

## CG4

# Analysis



### **Background**

Meta Games is a small-sized commercial business founded by James Newton, which sells video games. The business/shop is located in Cardiff, Vale of Glamorgan, Queen Street (a busy shopping road with lots of customers coming and going). The store provides customers with a wide range of games, both new releases and old, as well as a large selection of pre-owned games for a bargain price. Though the store competes with games stores such as Game or Cex, they

stand out in that they games generally at a slightly cheaper price than those stores. And Meta Games prides itself on its collection of older, more retro games that you probably would not find in stores such as Game and Cex. Meta Games has a very dedicated customer circle. The buyers are often local buyers that frequently visit the store. Therefore there is a paper-based database for customers because of those regular loyal customers, this helps benefit both parties because customers are able to go to the store and order a game, which will then be sent to their home.

As with many other shops and game stores, it is vital that they can store data on all their customers, as well as the games which are in stock and the orders placed for said games. As games are continually being sold stocks are depleted, which requires the business to frequently order new copies of games so as to adequately meet demands. However, this makes the task of keeping track of all orders and which orders are delivered more difficult due to the rate of which games are sold.

I have chosen Meta Games because there is a lot of room for improvement by implementing an automated system which can keep track of all sales, customer records, order records, games and stock. I also feel as though they would really benefit by computerisation and the ability to perform many tasks effortlessly.

## **Investigation and Analysis of the Current System**

I have performed a thorough investigation of the business by inspecting the company's records as well as talking to both James Newton and the staff concerning their means of data input, output and processing.

The business is run via a paper based system, storing information on:

- Customers details - Contact numbers, Contact's first name, Contact's surname, addresses, postcodes, email addresses, etc.
- Work Records - Date and times of sales, cost of games ordered, employee wages, etc.
- Customer payments - A record of all customer purchases

The advantage of the current paper based system is that it allows the store to very quickly keep records on pen and paper, without the reliance on technology as well as learning how to use electronic methods.

However, the problem is that there is no backup of records in case of accidents. Large amounts of paper are used in the process of making work records and it can be very difficult to locate specific information very quickly. It can also be difficult to update information, e.g. a new customer mobile number. If the shop converted to a digital, computer based system then it would allow them to keep

backups of his sales records, customer details and purchases on external storage devices such as an external hard drive, in case of accidents.

The business requires a computer based database system which will allow for quick access of a large amount of data. They want to be able to access specific information for specific purposes via the use of queries, for instance; 'which customer has spent the most at the store this month/year?' These questions are important to both the customer and the store because the business regularly likes to reward their most loyal customers with discounts.

## Interviews

I conducted an interview with the owner, James Newton, and asked a series of questions to help me gauge what the business use their system for and what the new system will need to incorporate.

1. What is your current system and what do you use?

"We tend to write down our work records as we go through the day. As an order is made we will write it down as new ones come in. Whenever a customer registers with us then we will write their information out there and then. All of this is done via paper."

2. What do you think of your current management system of records?

"It effectively does get the job done. However it definitely isn't the best system. Papers go missing, writing records takes time and storing the records can take up space which we'd otherwise be using for other purposes."

3. Do you often lose records?

"It isn't too common, though I would say at least once every month we will have a situation in which we simply cannot find an order record or customer record. I imagine it's normally a case of someone just thinking it's a useless piece of paper and then throwing it in the bin. It does suck though because when we write up our monthly reports the missing records give us an incomplete overview of the month because we are missing one of those records."

4. Why hasn't MetaGames updated to a computer based system yet?

"When we started the company it was an issue of finance. We simply couldn't afford a developer to create a nice database system for us. Whenever you start up a business it is always tight to begin with. But after the first year it really became habit. We got used to using paper for our records."

5. Do you think a computerised system would help the company or hinder it in any way?

"I think a computerised system would be much, much better. I think we've all had enough of the paper based system, to be honest. As the business

grows and we keep getting more orders, more customers, more games and more supplier contacts, it is becoming unrealistic and unmanageable to do this by paper. It is taking up too much time, and we can't really afford to lose records. I don't really think a computerised system will hinder us very much. However, we aren't all very experienced with computers, so some of us may have difficulty getting the hang of the computerised system."

6. What kind of features would you find useful for the new system to have?

"Being able to search for specific information would be helpful. Going through files can be a pain when you're trying to find one specific customer record."

## Questionnaires

Answer the following questions using a scale of 1-10, with 10 being the best and 1 being the worst.

- A. How easy is your current system to use?
- B. How reliable is your current system?
- C. How functional and practical is your current system?
- D. How important is an easy to use system for you?
- E. How important is functionality and practicality of the system to you?
- F. How eager are you on getting a new system?

| Question              | A | B | C | D | E | F |
|-----------------------|---|---|---|---|---|---|
| <b>Average Answer</b> | 7 | 8 | 6 | 9 | 9 | 8 |

## Problem Definition

The business wants to use the database for storing information on customers (names, contact numbers, addresses, postcodes, date of births), orders (date, time, cost, total cost) and games (titles, genre and cost of games) and suppliers (names, contacts, addresses, contact numbers, country, region, postcode, etc.)

It is important that the database is capable of using queries, so that the company can very quickly find specific data. For example, finding a specific customer by searching their ID number or finding all orders of games that have the RPG genre. These queries will allow managers to quickly find the information

they're looking for and can aid in their decision making. They've also requested that the database is able to perform specific calculations based upon game orders, such as the total cost of a customer's order (which will be dependent upon the number of games the customer buys and their prices).

A feature of the database which the business has stressed is that of reports. They want to be able to create a report which can calculate the total income over the month. This reports will help the company track their expansion by following their monthly incomes over the year. It will also make data comparisons much easier so the company can gauge their progress. If the average monthly income a year ago was higher than it is for the current year then it may mean that the company needs to invest more into advertisement.

The company has requested that their house style and logo be displayed on the background of their database system and that the overall feel of the system is professional, slick and smooth, whilst also being easy to read and use. The whole database will following the house-style, for example, buttons will be silver with a dark red text, headings will be in red and the store's logo will be displayed throughout all the forms.

## **Objectives**

I am going to create a database system for the store that will:

- Provide the company with an easy to use, unitive, menu driven interface so that it will not require any training to take advantage of.
- Store customer information, including: title, first name, surname, address line 1, address line 2, city, postcode, contact number and email address.
- Store game information, including: title, genre and cost.
- Store order information, including: date, time, delivery date, cost, VAT cost and total cost.
- Store supplier information, including: company name, contact name, company address, city, region, postcode, country and phone number.
- Have a login system to ensure that all customer information is secured.
- Have an 'add new order' form which is capable of calculating VAT cost of order and the total order cost (including the VAT).
- Have an 'add new game' form which contains a sub-form of the order form.
- Display a splash screen window before taking the user to the login form.
- Have the ability to edit/delete customer records.
- Have the ability to edit/delete game records.
- Have the ability to edit/delete supplier records.
- Have the ability to edit/delete order records.
- Provide users with the ability to ask the database queries and present that data in a simple, palatable manner.
- Produce a report that will display the company's total income for the month.
- Have the ability to printout reports with the click of a button.
- And lastly present a system that follows the house style of the company.



**CG4**

**Design**

**META  GAMES**

## **Output content and Format**

There will be a number of different data outputs within the database system, including data outputted via forms and tables. However, the primary source of output data in the system will be that of the reports. The reports are specifically for providing output data, so these will be the main source of data output.

Add New Order Form: This form is going to allow users to add a new form to the record (Order Table). By filling the order form out and selecting okay the data will be transferred and made into a record on the order table. There is a Total Cost textbox and a VAT Cost textbox in this form. Once the cost has been input into the Cost value has been input into the Cost textbox, this will trigger an automatic calculation of not only the total cost, but it will also display the VAT cost to the user. The reason for this output of data is to show the user what the total cost of their order is (which is of course essential) and it is both useful and interesting to see the VAT cost itself displayed as then you get a rough idea of how much of your purchase was spent on VAT alone. This form will also have another output and that will be via the implementation of the sub-form at the bottom of the form. This sub-form will be the link table 'Game Orders'. The table will display window ID, the cost of the game and the quantity of those games in stock. This output is useful as it allows the user to see relevant information (such as the quantity and game cost/ID) without having to switch between forms and tables.

\*All reports will be structured into columns and rows and feature a print button at the lower right hand corner and feature the current date at the top of the report, and will be sorted alphabetically as well as grouped by ID (numerically grouped).

Order Report: The data outputs that will be in the order report are as follows: total income for the month in orders, number of orders this month, order ID, order date, order total cost, month and customer ID. These outputs will be important because it is essential that a company can track their financial profit, and the best way to keep track of that is by knowing your total income over a given period, e.g. a month, a year, etc. By having this monthly total in the report it will allow the company to easily keep track of their earnings. Having the number of orders this month allows the business to gauge how busy their orders are, which is important for gauging the success and progression of the company. I have included a calculation which produces the total income for the month. This easy method of comparison allows managers to really get a feel for the progression of the company and whether they are losing business or gaining business. The average daily subtotal can be useful when budgeting and expanding a company as you can make accurate predictions about your income so when it comes to making a possibly risky or tight purchase you can roughly predict how many days it will take to make that money back. The rest of the fields will be used as output data for the purpose of keeping hardcopy records.



## Input Content, Capture and Format

The inputs in my database system are the areas of the project where users are required to enter data into the database. The majority of this input data will be stored within the database records (tables), however some will also be used in a function to produce an output, such as a query.

### Login Form:

Here the user will input the username and password, if the username and password are recognised then the user will gain access to the database, if not a message box prompt will appear saying "Login unsuccessful". I implemented the login system as a means of protecting customer's personal details and securing the system as a whole.

The logo for MetaGames, featuring the word "META" in black, a stylized red and grey "G", and the word "GAMES" in black.

Username:

Password:

Login

## Add New Customer Form:

Here the user is able to input data about the customer. By selecting add new customer (provided the data has been input) it will update the customer table and add that new customer to the list of customers. The user will find this form by selecting customer menu from the main menu, then from here they can simply enter the details, add the member and get back to their business. A very simple form, navigated by a very simple, menu-driven interface. The customer ID will not be a user input as this is an automatic value. Selecting 'return to main menu' will take the user back to the main menu, 'add new customer' will add the record to the table and 'return to customer menu' will take the user back to the customer menu (the previous form).

# METAGAMES

## Add New Customer

|                        |                      |  |
|------------------------|----------------------|--|
| Customer ID            | <input type="text"/> | <input type="button" value="Return to main menu"/>     |
| Customer title         | <input type="text"/> |  |
| Customer first name    | <input type="text"/> | <input type="button" value="Add new customer"/>        |
| Customer surname       | <input type="text"/> |  |
| Customer address       | <input type="text"/> | <input type="button" value="Return to customer menu"/> |
| Customer address       | <input type="text"/> |  |
| Customer city          | <input type="text"/> |  |
| Customer postcode      | <input type="text"/> |  |
| Customer contact       | <input type="text"/> | number   |
| Customer date of birth | <input type="text"/> |  |

## Add New Game Form:

This is the form where the user will inputting data to add a new video game to the database. The user will input the new data into all the fields and the data will be sent to the table 'Game' to be stored as a record. Game ID does not have to be entered by the user as it is an automatic value. Selecting 'return to main menu' will take the user back to the main menu, 'add new game' will add the record to the table and 'return to game menu' will take the user back to the game menu (the previous form).

# METAGAMES

**Add New Game**

|               |                      |                            |
|---------------|----------------------|----------------------------|
| Game ID       | <input type="text"/> | <b>Return to main menu</b> |
| Game title    | <input type="text"/> |                            |
| Game genre    | <input type="text"/> | <b>Add new game</b>        |
| Game cost     | <input type="text"/> |                            |
| Game quantity | <input type="text"/> | <b>Return to game menu</b> |
| Supplier ID   | <input type="text"/> |                            |

## Add New Order Form:

# Design

This is the form in which users will be entering new order information. By filling out the required fields and selecting 'add new order' the order table will be updated with a new order record. Selecting 'return to main menu' will take the user back to the main menu, 'add new order' will add the record to the table and 'return to order menu' will take the user back to the game menu (the previous form). The order ID does not have to be entered because it is an automatic value, however the order VAT cost and the total cost are both also automatic once the order cost has been entered. This was achieved by programming the calculations in visual basic. There is also a sub-form in the bottom of the form which is the link table (Game Order).

## META GAMES

**Add New Order**

|                     |                      |
|---------------------|----------------------|
| Order ID            | <input type="text"/> |
| Order date          | <input type="text"/> |
| Order time          | <input type="text"/> |
| Order delivery date | <input type="text"/> |
| Order cost          | <input type="text"/> |
| Order VAT cost      | <input type="text"/> |
| Total cost          | <input type="text"/> |
| Customer ID         | <input type="text"/> |

Return to main  
menu

Add new order

Return to order  
menu

| Game Order ID | Game ID | Cost | Quantity |
|---------------|---------|------|----------|
| 1             | 3       | £12  | 3        |
| 2             | 2       | £25  | 5        |

Add New Supplier Form:

# Design

This is the form that the user will be using to add records of new suppliers. When data is entered into the fields and add new supplier is selected the records will be sent to the table and the new supplier will appear in the table records.

Selecting 'return to main menu' will take the user back to the main menu, 'add new supplier' will add the record to the table and 'return to supplier menu' will take the user back to the supplier menu (the previous form). The supplier ID will not be filled in by the user as it is an automatic field.

# META GAMES

## Add New Supplier

Supplier ID

Company name

Contact name

Company address

City

Region

Postcode

Country

Company phone

number

Return to main menu

Add new supplier

Return to supplier menu

View/Edit Customer Form:

The user will use this form when they want to make changes in the records of their customers. Here they will be able to input a whole set of new fields under the same customer if they have to. Selecting 'return to main menu' will take the

user back to the main menu, 'return to customer menu' will take the user back to the customer menu, 'save changes' will update that customers records with whatever fields were updated with new data, 'delete' will delete the whole record set so that customer will no longer be in the database.

**META GAMES**

**View/Edit Customer**

Customer ID

Customer title

Customer first name

Customer surname

Customer address

Customer address

Customer city

Customer postcode

Customer contact

Customer date of

Save

Delete

line 2

number

birth

Return to main menu

Return to customer menu

### View/Edit Game Form:

Here the user will be able to view and edit a game record. A practical use of this form would be reducing the price of a video as the price decreases. However, the user has the ability to update/alter all fields, not just cost. Selecting 'return to main menu' will take the user back to the main menu, 'return to game menu' will

take the user back to the game menu, 'save changes' will update that game record with whatever fields were updated with new data, 'delete' will delete the whole game record set so that game record will no longer be in the database.

**META GAMES**

**View/Edit Game**

Game ID

Game title

Game genre

Game cost

Game quantity

Supplier ID

Save

Delete

Return to main menu

Return to game menu

### Add New Order Form:

In this form the user will be able to update a current order record. A practical application of this would be if a customer rung the company requesting that they change the delivery day of their order from the 05/05/2016 to the 06/05/2016. All fields will be updatable. Selecting 'return to main menu' will take the user back to the main menu, 'return to order menu' will take the user back to the order menu, 'save changes' will update that order record with whatever fields were

updated with new data, 'delete' will delete the whole order record set so that order record will no longer be in the database.

**META GAMES**

**View/Edit Order**  
--

|                     |                      |               |
|---------------------|----------------------|---------------|
| Order ID            | <input type="text"/> | <b>Save</b>   |
| Order date          | <input type="text"/> |               |
| Order time          | <input type="text"/> |               |
| Order delivery date | <input type="text"/> | <b>Delete</b> |
| Order cost          | <input type="text"/> |               |
| Order VAT cost      | <input type="text"/> |               |
| Total cost          | <input type="text"/> |               |
| Customer ID         | <input type="text"/> |               |

**Return to main menu****Return to order menu**

### View/Edit Supplier Form:

Here the user will be able to view and edit a supplier record. A practical use of this form would be updating a supplier's contact name, address, phone number, etc, in the event of a change of employee or location. The user has the ability to



# Design

update/alter all fields in the record. Selecting 'return to main menu' will take the user back to the main menu, 'return to supplier menu' will take the user back to the supplier menu, 'save changes' will update that supplier record with whatever fields were updated with new data, 'delete' will delete the whole record set so that supplier record will no longer be in the database.

META**G**AMES

View/Edit Supplier

Supplier ID

Company Name

Contact Name

Company Address

City

Region

Postcode

Country

Company Phone

Save

Delete

Number

Return to main menu

Return to order menu

## **Files and/or data structures, methods of access**

File structure: The programme will be stored as a .dba file, and it will be accessed in Microsoft Access. This is because access is easy to use and any problems can easily be fixed by a developer if necessary. The user interface in access is easy to use, familiar and not over complicated. The built in facilities on Microsoft Access creates for a useful environment in which users can take advantage of numerous features which aid in the creation of a database. The ability to create command buttons, text boxes, labels, tick boxes, etc, with complete ease will be very helpful for development.

## **Data Dictionary**

### **Customer**

| <b>Field Name</b>       | <b>Data Type</b> | <b>Validation/Input Mask</b>                 | <b>Description</b>   |
|-------------------------|------------------|--|--|
| Customer ID             | Autonumber       | No duplication                               | Automatically produces a number which will uniquely identify the customer. |
| Customer Title          | Text             | Drop down list box:<br>Mr, Mrs, Miss, Ms, Dr | Allows the user to select their title.                                     |
| Customer First Name     | Text             | Maximum length - 20<br>Presence check        | Allows the user to input customer's first name.                            |
| Customer Surname        | Text             | Maximum length - 20<br>Presence check        | Allows the user to input customer's surname.                               |
| Customer Address Line 1 | Text             | Maximum length - 60<br>Presence check        | Allows the user to input the first line of customer's address.             |
| Customer Address Line 2 | Text             | Maximum length - 50<br>Presence check        | Allows the user to input the second line of customer's address.            |
| Customer City           | Text             | Maximum length - 20<br>Presence check        | Allows the user to input customer's city.                                  |

|                         |           |   |   |
|-------------------------|-----------|---|---|
| Customer Postcode       | Text      | Maximum length - 10<br>Presence check<br>Format Check     | Allows the user to input customer's postcode.       |
| Customer Contact Number | Text      | Maximum length - 11<br>Presence check<br>Length check     | Allows the user to input customer's contact number. |
| Customer Date of Birth  | Date/Time | Presence check<br>Format check<br>Input mask (short date) | Allows the user to input customer's date of birth.  |

## Game Order - (Game Link)

| Field Name    | Data Type  | Validation/Input Mask                 | Description   |
|---------------|------------|---------------------------------------|---|
| Game Order ID | Autonumber | No duplication                        | Automatically produces a number which will uniquely identify the order details. |
| Order ID      | Number     | Foreign Key<br>Presence check         | Automatically produces a number which will uniquely identify the order.         |
| Game ID       | Number     | Foreign Key<br>Presence check         | Automatically produces a number which will uniquely identify the game.          |
| Quantity      | Text       | Maximum length - 20<br>Presence check | Allows the user to input customer's chosen quantity of game.                    |

## Game

| Field Name | Data Type  | Validation/Input Mask                 | Description  |
|------------|------------|---------------------------------------|--|
| Game ID    | Autonumber | No duplication                        | Automatically produces a number which will uniquely identify the game. |
| Game Title | Text       | Maximum length - 70<br>Presence check | Allows the user to input the title of the game.                        |
| Game Genre | Text       | Maximum length - 30<br>Presence check | Allows the user to input the genre of the game.                        |

# Design

|               |          |                               |  |
|---------------|----------|-------------------------------|--|
| Game Cost     | Currency | Presence check                | Allows the user to input the cost of the game.   |
| Game Quantity | Text     | Presence check                | Allows the user to input how many of each game are currently in stock.                 |
| Supplier ID   | Number   | Foreign Key<br>Presence Check | Automatically produces a number which will uniquely identify the supplier.             |
| Stock         | Number   | Presence check                | Allows user to input the number of copies of a given game that are currently in stock. |

## Order

| Field Name          | Data Type  | Validation/Input Mask  | Description  |
|---------------------|------------|--|--|
| Order ID            | Autonumber | No duplicates  | Automatically produces a number which will uniquely identify the order.                |
| Order Date          | Date/Time  | Short date<br>Presence check                                     | Allows the user to input the date of order.  |
| Order Time          | Date/Time  | Short date<br>Short time format and input mask<br>Presence check | Allows the user to input the time of order.  |
| Order Delivery Date | Date/Time  | Short date<br>Presence check                                     | Allows the user to input the delivery date of order.                                   |
| Order Cost          | Currency   | Presence check   | Automatically produces the cost of the customer's order.                               |
| Order VAT Cost      | Currency   | Presence check   | Automatically produces the cost of the VAT for the customer (VAT=20%)                  |
| Order Total Cost    | Currency   | Presence check   | Automatically adds the Order Cost and the Order VAT Cost together to produce the Total |

|             |        |                               |   |
|-------------|--------|-------------------------------|---|
|             |        |                               | Cost of the customer's order.   |
| Customer ID | Number | Foreign key<br>Presence check | Automatically places the Customer ID taken from the customer record – links this table with the customer table. |
| Month       | Text   | Presence check                | Allows user to input the month of the order.  |

## Supplier

| Field Name      | Data Type  | Validation/Input Mask                 | Description  |
|-----------------|------------|---------------------------------------|--|
| Supplier ID     | Autonumber | No duplicates                         | Automatically produces a number which will uniquely identify the supplier. |
| Company Name    | Text       | Maximum length – 70<br>Presence check | Allows the user to input the company's name                                |
| Contact Name    | Text       | Maximum length – 70<br>Presence check | Allows the user to input the company's contact name.                       |
| Company Address | Text       | Maximum length – 70<br>Presence check | Allows the user to input the company's address.                            |
| City            | Text       | Maximum length – 50<br>Presence check | Allows the user to input the city which the company/supplier resides in.   |
| Region          | Text       | Maximum length- 80<br>Presence Check  | Allows the user to input the region which the company/supplier resides in. |
| Postcode        | Text       | Maximum length – 10<br>Presence check | Allows the user to input the company's postcode.                           |

|                      |      |                                       |   |
|----------------------|------|---------------------------------------|---|
| Country              | Text | Maximum length - 30<br>Presence check | Allows the user to input the country which the company/supplier resides in.                         |
| Company Phone Number | Text | Maximum length - 11<br>Presence Check | Allows the user to input the contact phone number of the company/supplier.                          |
| Game ID              | Text | Maximum Length - 40<br>Presence Check | Automatically places the Game ID taken from the Game record - links this table with the Game table. |

## **Validation**

Data validation is necessary so that the information the user enters actually makes sense, is true, relevant and not excessive. Validation simply is a method of ensuring that data is more truthful by eliminating errors that may be made (for instance adding an extra digit in a phone number).

### **Technique:**

Presence Check - This validation check will be used for all fields to check if data has been entered into the relevant and required fields which the user must fill out. I will be using this technique because it will ensure that data is entered into relevant fields, in an effort to ensure data is sensible and reasonable.

### **Technique:**

Format Check - This validation check will be used in the customer table and the order table for the contact number fields. It will require that the number is entered in the correct format: digits. This will be formatted as '#'. I will be using this technique to ensure that the required field's data is entered in the correct format, in an effort to ensure data is sensible and reasonable. If the user enters data which is not in the correct format, for example, a letter, a validation text box will appear that says "Incorrect format".

### **Technique:**

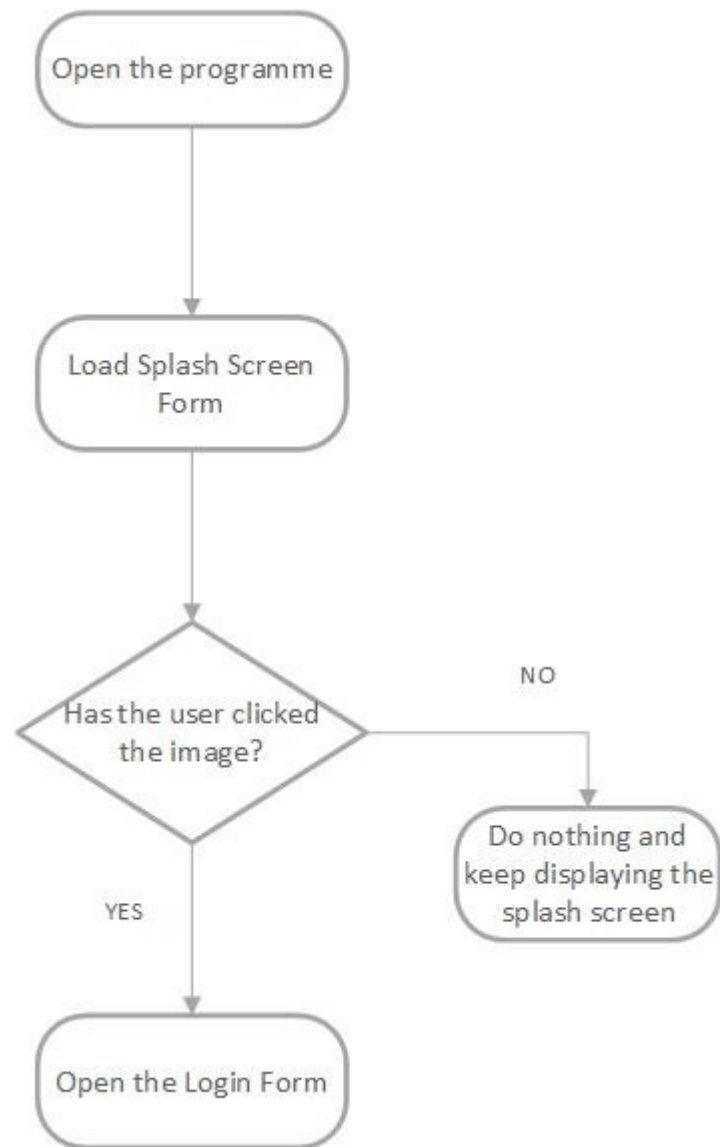
Range Check - This validation check will be used in the order table under the game quantity field. It will ensure that an unrealistic number of games are not entered into the field. E.g. a game with a quantity of 100 will not be accepted and a message box will pop up saying 'Invalid quantity'.

This will ensure that the quantity of games entered is sensible, reasonable and not excessive. Many of the games also do not have 100 copies in stock

## **Processing Stages**

### Flowcharts

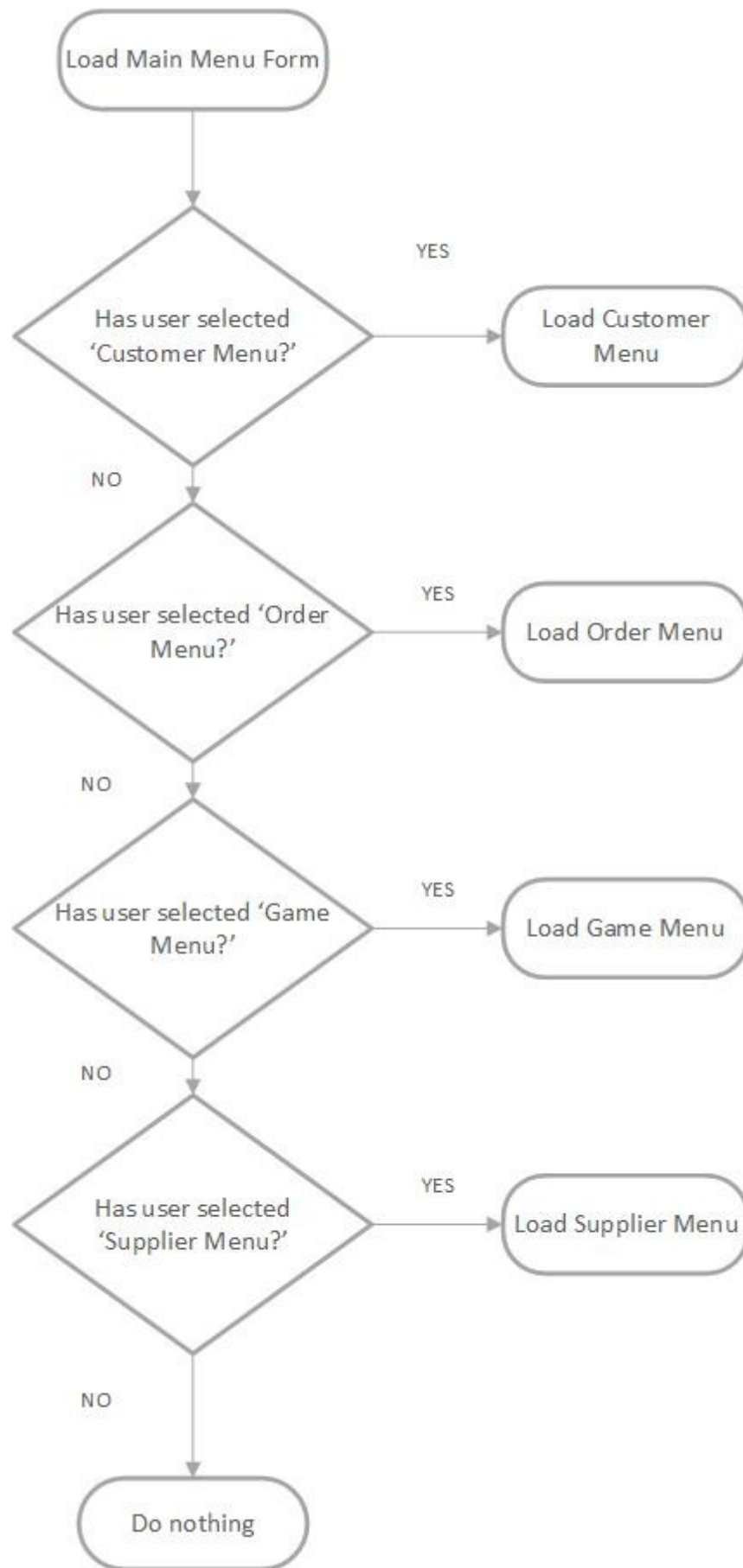
1. Splash screen:



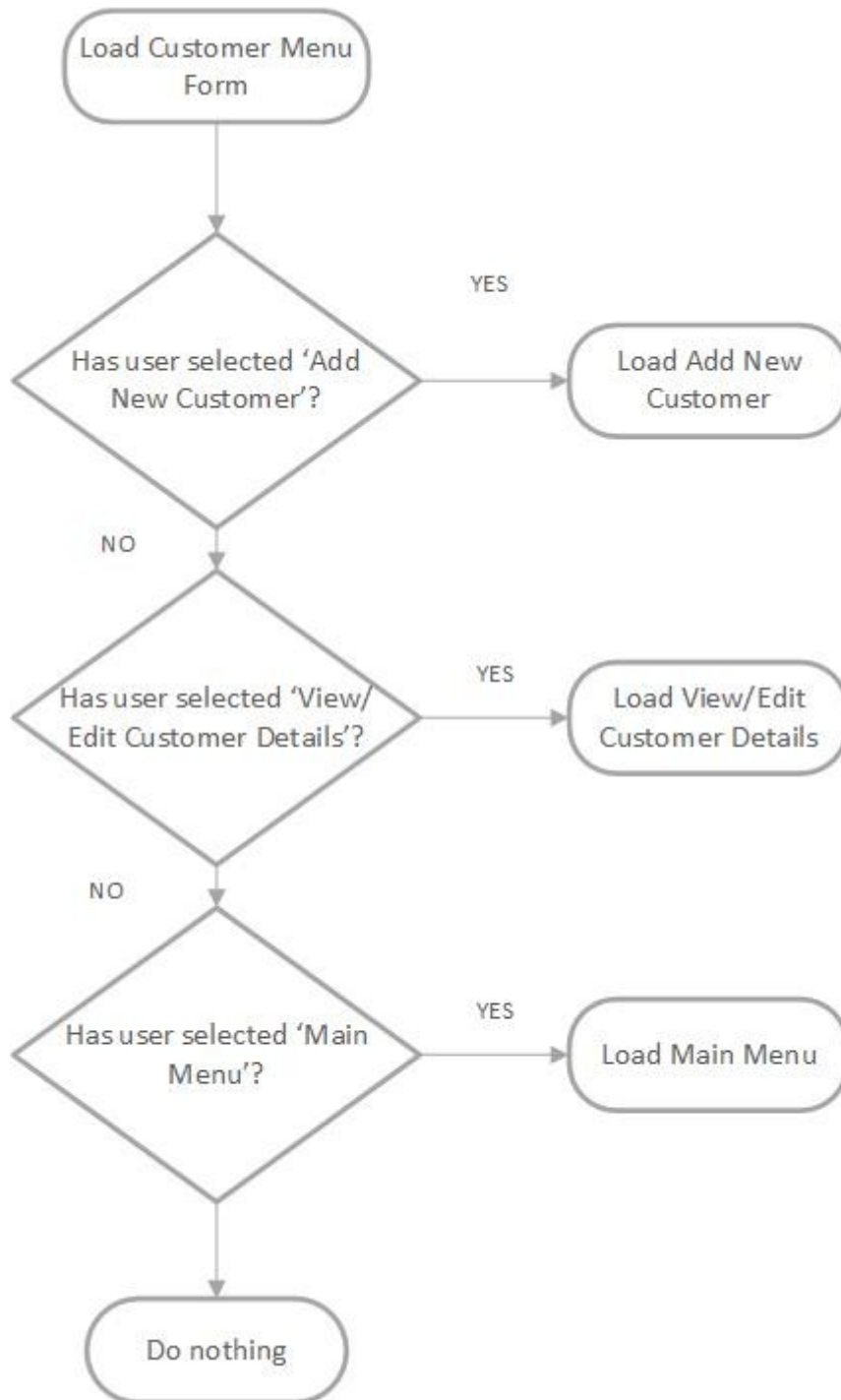
2. Login Form

3. Main Menu form

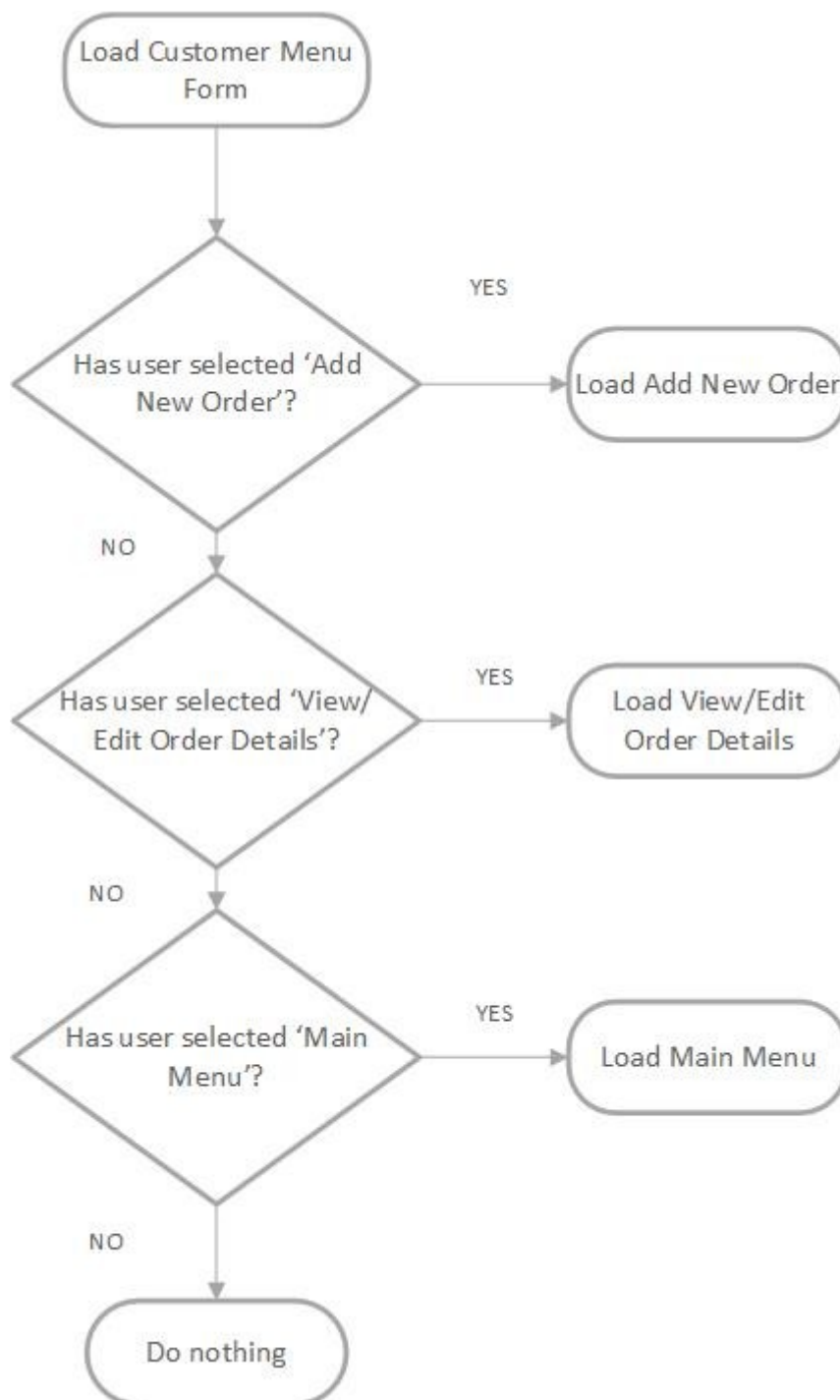




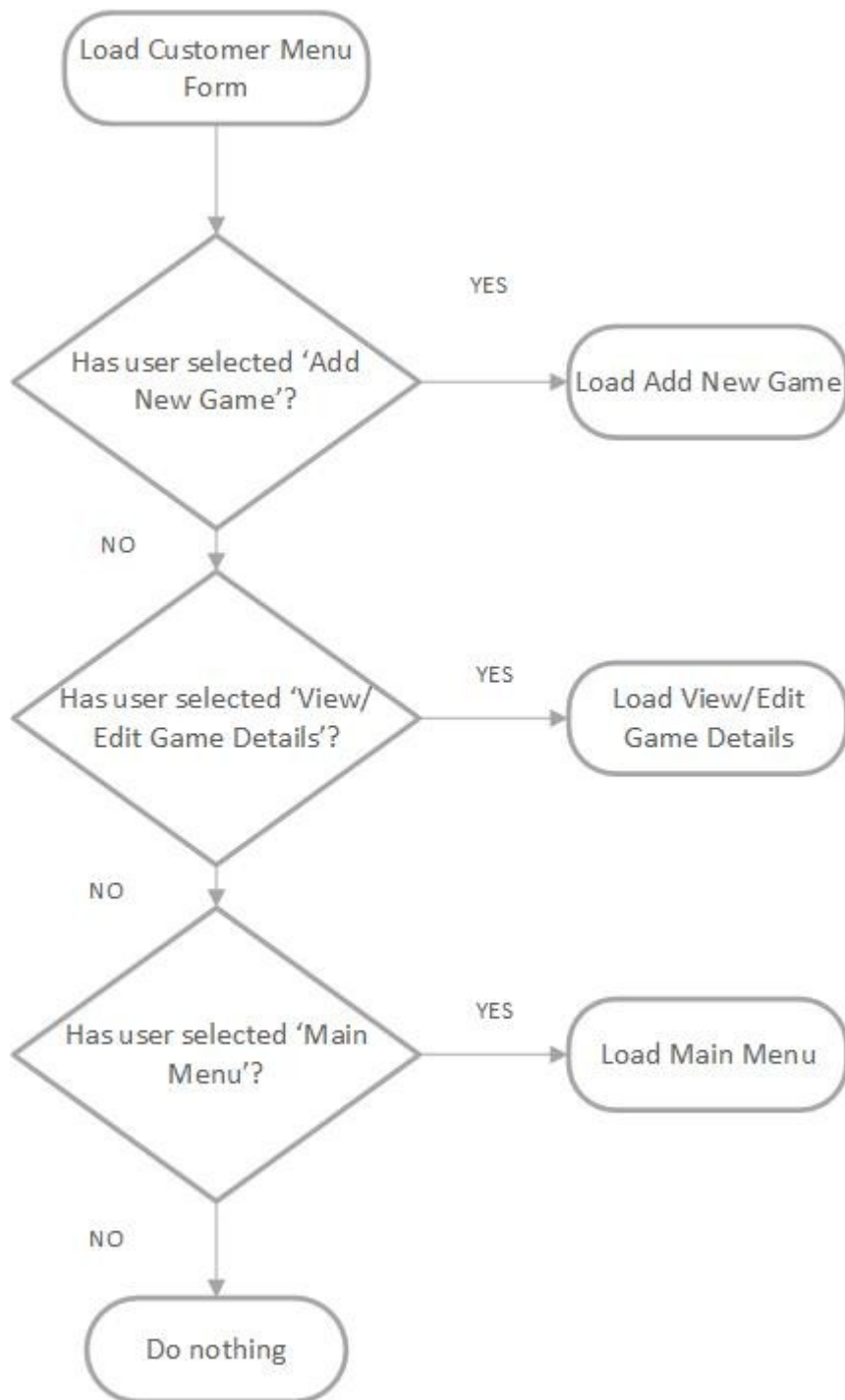
4. Customer Menu



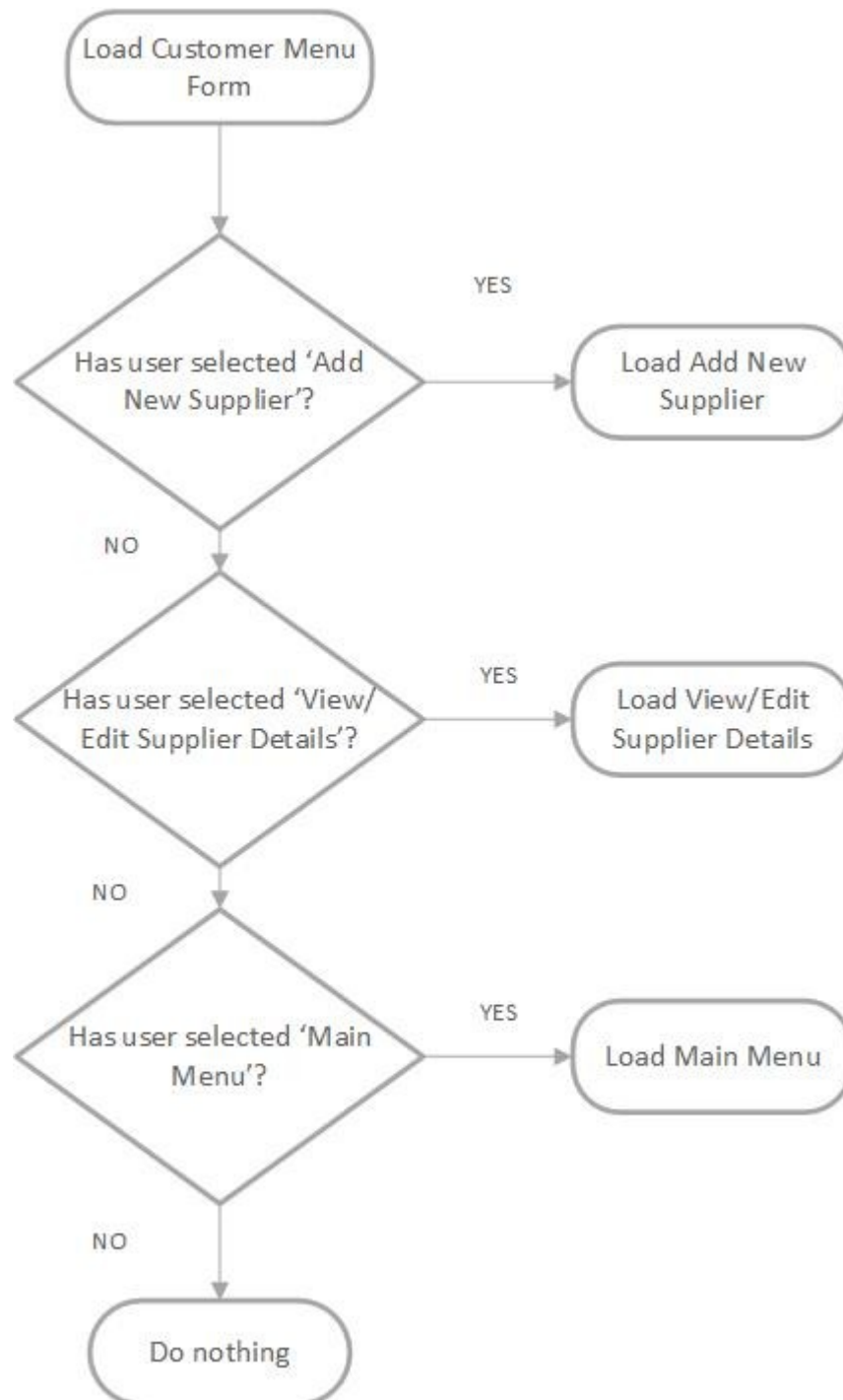
## 5. Order Menu



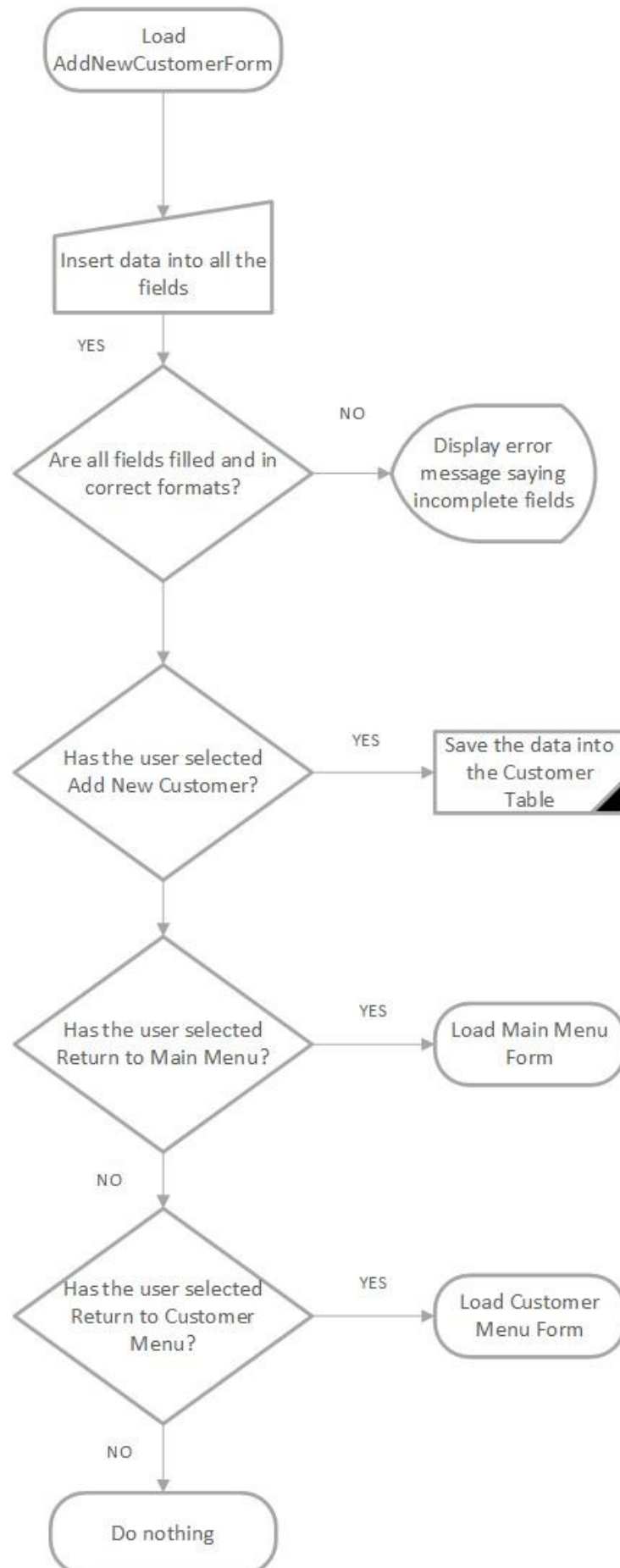
## 6. Game Menu



## 7. Supplier Menu



## 8. Add New Customer



9. Add New Game

9. Add New Supplier

10. Add New Order

**CG4**

# **Maintenance Documentation**

**META  GAMES**

## **Annotated Listings**

### 1. Splash Screen

```
Private Sub Detail_Click()
```

```
DoCmd.Close
```

```
DoCmd.OpenForm "LoginForm"
```

```
"When the user clicks the image then the loginform will open and  
the splashscreen will close"
```

```
End Sub
```

## 2. Login Form

```
Private Sub LoginButton_Click()
```

```
If UsernameText = "MetaGames" And PasswordText = "helloworld42"  
Then
```

```
"If the username and password are correct then close the login  
form"
```

```
DoCmd.Close
```

```
DoCmd.OpenForm ("Main Menu") "Open the main menu form once  
the login form has been closed"
```

```
Else
```

```
MsgBox "Please try again" "If password is incorrect then produce an  
error message saying try again"
```

```
End If
```

```
End Sub
```

## 3. Main Menu

```
Private Sub CloseDatabaseButton_Click()
```

```
Application.Quit acQuitPrompt
```

```
"When user clicks CloseDatabaseButton then the database  
application will be closed"
```

```
End Sub
```

```
Private Sub CustomerMenuButton_Click()
```

```
DoCmd.OpenForm ("Customer Menu")
```

```
DoCmd.Close acForm, "Main Menu"
```

```
"Open customer menu"
```

```
End Sub
```



```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub GameMenuButton_Click()
```

```
DoCmd.OpenForm ("Game Menu")
```

```
DoCmd.Close acForm, "Main Menu"
```

```
"Open game menu"
```

```
End Sub
```

```
Private Sub OrderMenuButton_Click()
```

```
DoCmd.OpenForm ("Order Menu")
```

```
DoCmd.Close acForm, "Main Menu"
```

```
"Open order menu"
```

```
End Sub
```

```
Private Sub SupplierMenuButton_Click()
```

```
DoCmd.OpenForm ("Supplier Menu")
```

```
DoCmd.Close acForm, "Main Menu"
```

```
"Open supplier menu"
```

```
End Sub
```

## 4. Customer Menu

```
Private Sub AddNewCustomerButton_Click()
```

```
DoCmd.OpenForm ("AddNewCustomerForm")
```

```
DoCmd.Close acForm, "Main Menu"
```

```
"Open the AddNewCustomer form"
```

```
End Sub
```

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub MainMenuButton_Click()
```

```
DoCmd.OpenForm ("Main Menu")
```

```
DoCmd.Close acForm, "Customer Menu"
```

```
"Open the main menu"
```

End Sub

```
Private Sub View_EditNewCustomerButton_Click()  
DoCmd.OpenForm ("View/EditCustomerForm")  
DoCmd.Close acForm, "Main Menu"
```

"Open the view/edit customer form"

End Sub

## 5. Order Menu

```
Private Sub AddNewOrderButton_Click()  
DoCmd.OpenForm ("AddNewOrderForm")  
DoCmd.Close acForm, "Order Menu"
```

"Open add new order form"

End Sub

```
Private Sub Detail_Click()
```

End Sub

```
Private Sub MainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "Order Menu"
```

"Open main menu form"

End Sub

```
Private Sub View_EditOrderButton_Click()  
DoCmd.OpenForm ("View/EditOrderForm")  
DoCmd.Close acForm, "Order Menu"
```

"Open view/edit order form"

End Sub

## 6. Supplier Menu

```
Private Sub AddNewSupplierButton_Click()
```

```
DoCmd.OpenForm ("AddNewSupplierForm")  
DoCmd.Close acForm, "Supplier Menu"
```

"Will add a new supplier record to the supplier table once clicked"

End Sub

Private Sub View\_EditSupplierButton\_Click()

DoCmd.OpenForm ("View/EditSupplierForm")

DoCmd.Close acForm, "Supplier Menu"

"Will take user to the view/edit supplier form"

End Sub

Private Sub MainMenuButton\_Click()

DoCmd.OpenForm ("Main Menu")

DoCmd.Close acForm, "Supplier Menu"

"Opens the main menu form"

End Sub

## 7. Game Menu

Private Sub AddNewGameButton\_Click()

DoCmd.OpenForm ("AddNewGameForm")

DoCmd.Close acForm, "Main Menu"

"Open the AddNewGameForm"

End Sub

Private Sub Detail\_Click()

End Sub

Private Sub MainMenuButton\_Click()

DoCmd.OpenForm ("Main Menu")

DoCmd.Close acForm, "Game Menu"

"Open the main menu"

End Sub

Private Sub View\_EditGameButton\_Click()

DoCmd.OpenForm ("View/EditGameForm")

DoCmd.Close acForm, "Main Menu"

"Open the view/edit game form"

End Sub

## 8. Add New Customer

Private Sub AddNewCustomerButton\_Click()

Dim db As DAO.Database  
Dim rs As DAO.Recordset

Set db = CurrentDb()  
Set rs = db.OpenRecordset("Customer", dbOpenDynaset) "Open the customer recordset so that it can be updated"

rs.AddNew  
rs("Customer Title") = CustomerTitle  
rs("Customer First Name") = CustomerFirstName  
rs("Customer Surname") = CustomerSurname  
rs("Customer Address line 1") = CustomerAddressLine1  
rs("Customer Address line 2") = CustomerAddressLine2  
rs("Customer City") = CustomerCity  
rs("Customer Postcode") = CustomerPostcode  
rs("Customer Contact Number") = CustomerContactNumber  
rs("Customer Date of Birth") = CustomerDateOfBirth

"Each text box on the form corresponds to a field in the table"

rs.Update  
MsgBox "Customer records have been updated."  
CustomerTitle = ""  
CustomerFirstName = ""  
CustomerSurname = ""  
CustomerAddressLine1 = ""  
CustomerAddressLine2 = ""  
CustomerCity = ""  
CustomerPostcode = ""  
CustomerContactNumber = ""  
CustomerDateOfBirth = ""

"Once the changes have been made and the user has clicked the AddNewCustomerButton then the table will be updated with a new recordset"

CustomerTitle.SetFocus  
rs.Close  
Set rs = Nothing  
End Sub

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub ReturnToCustomerMenuButton_Click()  
DoCmd.OpenForm ("Customer Menu") "Open customer menu"  
DoCmd.Close acForm, "ReturnToCustomerMenu"  
End Sub
```

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu") "Open main menu"  
DoCmd.Close acForm, "ReturnToMainMenu"
```

```
End Sub
```

## 9. Add New Order

```
Private Sub Form_Current()
```

```
    If IsNull(OrderIDText) Then  
        AllowEdits = True  
    Else  
        AllowEdits = False  
    End If
```

```
    "Only allow the user to change the OrderID if it is equal to zero"
```

```
End Sub
```

```
Private Sub AddNewOrderButton_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset
```

```
Set db = CurrentDb()  
Set rs = db.OpenRecordset("Order", dbOpenDynaset)
```

```
"Open the recordset of the order table so that new records can be  
added"
```

```
rs.AddNew  
rs("Order Date") = OrderDateText  
rs("Order Time") = OrderTimeText  
rs("Order Delivery Date") = OrderDeliveryDateText  
rs("Order Cost") = OrderCostText  
rs("Order VAT Cost") = OrderVATCostText  
rs("Order Total Cost") = TotalCostText
```

```
rs("Customer ID") = CustomerIDText  
rs("Month") = MonthText
```

"Relating each form field to the order table fields"

```
rs.Update  
MsgBox "Order Details have been added to the system. Thank you."  
OrderDateText = ""  
OrderTimeText = ""  
OrderDeliveryDateText = ""  
OrderCostText = ""  
OrderVATCostText = ""  
TotalCostText = ""  
CustomerIDText = ""  
MonthText = ""
```

"Once data has been entered and the user has clicked add new order then a new record will be added to the order table"

```
OrderDateText.SetFocus  
rs.Close  
Set rs = Nothing
```

End Sub

```
Private Sub OrderCostText_Click()
```

End Sub

```
Private Sub OrderVATCostText_Click()
```

End Sub

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "ReturnToMainMenu"
```

"Open main menu"

End Sub

```
Private Sub ReturnToOrderMenuButton_Click()  
DoCmd.OpenForm ("Order Menu")  
DoCmd.Close acForm, "ReturnToOrderMenuButton"
```

"Open order menu"

End Sub

```
Private Sub TotalCostValue_Click()
```

End Sub

10.      Add New Supplier

Private Sub AddNewSupplierButton\_Click()

Dim db As DAO.Database  
Dim rs As DAO.Recordset

Set db = CurrentDb()  
Set rs = db.OpenRecordset("Supplier", dbOpenDynaset)

"Open the supplier table recordset so that records can be added"

rs.AddNew

rs("Company Name") = CompanyNameText

rs("Contact Name") = ContactNameText

rs("Company Address") = CompanyAddressText

rs("City") = CityText

rs("Region") = RegionText

rs("Postcode") = PostcodeText

rs("Country") = CountryText

rs("Company Phone Number") = CompanyPhoneNumberText

"Relating each form field to the fields in the order table"

rs.Update

MsgBox "Supplier information has now been added to the database"

CompanyNameText = ""

ContactNameText = ""

CompanyAddressText = ""

CityText = ""

RegionText = ""

PostcodeText = ""

CountryText = ""

CompanyPhoneNumberText = ""

"Once user has entered data into the fields and clicked the add new supplier button then a new record will be added to the supplier table"

CompanyNameText.SetFocus

rs.Close

Set rs = Nothing

End Sub

Private Sub Detail\_Click()

End Sub

```
Private Sub ReturnToSupplierMenuButton_Click()  
DoCmd.OpenForm ("Supplier Menu")  
DoCmd.Close acForm, "ReturnToSupplierMenuButton"
```

"Open the supplier menu"

End Sub

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "ReturnToMainMenu"
```

"Open the main menu"

End Sub

#### 11. Add New Game

```
Private Sub AddNewGameButton_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset
```

```
Set db = CurrentDb()  
Set rs = db.OpenRecordset("Game", dbOpenDynaset) "Open the  
game table so that a new record can be added"
```

```
rs.AddNew  
rs("Game Title") = GameTitleText  
rs("Game Genre") = GameGenreText  
rs("Game Cost") = GameCostText  
rs("Stock") = StockText
```

"relating each text box field with the actual table field"

```
rs.Update  
MsgBox "A new game has been added to the database"  
GameTitleText = ""  
GameGenreText = ""  
GameCostText = ""  
StockText = ""
```

"Once new data is entered and the user has selected the add new game button then the table will update with the new record"



```
GameTitleText.SetFocus  
rs.Close  
Set rs = Nothing
```

```
End Sub
```

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub ReturnToGameMenuButton_Click()  
DoCmd.OpenForm ("Game Menu")  
DoCmd.Close acForm, "ReturnToGameMenuButton"
```

```
"Open the game menu"
```

```
End Sub
```

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "ReturnToMainMenu"
```

```
"Open the main menu"
```

```
End Sub
```

## 12. View/Edit Customer

```
Private Sub ReportButton_Click()  
DoCmd.OpenReport ("Monthly Income Report")  
DoCmd.Close acForm, "View/EditCustomerForm"
```

```
"Open monthly income report"
```

```
End Sub
```

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "View/EditCustomerForm"
```

```
"Open main menu"
```

```
End Sub
```

```
Private Sub ReturnToOrderMenuButton_Click()  
DoCmd.OpenForm ("Customer Menu")  
DoCmd.Close acForm, "View/EditCustomerForm"
```

```
"Open customer menu"
```

End Sub

Private Sub SaveChangesButton\_Click()

Dim db As DAO.Database

Dim rs As DAO.Recordset

Dim intCustID As Long

Set db = CurrentDb()

Set rs = db.OpenRecordset("Customer", dbOpenDynaset)

intCustID = CustomerID

rs.MoveFirst

'Finds the record using the ID and edits the record

rs.FindFirst ("CustomerID = " & intCustID)

rs.Edit

rs("Customer Title") = CustomerTitle

rs("Customer First Name") = CustomerFirstName

rs("Customer Surname") = CustomerSurname

rs("Customer Address Line 1") = CustomerAddressLine1

rs("Customer Address Line 2") = CustomerAddressLine2

rs("Customer City") = CustomerCity

rs("Customer Postcode") = CustomerPostcode

rs("Customer Contact Number") = CustomerContactNumber

rs("Customer Date of Birth") = CustomerDateOfBirth

rs.Update

"When changes to the record have been made the original record in the customer table will be updated when the user clicks the save button"

MsgBox ("Customer details have been edited") "Display message box"

End Sub

### 13. View/Edit Order

Private Sub DeleteButton\_Click()

Dim db As DAO.Database

Dim rs As DAO.Recordset

'Sets variables

Set db = CurrentDb()

Set rs = db.OpenRecordset("Order", dbOpenDynaset)

'Declares variables

Dim intOrderID As Integer

```
intOrderID = OrderID
rs.MoveFirst

'Finds the recordset using the order id
rs.FindFirst ("OrderID =" & intOrderID)
'Deletes the record with the same ID
rs.Delete
rs.Close
MsgBox "Order record has been deleted from the database."
```

```
End Sub
```

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub ReturnToMainMenuButton_Click()
DoCmd.OpenForm ("Main Menu")
DoCmd.Close acForm, "View/EditOrderForm"
```

```
"Open main menu"
```

```
End Sub
```

```
Private Sub ReturnToOrderMenu_Click()
DoCmd.OpenForm ("Order Menu")
DoCmd.Close acForm, "View/EditOrderForm"
```

```
"Open order menu"
```

```
End Sub
```

```
Private Sub SaveChangesButton_Click()
```

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim intCustID As Long
Set db = CurrentDb()
Set rs = db.OpenRecordset("Game", dbOpenDynaset)
```

```
intCustID = OrderID
rs.MoveFirst
'Finds the record using the ID and edits the record
rs.FindFirst ("OrderID = " & intCustID)
```

```
rs.Edit
rs("Order Date") = OrderDate
rs("Order Time") = OrderTime
```

```
rs("Order Delivery Date") = GameCost  
rs("Order Cost") = OrderCost  
rs("Order VAT Cost") = OrderVATCost  
rs("Order Total Cost") = OrderTotalCost  
rs.Update
```

'Upon clicking the save button all changes made to the record will be saved and the record will be updated in the order table'

```
MsgBox ("Order details have been edited.")
```

```
End Sub
```

#### 14. View/Edit Game

```
Private Sub DeleteButton_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset  
'Sets variables  
Set db = CurrentDb()  
Set rs = db.OpenRecordset("Game", dbOpenDynaset)
```

```
'Declares variables  
Dim intGameID As Integer  
intGameID = GameID  
rs.MoveFirst
```

```
'Finds the recordset using the game id  
rs.FindFirst ("GameID =" & intGameID)  
'Deletes the record with the same ID  
rs.Delete  
rs.Close  
MsgBox "Game record has been deleted from the database."
```

```
End Sub
```

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub ReturnToGameMenuButton_Click()  
DoCmd.OpenForm ("Game Menu")  
DoCmd.Close acForm, "View/EditGameForm"
```

```
"Open game menu"
```

End Sub

```
Private Sub ReturnToMainMenuButton_Click()  
DoCmd.OpenForm ("Main Menu")  
DoCmd.Close acForm, "View/EditGameForm"
```

"Open main menu"

End Sub

```
Private Sub SaveChangesButton_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset  
Dim intCustID As Long  
Set db = CurrentDb()  
Set rs = db.OpenRecordset("Game", dbOpenDynaset) "Open the  
game table recordset so that new records can be added"
```

```
intCustID = GameID  
rs.MoveFirst  
'Finds the record using the ID and edits the record  
rs.FindFirst ("GameID = " & intCustID)
```

```
rs.Edit  
rs("Game Title") = GameTitle  
rs("Game Genre") = GameGenre  
rs("Game Cost") = GameCost  
rs("Game Quantity") = GameQuantity  
rs.Update
```

"When the save button is clicked the record will be updated"

```
MsgBox ("Game details have been edited.")
```

End Sub

## 15. View/Edit Supplier

```
Private Sub DeleteButton_Click()
```

```
Dim db As DAO.Database  
Dim rs As DAO.Recordset  
'Sets variables  
Set db = CurrentDb()  
Set rs = db.OpenRecordset("Order", dbOpenDynaset)
```

```
'Declares variables
```

```
Dim intSupplierID As Integer
intSupplierID = SupplierID
rs.MoveFirst

'Finds the recordset using the game id
rs.FindFirst ("SupplierID =" & intSupplierID)
'Deletes the record with the same ID
rs.Delete
rs.Close
MsgBox "Supplier record has been deleted from the database."
```

```
End Sub
```

```
Private Sub Detail_Click()
```

```
End Sub
```

```
Private Sub ReturnToMainMenuButton_Click()
DoCmd.OpenForm ("Main Menu")
DoCmd.Close acForm, "View/EditSupplierForm"
```

```
"Open main menu"
```

```
End Sub
```

```
Private Sub ReturnToSupplierMenuButton_Click()
DoCmd.OpenForm ("Supplier Menu")
DoCmd.Close acForm, "View/EditSupplierMenus"
```

```
"Open supplier menu"
```

```
End Sub
```

```
Private Sub SaveChangesButton_Click()
```

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
Dim intCustID As Long
Set db = CurrentDb()
Set rs = db.OpenRecordset("Supplier", dbOpenDynaset)
```

```
intCustID = SupplierID
rs.MoveFirst
'Finds the record using the ID and edits the record
rs.FindFirst ("SupplierID =" & intCustID)
```

```
rs.Edit
rs("Company Name") = CompanyName
```

```
rs("Contact Name") = ContactName  
rs("Company Address") = CompanyAddress  
rs("City") = City  
rs("Regions") = Regions  
rs("Postcode") = Postcode  
rs("Country") = Country  
rs("Company Phone Number") = CompanyPhoneNumber  
rs.Update
```

"All changes made to the record will be saved upon clicking the save button and the record in the supplier table will be updated"

MsgBox ("Supplier details have been edited.")

End Sub

#### 16. Monthly Income Report

```
Private Sub PrintButton_Click()
```

```
    DoCmd.RunCommand acCmdPrint "When the user clicks the print  
    button run the print process"
```

```
PrintButton_Click_Exit: "If user cancels the print then the print  
window will be closed"
```

```
    Exit Sub
```

End Sub

```
Private Sub ReportHeader_Format(Cancel As Integer, FormatCount  
As Integer)
```

End Sub

## Procedure / Subroutine Details

| <u>Form / Procedure</u>            | <u>Purpose / Action</u>  |
|------------------------------------|--|
| <b>Login Form</b>                  | <b>To log into the database</b>  |
| LoginButton_Click()                | Checks that the password and username are correct, if it is then the main menu will open, if not then a message box will appear that says "Please Try Again" |
| <b>Splash Screen form</b>          | <b>Displays the logo upon opening the database</b>   |
| Detail_Click()                     | Once the user clicks on the form then they will be taken to the Login Form and the splash screen form will be closed.  |
| <b>Main Menu form</b>              | <b>Helps user navigate to all forms</b>  |
| GameMenuButton_Click()             | Opens the game menu and closes the main menu   |
| OrderMenuButton_Click()            | Opens the order menu and closes the main menu  |
| SupplierMenuButton_Click()         | Opens the supplier menu and closes the main menu   |
| CustomerMenuButton_Click()         | Opens the customer menu and closes the main menu   |
| CloseDatabaseButton_Click()        | Closes/quits the entire database   |
| <b>Customer Menu</b>               | <b>Helps user navigate all customer based forms</b>  |
| AddNewCustomerButton_Click()       | Opens the 'AddNewCustomer' data entry form and closes the customer menu form   |
| MainMenuButton_Click()             | Opens the main menu form and closes the customer menu form   |
| View_EditNewCustomerButton_Click() | Opens the 'View/EditCustomer' data entry form and closes the customer menu form  |
| <b>Order Menu</b>                  | <b>Helps user navigate all order</b>   |



# Maintenance Documentation

|  | based forms |
|--|-------------|
|  |             |
|  |             |
|  |             |
|  |             |
|  |             |

CG4

Testing

METAGAMES

When developing a system it is essential that it is tested thoroughly and throughout. A programme that doesn't work isn't much use to anybody, so I will be testing all areas of the system and fixing them upon discovery, before continuing through my search for bugs. If errors are not dealt with then it could influence the function and outcome of various other processes within the database.

I will be carrying out a detailed test on my completed system myself, to ensure all processes are working correctly and it is functioning the way I want it to. I will then ask a student who knows nothing of the system to test it. This type of testing is useful because without a knowledge of the system it may be easier to find errors that you were unaware of.

My testing will be split into:

- Navigational interface testing – this will include ensuring that all command buttons work and I can easily navigate from one form to the other via the command buttons.
- Data handling – as there is a lot of data stored and manipulated in my programme, it is important that this is tested and works as it should. For this I'll be testing things like the adding, editing and deleting record functions.
- Validation tests – This will involve testing the validation rules I have implemented. I will be testing two validation techniques, a range check and a format check. The way in which this will be structured is by testing for: normal data, invalid data, extreme data (high) and extreme data (low).

| Test No. | Test Type        | Test Data   | Expected Result                | Actual Result         |
|----------|------------------|---|--------------------------------|-----------------------|
| 1.       | Add New Customer | Mrs, Sarah, Marshal, 45 St. Andrews Street, Cardiff, Cardiff, CF42 9PD, 0753264858 5, 09/10/1982. | Saved into the Customer Table. | See print screen 1-2. |
| 2.       | Add New Order    | 05/05/2016, 11:30, 10/05/2016, 20, May. Sub-form: 3, 2.   | Saved in the Order Table.      | See print screen 3-4. |

|    |  |   |  |   |
|----|--|---|--|---|
| 3. | Add New Game   | Metal Gear Solid: The Phantom Pain, Stealth, £25.00, 45, 7.                           | Saved in the Game Table.   | See print screen 5-6.   |
| 4. | Add New Supplier   | Konami, Henry Lincoln, 12 Queen Street, Bristol, Bristol, BR20 8HD, UK, 0783746256 1. | Saved in the Supplier Table.   | See print screen 7-8.   |
| 5. | Single Query 1 / View RPGs                               | N/A   | I expect to view records of all the games that have the genre 'RPG'.   | Correct data is displayed. See print screen 9.  |
| 6. | Single Query 2 (parameter) / View Delivery date order(s) | Delivery date = 07/05/2016  | I expect to view all order details for two order records on the 07/05/2016   | Correct data is showing. All order fields and two separate order records. See print screen 10-11.                                   |
| 7  | Multiple Query / View First Person Shooters              | N/A   | I expect to view all games that have the 'First Person Shooter' genre and the date they were ordered, as well the customer ID. | Correct data is being displayed, showing all records of 'First Person Shooter' orders along with the date of order and customer ID. |

# Testing

|     |  |                       |   |   |
|-----|--|-----------------------|---|---|
|     |  |                       |   | See print screen 12.  |
| 8.  | Multiple Query 2 / View Customer Details and Order | N/A                   | I expect to view all orders, along with customer details, such as ID, surname and contact number, as well as the game ID, order ID, game quantity and order total cost. | Correct data is shown, including all fields as expected. See print screen 13.                           |
| 9.  | Parameter Query / View Customer                    | Customer ID = 23      | I expect to view all of the customer data for one particular customer ID.   | Correct data is shown, including all customer fields for the right customer ID. See print screen 14-15. |
| 10. | Action Query / Update Window Cost                  | N/A                   | I expect to see the cost of the 'RPG' genre to be reduced by 20%.   | Data was correctly updated. 20% decrease in the cost of RPGs. See print screen 16-18.                   |
| 11. | View/Edit Customer                                 | 66 Bay Road, CF27 5BR | I expect that the customer with ID 10 will have his address line 1 and  | Data was correctly edited, the customer's address line 1 and postcode                                   |

# Testing

|     |                                   |                                      |   |   |
|-----|-----------------------------------|--------------------------------------|---|---|
|     |                                   |                                      | postcode changed.   | changed. See print screen 19-21.  |
| 12. | View/Edit Order                   | Delivery date:<br>26/04/2016         | I expect that the delivery date of an order will change from 07/05/2016 to 09/05/2016.                | Data is correctly updated, now showing the delivery date for this order as 09/05/2016. See print screen 22-24.                      |
| 13. | View/Edit Game                    | Stock:<br>15                         | I expect that the stock of the game will be changed from 20 to 15                                     | Data has been correctly edited and the stock of the game is now 15 instead of 20. See print screen 25-27.                           |
| 14. | View/Edit Supplier                | Company Address:<br>52 Willow Street | I expect to see the company address of a supplier to change from 73 broad street to 52 Willow Street. | Data has been correctly edited and the company address of the supplier has now changed to 52 Willow Street. See print screen 28-30. |
| 15. | Test calculated field in sub-form | Game ID:<br>3<br>Quantity:<br>2      | I expect that the total cost of the order should be £23.04.   | Calculation DID NOT produce the correct result.   |

|     |   |                    |  |                                       |
|-----|---|--------------------|--|---------------------------------------|
|     |   |                    | ((£9.60 x 2) + 20%)  | See print screen 31.                  |
| 16. | Data Validation 1 - Normal Data                     | Quantity: 4        | Accepted   | See print screen 32-33.               |
| 17. | Data Validation 1 -Invalid Data                     | Quantity: 101      | "Invalid Quantity"   | See print screen 34.                  |
| 18. | Data Validation 1 - Extreme Data 1                  | Quantity: 99       | Accepted   | See print screen 35.                  |
| 19. | Data Validation 1 - Extreme Data 2                  | Window Quantity: 0 | Accepted   | See print screen 36.                  |
| 20. | Data Validation 2 - Normal Data                     | Postcode: CF82 6DP | Accepted   | See print screen 37.                  |
| 21. | Data Validation 2 - Invalid Data                    | Postcode: CF23-8DH | "Incorrect Format"   | See print screen 38.                  |
| 22. | Data Validation 2 - Extreme Data 1                  | Postcode: AA99 9AA | Accepted   | See print screen 39.                  |
| 23. | Data Validation 2 - Extreme Data 2                  | Postcode: ZZ00 0ZZ | Accepted   | See print screen 40.                  |
| 24. | Test Navigation of User Interface (Command Buttons) | N/A                | All commands to navigate the system are working properly.        | See lecturer's written confirmation . |
| 25. | Splash Screen.                                      | N/A                | I expect the Splash screen to pop up in a separate window when I | See print screen 41-42.               |

|     |  |   |   |                         |
|-----|--|---|---|-------------------------|
|     |  |   | select the form, then when I click the window I expect it to close and take me to the login form. |                         |
| 24. | Close Database                                       | N/A   | I expect to see the whole database close when I click the 'Close Database' button.                | See print screen 43-44. |
| 25. | Test Security 1<br>(correct password and username)   | Username: MetaGames<br>Password: Helloworld42 | Database will open.   | See print screen 45-46. |
| 26. | Test Security 2<br>(incorrect password and username) | Any random password that is incorrect.        | Error message reading "Please try again"  | See print screen 47-48. |



## CG4

# Evaluation



## Questionnaire

Suitability:

Please enter Yes/No

1. Are you able to create and open a table?

☐

-----

# Evaluation

2. Are you able to access and view the information in all 3 tables? ----- ☐

3. Are you able to manipulate the data (Edit, delete, add)? ----- ☐

4. Are you able to produce reports? ----- ☐

5. How professional does it look? ----- ☐

6. Are the reports professional in design and look? ----- ☐

7. Is the data in the reports well-structured and organised? ----- ☐

8. Are you able to search through the data by certain fields? ----- ☐

**TOTAL**

-----

# Evaluation

## Questionnaire

### Usability:

Please enter a number  
from 1-10

1. How easy and intuitive does it feel to use the user interface? ----- ☐
2. How easy does it feel to navigate the user interface? ----- ☐
3. How fitting is the colour scheme? ----- ☐
4. How well designed in terms of layout is the program? ----- ☐
5. How professional does it look? ----- ☐
6. Does the program look well designed? ----- ☐
7. How functional / useful do you think the program is? ----- ☐
8. Does the system allow you to easily organise data? ----- ☐

**TOTAL**

---