National College of Ireland

BSc in Computing

2015/2016

Smart-Shop (Group B)

Project Report

|  |  |
| --- | --- |
| Number | Name |
| X13114352 | Declan Barnes |
| X12724145 | Chris Doran |
| X13406038 | Sean McDermott |
| X13401792 | Daniel Gorman |
| X13114581 | John McNamara |
| X13114557 | Youcef O Conner |

**Table of Contents**

[Executive Summary 3](#_Toc433028362)

[1 Introduction 4](#_Toc433028363)

[1.1 Background 4](#_Toc433028364)

[1.2 Aims 4](#_Toc433028365)

[2 Proposal: 5](#_Toc433028366)

[2.1 Technologies 6](#_Toc433028367)

[3 System 7](#_Toc433028368)

[3.1 Business Rules 7](#_Toc433028369)

[3.2 Requirements 8](#_Toc433028370)

[3.3 Functional requirements 8](#_Toc433028371)

[3.3.1 Use Case Diagram 8](#_Toc433028372)

[3.3.2 Requirement 1 <Create Customers> 8](#_Toc433028373)

[3.3.3 Requirement 2 <Modify Customers> 9](#_Toc433028374)

[3.3.4 Requirement 3 <Place Order> 10](#_Toc433028375)

[3.3.5 Requirement 4: <Payment> 11](#_Toc433028376)

[3.3.6 Requirement 5: <Delivery and Status> 12](#_Toc433028377)

[3.4 Data Requirements 13](#_Toc433028378)

[3.5 Data Dictionary: 13](#_Toc433028379)

[3.6 Design and Architecture 15](#_Toc433028380)

[3.7 Conceptual ERD 16](#_Toc433028381)

[3.8 Logical ERD 17](#_Toc433028382)

[3.9 CRUD: 18](#_Toc433028383)

[4 Conclusions 19](#_Toc433028384)

[4.1 Appendix-Project Plan 20](#_Toc433028385)

# Executive Summary

 Our database was created for an online retail store called Smart-Shop database it has multiple tables that store’s the data that we need for our Smart-shop database, it allows Smart-Shop to sell its clothing products to customers while tracking the order status of those products the following tables are what’s in our database:

CUSTOMER

ORDER

ORDER DETAILS

PRODUCT

PRODUCTS LINE

PAYMENT

SUPPLIER

SUPP\_PROD

Each of the above tables have primary keys, Foreign Key’s etc. the database provides us with the opportunity to create, check transactions and customer orders etc.

# Introduction

The purpose of this document is to set out the requirements for the development of a Corporate Database. The corporate database system will be designed and populated to represent an online clothes shop called Smart Shop.

The intended customers are men and women of all ages and sizes.

The intended user of the corporate database system will be an employee of Smart Shop.

## Background

The background of the project is to design, develop and populate a corporate database system for an online clothes shop called Smart Shop. To arrive at the completion of the project successfully the following steps will need to be implemented properly in succession.

1. The requirements specification document will required to be written in detail including functional requirements, non-functional requirements, user requirements and outline how the database system can evolve in the system evolution section.
2. A data dictionary will also be constructed. A data dictionary is a centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format.
3. Entity Relationship Diagram (ERD). An ERD is a chart that visually represents the relationship between database entities.
4. Script files have to be created to help populate our corporate database. The database needs to contain 100,000 records.
5. Unstructured data will also need to be stored in an xml file through a GUI.

The time frame of this project is 5 weeks.

## Aims

The aim of the project is to create a database for our online clothing retail store (Smart-Shop), the database should contain 100,000 records and have several tables within it, for example: Customers, Orders, etc.

# Proposal:

**Overview:** The background of the project is to design, develop and populate a corporate database system for an online clothes shop called Smart Shop and must contain up to 100k records overall, we will be making our final decision on Thursday 17th of September.

**Project Team**

|  |  |  |
| --- | --- | --- |
| **Student Number** | **Student name** | **Role** |
| X12724145 | Chris Doran | Data Administrator |
| X13114352 | Declan Barnes | Project Manager |
| X13114581 | John McNamara | Data Analyst |
| X13406038 | Sean McDermott | Data Architect |
| X13114557 | Yousef O Conner | Web Design & Developer |
| X13401792 | Daniel Gorman | Business Analyst |

**Roles:**

1. Project Manager
2. Data Architect/Administrator
3. Business/Data Analyst
4. Web Designer & Developer

## Technologies

**MySQL:** is an open-source RDBMS, it is the world’s 2nd most widely used RDBMS, and the most widely used open-source client-server model RDBMS. We used this to create our database.

**Notepad++:** Notepad++ is a popular and free source code editor, we used this to type up our SQL statements before doing them for real in MySQL.

**Microsoft Visio:** Microsoft Visio is a [diagramming](https://en.wikipedia.org/wiki/Diagramming_software) and graphics application and is part of the [Microsoft Office](https://en.wikipedia.org/wiki/Microsoft_Office) family. We used this to do all our diagrams (Logical ERD, Conceptual ERD) as it had several built in features that made creating diagrams much simpler, saving valuable time.

**Microsoft Project:** Is a project management software that is used to help project managers structure what needs to be done, by what date it needs to be done by and by who. Our team used Microsoft project to create Gantt charts at the start of our project.

**WhatsApp:**  is an [instant messaging](https://en.wikipedia.org/wiki/Instant_messaging) app for [smartphones](https://en.wikipedia.org/wiki/Smartphones), we used this to create our group on WhatsApp, which allowed us to communicate with each other when not in class (days off etc.).

# System

## Business Rules

**BR1**: One CUSTOMER may have many ORDERS, many ORDERS may relate to one CUSTOMER.

**BR2:** One PAYMENT may relate to many ORDER’S, many ORDER’S may be paid by one PAYMENT.

**BR3:** One ORDER may contain many ORDER DETAILS, many ORDER DETAILS may relate to one ORDER.

**BR5:** One PRODUCT may be listed in many ORDER DETAILS, many ORDER DETAILS can relate to one PRODUCT.

**BR6:** One PRODUCT can relate to many SUPP\_PROD’S, many SUPP\_PROD’S may contain many PRODUCTS.

**BR7:** One PRODUCT LINE can have many PRODUCTS, many PRODUCTS may relate to one PRODUCT LINE.

**BR8**: One SUPPLIER can supply many SUPP\_PROD, many SUPP\_PROD relate to one SUPPLIER.

## Requirements

**Requirement <Database>**

The clients want my team to create and populate a database for their Smart-Shop online shopping business. The database will be created using MySQL and it will be developed for fast and efficient querying.

**Requirement <Unstructured Data>**

My team will build an automated Website that will create unstructured data file format. This will help the clients in the near future as the clients have plans to link the website to the database but for now just want to create bought separately.

## Functional requirements

### Use Case Diagram

N/A

### Requirement 1 <Create Customers>

**Description & Priority:**

This requirement is to create customers in the database. This is the main function as the database needs customers to progress.

**Use Case:**

The scope of this use case is to get the customers details added to the database. The system will to able to create customers such as add their names, customer ID's and additional information that will be helpful in creating the system. We will input the required fields for the customers to put their details in.

**Flow Description**

**Precondition**

The Customer accesses the database.

**Activation**

The Customer enters their details in.

**Main flow**

1. The system allows the actor in
2. The Customer enters their details
3. The system creates the customer
4. The Customer is logged in

**Post condition**

The system tells the customer they have been successfully added to the database.

### Requirement 2 <Modify Customers>

**Description & Priority**

This requirement is for the system to be able to add, update or delete a customer’s record.

**Use Case:**

The system will be able to modify the customers table such as add customers, update customers and delete customers from the table.

**Flow Description**

**Precondition**

The user accesses the customers table

**Activation**

The user adds updates or deletes a record

**Main flow**

* 1. The user enters the customer table
  2. The user modifies a record
  3. The system accepts the modification
  4. The users leaves the customer table

**Post condition**

The customer table is modified.

### Requirement 3 <Place Order>

**Description & Priority**

This requirement is for the customer to be able to place an order.

**Use Case:**

The system will allow the customer to place an order. The customer’s details such as the customer ID will be put into the orders table so the database knows what customer is making which order.

**Flow Description**

**Precondition**

The customer is logged in

**Activation**

The customer places an order

**Main flow**

1. The system ensures the customer is logged in
2. The customer places an order
3. The system accepts the order
4. The customer continues

**Post condition**

The order is placed

### Requirement 4: <Payment>

**Description & Priority**

Payment has a very high importance, the order can’t be created without payment first. Payment takes the Card details from a customer, processes the details before placing a delivery order.

**Use Case:**

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

The <Actor> logs in, places an order, and enters payment details, system checks details.

**Flow Description**

**Precondition**

The process begins when the <Actor> logs in.

**Activation**

The process begins when the <Actor> proceeds to checkout.

**Main flow**

1. The database identifies the <Actor> log in credentials.
2. The <Actor> picks items.
3. The <Actor> proceeds to checkout.
4. The <Actor> enters credit card details.
5. The System processes the <Actor> details.
6. The system notifies customer of successful payment.

**Post condition**

The system continues to the delivery page

### Requirement 5: <Delivery and Status>

**Description & Priority:**

Delivery and status is a vital requirement after a customer has logged in and places an order, the order is then sent to the delivery department to process the order and deliver the items to the customer. It will also create a unique delivery number assigned to each order for the customer to track from loading bay to delivery to the destination.

**Use Case:**

The Customer logs in, places an order, the order is then sent to the Delivery department.

**Flow Description**

**Precondition**

The process begins when the <Actor> logs in.

**Activation**

This use case starts when an <Actor> places an order

**Main flow**

1. The database identifies the <Actor> log in credentials.
2. The <Actor> picks items.
3. The <Actor> proceeds to checkout.
4. The <Actor> enters credit card details.
5. The System processes the <Actor> details.
6. The system sends delivery request

**Post condition**

The system notifies the <Actor> their order was placed successfully.

## Data Requirements

The Smart-Shop database will need several tables to store data on customers, orders, the orders status, product, suppliers etc. This will allow us to see the customer’s details, what they have purchased, the quantity and price, what the order status is i.e. sent, cancelled etc. Payments by customers will also be noted with its own Payment table, it will also allow us to keep track on our suppliers as well. The data dictionary is shown below for further information on the data requirements (Entities, Attributes etc.)

## Data Dictionary:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***CUSTOMER*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Cust\_ID | NUMBER | 11 | PRIMARY KEY | Cust\_ID, auto-Generated |
| F\_Name | VARCHAR | 50 | DEFAULT NULL | First name |
| L\_Name | VARCHAR | 50 | DEFAULT NULL | Surname |
| Phone | NUMBER | 50 | DEFAULT NULL | Phone number |
| Address | VARCHAR | 50 | DEFAULT NULL | Address |
| PostalCode | NUMBER | 50 | DEFAULT NULL | Postal Code |
| Country | VARCHAR | 50 | DEFAULT NULL | Country |
| ***ORDERS*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Order\_ID | NUMBER | 11 | PRIMARY KEY | Order\_ID, auto-Generated |
| Order\_Date | DATE |  | NOT NULL | Date of Order |
| Shipping\_Date | DATE |  | NOT NULL | Date of Shipping |
| Pay\_ID | NUMBER | 11 | FOREIGN KEY | The Pay\_ID |
| Status | VARCHAR | 45 | DEFAULT NULL | Order Status |
| Cust\_ID | NUMBER | 11 | FOREIGN KEY | Customers ID |
| ***ORDER DETAILS*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Prod\_ID | NUMBER | 11 | PRIMARY KEY | Product id from products table |
| Order\_ID | NUMBER | 11 | FOREIGN KEY | Order id from orders table |
| Quantity ordered | NUMBER | 11 | NOT NULL | Quantity bought |
| Price | NUMBER | 10,0 | NOT NULL | Total Price |
| ***PRODUCTS*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Prod\_ID | NUMBER | 11 | PRIMARY KEY | Prod\_ID, auto-Generated |
| Price | NUMBER | 10,0 | NOT NULL | Price of item |
| PLine\_ID | NUMBER | 11 | FOREIGN KEY | From Product Line Table |
| Prod\_Type | VARCHAR | 100 | DEFAULT NULL | Product Type i.e Jeans,Jumper |
| ***PRODUCTS LINE*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| PLine\_ID | NUMBER | 11 | PRIMARY KEY | PLine\_ID, auto-Generated |
| Quantity\_Instock | NUMBER | 11 | DEFAULT NULL | Quantity in stock |
| ***PAYMENTS*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Pay\_ID | NUMBER | 11 | PRIMARY KEY | Pay\_ID, auto-Generated |
| Order\_ID | NUMBER | 11 | FOREIGN KEY | From Orders Table |
| Payment\_Date | DATE |  | DEFAULT NULL | Payment Date |
| Card\_Type | VARCHAR | 45 | DEFAULT NULL | Debit, Laser etc. |
| Cardholder\_Name | VARCHAR | 100 | DEFAULT NULL | Name of Card Holder |
| Card\_Num | NUMBER | 11 | NOT NULL | Number on Card |
| Expiry\_Date | DATE |  | DEFAULT NULL | Card Expiry Date |
| CV2\_Num | NUMBER | 11 | NOT NULL | Security number on card |
| ***SUPPLIER*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Supp\_ID | NUMBER | 11 | PRIMARY KEY | Supp\_ID, auto-Generated |
| Prod\_ID | NUMBER | 11 | FOREIGN KEY | From Products Table |
| Supp\_Name | VARCHAR | 45 | DEFAULT NULL | Suppliers Name |
| Supp\_Phone | NUMBER | 11 | NOT NULL | Suppliers phone number |
| ***SUPP\_PROD*** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Prod\_ID | NUMBER | 11 | PRIMARY KEY | From Products Table |
| Supp\_ID | NUMBER | 11 | PRIMARY KEY | From Supplier Table |

## Design and Architecture

The design of the project was straightforward but taxing at times, first we had to come up with an idea for our project to which we all agreed, then we needed to come up with a project plan and requirements spec, after this was done we got deeper into the design of the project by creating ERD’s. After the ERD’s we moved on to developing the actual database on MySQL along with doing the CRUD diagram.

As part of the design we had to create script files, these script files helped us create and populate our database along with inserting tables into our database.

As part of testing of our project we made sure to test that our database had what it need and what we defined in our requirements specification, we done this by running queries in MySQL to see the results.

## Conceptual ERD



## Logical ERD



## CRUD:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity Process** | Customer | Order | Order Details | Payment | Product | Product Line | Supplier | Supplier\_Prod |
| Create Customers |  |  |  |  |  |  |  |  |
| Maintain Customer | **R** |  |  |  |  |  |  |  |
| Modify  Customers | **UD** |  |  |  |  |  |  |  |
| Place Order | **CRU** | **CUD** |  |  |  |  |  |  |
| Payment | **R** |  | **R** | **CUD** |  |  | **R** |  |
| Delivery | **RU** | **U** | **R** |  | **R** | **R** | **RU** |  |
| Status | **R** |  |  |  |  |  | **RU** |  |
| Stock | **R** | **U** |  |  | **UD** | **U** | **RUD** |  |

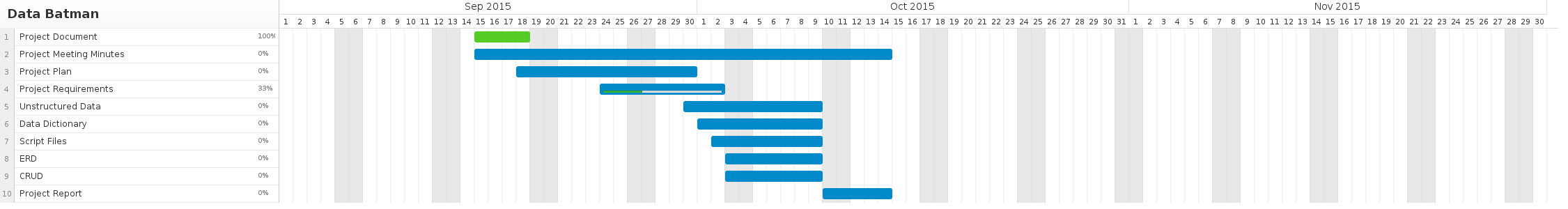
**C= Create R=Read U=Update D=Delete**

# Conclusions

After completing the first part of our project I can say that we are happy with the overall result, the project has many advantages for us as a group, for one it has allowed us to become more efficient with databases and has taught us more about setting up databases than we learned last year in our introduction to databases class, this due to the more hand on practical work that we have to do.

The disadvantages would be the time constraint on our group members as we had several other projects and continued assessments to study for.

## Appendix-Project Plan



Project Document (Declan)

Project Meeting Minutes (Declan, Chris, Daniel, Youcef, Sean, John)

Project Requirements (Declan, Chris, Daniel, Youcef, Sean, John)

Unstructured data (Youcef)

Data Dictionary (Daniel)

Script Files (Youcef)

ERD (Chris, Sean)

Crud (John, Daniel)

Project Report (Declan)

**Did the plan work well and was it followed?**

Yes our plan worked very well for us, all of our assigned work was completed in a timely manner. Everyone in our Group followed the plan to the letter in order to get the most out of it and prevent problems.