button was tested/referenced and were correctly linked to their intended destinations. It was professional, following the brand's colour scheme with an

easy-to-read layout. Most importantly, it was quick, with load times being less than 4 seconds. Initially, the switchboard failed to be displayed on startup but that was a simple case of enabling the Display Form feature in the program options.

- **Archiving orders** – An order archive macro was created so that users can append and delete orders older than 6 months to improve system efficiency and decrease storage space. Two queries are utilised to perform the actions required as well as message boxes to notify the user of its status. It passed the tests required – the macro processed the data in less than 5 seconds, it performed each action in order and the message boxes appeared as expected. Users can easily perform this action by selecting "Archive Old Orders" in the Order Menu of the switchboard.
    - **Two queries are required for this macro** – An order archive and delete order query were created to function as action queries for the order archive macro. Both utilised criteria to only append/delete orders older than 6 months. It passed the tests required – when the macro was executed, the system appended an accurate list of old orders and deleted them from their old tables. It did so in less than 5 seconds after it was run. Message boxes asking for confirmation also appeared as expected.
- **Preventing members with allergies from ordering flagged products** – This was to be implemented as part of the ordering system – a staff member would receive an error message no later than 3 seconds after a member with a milk allergy attempted to order a product containing milk. Unfortunately, this requirement has not yet been fulfilled because this is an extremely complicated feature to implement; a lot of time would have to be invested, which wasn't feasible given the 60-hour time frame.
- **Show a splash screen on start-up** – This was to be implemented during the startup sequence. The Beltel logo would be presented prominently and follow the brand's colour scheme. Unfortunately, this requirement has not been fulfilled because I lack the experience of implementing a splash screen using Access. The fact that it was non-essential meant it was also a lower priority to apply than other features; it just wasn't necessary.

Overall, the development and implementation of the system has been successful. 22/25 user requirements have been met – all essential and 5 non-essential. If there was more time and experience, it could've been possible to fix/implement those unmet requirements. Nevertheless, at least the system performs the essential functions.

System strengths

- **Main order form** – Complex yet simple, this form is the backbone of the ordering system and thus was essential to make as user friendly as possible. This was achieved by combining drop-down menus and autofill; even the most novice of staff could choose from a list of members or products and the system would handle the rest. Likewise, the system's automatic calculations prevent potential errors from staff performing their own. It is also beneficial to service – if a member is served, for example, 10 minutes after placing an order, they are likely to leave a positive review either via the internet or word of mouth. As a result, this will undoubtedly increase profits and improve the hotel's reputation.
- **Switchboard/Menu system** – This is necessary for users to navigate the system efficiently. It will be the first thing they see when opening the database and was designed with intuitiveness

in mind. Each button has been labelled clearly describing to the user what they do e.g., “View Receipts” - this opens the receipt report. It was professional, following the brand’s colour scheme with an easy-to-read layout; the various aspects of the system have been divided into submenus and each button was presented as a list. The fact that each button is linked reduces the risk of errors (unless that link is changed in the future). It was efficient, as load times took less than 4 seconds.

- **Monthly sales report** – This was intended to be printed and sent to management. As such, it needed to be presented professionally. In that regard, it was successful, following the brand’s colour scheme with an easy-to-read layout. The report was grouped accurately which enabled management to evaluate the monthly profits for each year separately. Likewise, the system’s automatic calculations prevent potential errors from management performing their own. Despite the compromise regarding the grand total, it is still useful to management when evaluating the restaurant’s performance as it provides knowledge on how much profit it has generated since the new system was implemented.
- **Order archive macro** – This was aimed to free up storage space and improve system efficiency. In that regard, it works as intended – the system executes the action queries required and did so accordingly by the criteria provided. Despite the complexity of this process, users can easily perform this action by selecting “Archive Old Orders” in the Order Menu of the switchboard. The message boxes displayed throughout the process notify the user of its status and if any errors have been detected. Most importantly, it processed the data in each stage no more than 5 seconds.

Limitations

- **Grand total calculation in monthly sales report** – As stated previously, the calculation for the grand total was planned to be more sophisticated than the final release. Management could evaluate the total sales generated by the restaurant yearly rather than in a lifetime. After some research, it was found to be simpler than earlier thought. This can be implemented in a later patch by moving the calculations required for the grand total (the sum of the monthly total and sum of the monthly discount) from the report header to the Order Year header. The report will now display the total sales for each year as intended.
- **Missing splash screen before displaying the switchboard** – First impressions matter. Research found that there is a common misconception among users that Access is not a “real” database application. This presents a problem; consequently, they may refuse to use Beltel’s system. To fix this delusion, a splash screen will be implemented in a later patch to omit Access/Microsoft branding – a copy of the Beltel logo will be converted to a bitmap (.bmp) image and renamed to match the name of the database. This should display Beltel’s logo instead of Access while the system starts up.

Improvements

- **Expanding the ordering system to include spa products** – Business requirements are forever changing; one day, Beltel will decide that the current ordering system (which is developed specifically for the restaurant) no longer fulfils its purpose because they want to branch out by allowing members to order spa products. As such, new tables (to store spa products), reports (to print sales regarding the spa) and forms (for ordering said products) must be developed all while the current ordering system is not disrupted. In addition, the main order form must be redesigned to include a spa order subform and the system must be adapted to manage new calculations. This could take a while; time and money will need to be invested into this development. Nevertheless, it will improve day-to-day operations of Beltel by increasing efficiency.
- **Stock management system** – The current system lacks any kind of stock system. This presents a problem; only the chefs and management can track stock levels from time-to-time by surveying the freezer/refrigerator. It is then likely that members will be inconvenienced if the restaurant is out of a product they wanted to order. A stock system can be developed to prevent this – when a product is low on stock, the system will detect this, and reorder said product. The product table will be redesigned to accommodate stock levels. The main order form must also be remodelled to update stock levels each time an order is placed. Most beneficially, everybody can track the stock levels in real-time via a report or form.
- **Website frontend for registering new members** – Rather than a staff member inputting member details over the phone, new members can be directed to Beltel's website where they can key information about themselves online. This is beneficial for various purposes – it reduces the effort required for staff regarding members; they can focus their workload elsewhere. It also reduces the risk of data inconsistency when keying in data (e.g., Sean being spelt as Shaun). Once submitted, member details are automatically stored in the database which staff members can view later.
- **Feedback system** – Members can leave feedback directly to Beltel using a dedicated form whereby they rate various aspects of their stay on a scale of 1 to 10. Since their reviews will not be published online (without consent), it provides them the opportunity to be honest. Once submitted, each review is automatically stored in the database which enables management to review the hotel's reputation overtime either via a report or form; they don't have to waste time searching the internet and comparing reviews on different sites. With permission, they can also use select reviews as part of advertising.