

KISACASA

# eMart Database System

---

## Final Report

**Kirsty Williams, Sara Chattun, Samantha Kviecinskas**

**5/1/2012**

This final report presents a general summary of the creation of a database system developed for a fictional mini market, eMart.

## **Acknowledgements**

To every team member, Sara Chattun, Kirsty Williams and Samantha Kviecinskas and a special mention to Carl Blakemore who was a part of Kisacasa throughout the first two phases but unfortunately left university. Finally, we would like to thank Sebastian Coopes for his help, support and guidance throughout the project.

## Contents

Acknowledgements.....	2
List of Figures and Tables.....	Error! Bookmark not defined.
Members and Roles .....	11
Application Description .....	12
Application Domain.....	12
Types of User.....	12
Customers .....	12
Staff .....	12
Typical Queries.....	12
Typical Reports.....	12
System Achievements.....	13
Conformance with Standards .....	13
Code Structure .....	13
Tools .....	14
Security .....	14
Session Handling .....	15
Search Engine Optimisation.....	15
Forgotten Passwords .....	15
Administering Products.....	15
Shopping Cart .....	15
Customer Account Functionality .....	16
Order Process.....	16
Input Validation .....	16
Evaluation.....	16
Strengths .....	16
Weaknesses and their Improvements .....	17
Working as a team .....	17
Extensions.....	17
Professional Issues .....	18
Public Interest .....	18
Professional Competence and Integrity.....	18
Duty to Relevant Authority.....	19
Duty to the Profession.....	19

Items not complied with.....	19
Bibliography .....	20
Appendix .....	21
Appendix A.....	22
1. Introduction.....	23
1.1 Project Statement .....	23
1.2 Feasibility Study .....	23
Implementation:.....	23
Technical: .....	23
Operational: .....	23
Economic:.....	23
1.3 Data Protection and eMart .....	24
2. Database Planning.....	24
2.1 Mission Statement.....	24
2.2 Mission Objectives.....	24
3. System Definition .....	26
3.1 System Scope and Boundary.....	26
3.2 User Views.....	26
Organisation owner.....	26
System Administrator .....	26
Manager .....	26
Department Managers .....	27
Supervisors .....	27
Merchandiser .....	27
Assistants .....	27
Customers .....	27
4. Requirements Collection.....	28
4.1 Interviews .....	28
Interview with the Managing Director of eMart.....	28
Interview with the Store Manager of eMart .....	28
Interview with a regular shop customer of eMart.....	30
4.2 Ecommerce Walkthrough.....	30
4.3 Domain Analysis.....	31
Overview .....	31

Stock Management System .....	31
Payroll.....	32
4.4 Organisational Structure .....	33
4.5 Context Diagram.....	34
Requirements Analysis .....	35
5.1 Stakeholders.....	35
5.2 Actors and Goals.....	35
5.3 Use Case Descriptions .....	36
5.4 Use Case Diagrams .....	66
Orders .....	66
System .....	66
Accounts .....	66
HR Management.....	67
5.5 Transaction Requirements.....	69
5.6 System Specifications .....	70
Initial database size .....	70
Database rate of growth .....	70
The types and average number of record searches .....	70
Networking and shared access requirements.....	70
Performance .....	70
Security .....	70
User interface .....	71
5.7 Functional Requirements .....	71
5.8 Non-functional Requirements .....	79
5.9 Prioritisation of Requirements .....	84
5.10 Success Criteria for Requirements .....	85
Appendix B .....	86
6. Conceptual Database Design.....	87
6.1 Business Rules .....	87
6.2 Data Flow Diagram.....	89
6.3 Functional Descriptions .....	90
7. Logical Database Design .....	90
7.1 ER Model.....	90
7.1.1 Entity Identification.....	90

7.1.2	Relationship Identification .....	91
7.1.3	ER Diagram.....	92
7.1.4	Interaction Diagrams .....	93
7.1.5	Sequence Diagrams .....	96
7.2	Table Structures .....	104
7.3	Data Dictionary .....	106
7.4	Transaction Matrix .....	114
8.	Physical Database Design .....	119
8.1	Base Tables .....	119
8.1.1	Department Table .....	119
8.1.2	UserLogin Table.....	119
8.1.3	Staff Table.....	119
8.1.4	Stock Table .....	120
8.1.5	StockControl Table .....	120
8.1.6	Category Table .....	121
8.1.7	Subcategory Table.....	121
8.1.8	Suppliers Table .....	121
8.1.9	SupplierDeliveries Table .....	122
8.1.10	Customer Table .....	122
8.1.11	ContactDetails Table .....	122
8.1.12	Orders Table.....	123
8.1.13	ShoppingCart Table .....	123
8.1.14	AttributeCategory Table.....	124
8.1.15	Attribute Table .....	124
8.1.16	OrderItem Table .....	124
8.1.17	ShopConfig Table .....	125
8.1.18	PromotionDiscounts Table .....	125
8.1.19	ErrorLog Table.....	125
8.1.20	TransactionLog Table .....	125
8.2	Stored Procedures .....	126
8.3	Security Mechanisms .....	126
8.4	Performance .....	127
8.5	Network Protocol .....	127
8.6	Pseudo code for Key Algorithms .....	127

8.6.1	Password Hashing .....	127
8.6.2	Calculating cart totals.....	129
8.7	Global Control Flow .....	130
8.7.1	Execution Order .....	130
8.7.2	Time Dependency .....	130
8.7.3	Concurrency .....	130
9.	Interface Design.....	131
9.1	Web Site Design .....	131
9.1.1	Web Site Overview .....	131
9.1.2	Target Audience .....	131
9.1.3	Site Style and Appearance.....	132
7.2.1	Promotion.....	133
9.1.4	Idea Map .....	133
9.1.5	Layout Sketches .....	135
9.1.6	Navigation Path Diagrams.....	138
9.1.7	Personas .....	140
9.1.8	Accessibility.....	143
9.1.9	Concept Sketches .....	143
9.2	Administrative Interface.....	150
9.2.1	Layout Sketches .....	150
9.2.2	Concept Sketches .....	151
Appendix C.	.....	156
Realisation .....	.....	157
System Overview .....	.....	157
Source Code .....	.....	157
MySQL Implementation Script .....	.....	157
Screen Shots .....	.....	184
Appendix D .....	.....	199
Overview .....	.....	200
Introduction to testing .....	.....	200
Objectives and tasks.....	.....	200
Scope of the testing.....	.....	200
Testing strategy.....	.....	200
Reference Materials .....	.....	201

Requirements Document.....	201
Design Document .....	201
Specific Testing Areas .....	201
Static Testing .....	201
Content Checking .....	201
Browser Compatibility.....	202
Test Browsing .....	206
Browsing the Site.....	206
Functional Testing .....	206
Browser-Page Tests .....	206
Non-Functional Testing .....	206
Performance .....	206
How it will be tested .....	206
Security.....	206
How it will be tested .....	207
Component Testing.....	207
Test Strategy .....	207
Conclusion.....	245
Appendix 4 .....	248
Test hasher .....	248
Accuracy.....	249
Consistency.....	249
Spelling.....	250
Accessibility .....	250
Other.....	250
Appendix E.....	251
Installation.....	252
System Requirements .....	252
Installation .....	252
Database .....	252
Application.....	252
Solutions to possible compatibility issues.....	252
Products .....	253
Add a department:.....	254

Edit a department:	254
Delete a department:	255
View department categories:	255
Add a category:	256
Delete a category:	256
Edit a category:	256
View products:	257
Add a product:	258
Edit a product:	259
Edit basic details of product:	260
Edit product locations:	261
Edit product images:	261
Stock Control	262
Products and their attributes	264
Add an attribute:	264
Edit an attribute:	265
View attribute values:	265
Edit attribute value:	266
Shopping Carts	267
Counting old shopping carts:	268
Delete old shopping carts:	268
Customers	268
Add new customer:	269
View customer details:	269
Edit customer details:	270
Delete a customer:	270
Suppliers	271
Add new supplier:	271
View supplier details:	272
Edit supplier details:	272
Delete a supplier:	272
View supplier deliveries:	273
Add a supplier delivery:	273
Edit a supplier delivery:	273

Edit a supplier delivery:	274
Orders .....	275
View and Edit order details:	279
Order Processing status:	279
Shipping.....	283
Add a new shipping type:.....	283
Update or delete shipping types: .....	284
Configuration.....	285
Update sysadmin password:.....	285
Update products displayed per page:.....	286
Access error logs:.....	287
Maintaining eMart .....	288
Appendix F .....	289
Summary of Changes .....	290
Added Use Cases:.....	290
Redundant Use Cases: .....	290
Modified Use Cases:.....	291
Appendix G .....	292
Appendix H .....	321

## Members and Roles

The team for this project is Kisacasa. It consists of Kirsty Williams, Sara Chattun, Carl Blakemore and Samantha Kviecinskas. All individual contributions are detailed in the Gantt Charts throughout the documentation. Additionally, a summary of key contributions can be found in Figure 1.1.

Unfortunately, at the commencement of the design phase Carl Blakemore made the decision to leave university. Hence forth, for the duration of phases two and three Kisacasa had only three team members.

*Figure 1.1*

	Requirements	Design	Implementation and Reports
Kirsty Williams	<ul style="list-style-type: none"> <li>• Gantt chart</li> <li>• Project Statement</li> <li>• Feasibility study</li> <li>• Actors and goals</li> <li>• Use case descriptions</li> <li>• Prioritisation and success criteria of requirements</li> </ul>	<ul style="list-style-type: none"> <li>• Gantt chart</li> <li>• System architecture model</li> <li>• Transaction Matrix</li> <li>• SQL base tables</li> <li>• Stored Procedures</li> <li>• Performance</li> <li>• Network Protocol</li> <li>• Algorithms</li> <li>• Global Control Flow</li> <li>• User interface design</li> </ul>	<ul style="list-style-type: none"> <li>• Gantt chart</li> <li>• Implementation of system</li> <li>• Installation instructions</li> <li>• Summary of Changes</li> <li>• Realisation</li> <li>• Report – What was produced, Evaluation</li> </ul>
Sara Chattun	<ul style="list-style-type: none"> <li>• Transaction requirements</li> <li>• Functional requirements</li> <li>• Use case diagrams</li> </ul>	<ul style="list-style-type: none"> <li>• ER diagram</li> <li>• Class diagram</li> <li>• Data dictionary</li> <li>• Table mapping</li> <li>• Sample data</li> </ul>	<ul style="list-style-type: none"> <li>• User guide</li> <li>• Testing</li> <li>• Realisation - Screen shots</li> <li>• Report- Evaluation, extensions</li> </ul>
Carl Blakemore	<ul style="list-style-type: none"> <li>• User views</li> <li>• System scope and boundaries</li> <li>• System specification</li> </ul>	<ul style="list-style-type: none"> <li>• ER diagram planning</li> <li>• Interaction diagrams</li> </ul>	N/A
Samantha Kviecinskas	<ul style="list-style-type: none"> <li>• Mission statement and objectives</li> <li>• Non Functional requirements</li> <li>• Domain research</li> </ul>	<ul style="list-style-type: none"> <li>• Business rules</li> <li>• Data flow diagrams</li> <li>• Security mechanisms</li> </ul>	<ul style="list-style-type: none"> <li>• Report - application description, professional issues, Evaluation</li> <li>• Testing</li> </ul>

Organisation was maintained throughout the project by regular meetings, the details of which can be found in the meetings and agendas. Keeping the minutes and agendas organised aided in the progress

of the project as it enabled the group to verify that targets were being met, review the work completed by each member and address any problems that may have occurred.

Meetings were held weekly except during the Easter break where no formal meetings were held. Each Kisacasa member attended and contributed to meetings fairly.

## Application Description

### Application Domain

The mission statement for the project is

“The purpose of eMart’s system is to provide a complete and comprehensive database system with an easy to use interface for an online retailer of grocery goods. The database will collect, store, manage and control access to the data for stock management, customer orders, HR management, and company finances.”

Expanding on the mission statement, the project involved creating a web interface incorporating a database for customers to browse and purchase products online from eMart. The system was also extended to all eMart employees to control stock management and deal with customer orders.

### Types of User

The system has two types of user; customers and eMart employees. There are two interfaces to the system, each of which is tailored to the needs of the relevant user. The front-end, “shop” interface is used by the customers and is updated by changes made in the database via the back-end by the staff.

#### Customers

During the design phase two target customers were identified; Alice and Mark. Generally our target audience is middle class men or women who are eco conscious and care about local produce.

They will use the system for convenience. Online grocery is becoming a very popular means of grocery shopping and is forecast to become increasingly more popular over time.

#### Staff

Staff will use the system to process and manage the storefront and database, including orders and inventory.

### Typical Queries

Both the requirements and design phase illustrated what queries will be implemented within the system.

The interviews with the MD and a current store manager of eMart identified that queries needed were to see the delivery in takes, adjust stock levels and edit orders.

### Typical Reports

During the requirements phase interviews were held with the MD of eMart who stated that he would like to be able to view reports on “stock, accounts and customer orders”.

An interview was also held with a current Store Manager who stated that he needs to be able to create reports on “stock and orders especially”. Also “employee reports” to see what employees are in each day.

## System Achievements

The final web application developed was a high quality ecommerce solution that is fully functionally and able to process credit card transactions. A live demo site is viewable at the domain [www.kisacasa.com](http://www.kisacasa.com). The Administrative back-end can be accessed with the following login credentials:

Username: sysadmin

Password: Password2012

A Windows live email account has been set up in order to receive the various email communication prompted by the system. The email provider can be accessed via [www.hotmail.co.uk](http://www.hotmail.co.uk) with the following login credentials:

Email address: [kisacasa@hotmail.co.uk](mailto:kisacasa@hotmail.co.uk)

Password: Password2012

### Conformance with Standards

A web application can sometimes be created very quickly determined on certain initial requirements, but if not properly architected, it can become difficult, if not, impossible to alter. The system produced must be scalable, i.e. open to future expansion. For this reason it was important to use standard coding conventions in order to help future developers better comprehend and make use of the source code. Some of the coding conventions adopted are as follows:

- Class names and methods were written using Pascal casing (e.g. TheClass).
- Public class variable names follow the same standard although are prepended with the letter ‘m’ (e.g. \$mPublicVariable).
- Private class variable names follow the same standard as the public class variable names but are prepended with an underscore (e.g. \$\_mPrivateVariable).
- Method parameters use camel casing, upper-case for the first letter of every word except the first one (e.g. \$theParameter).

With the help of a consistent, comprehensible, custom error-handling object and a powerful relational database, data integrity is ensured and any errors that arise in the system are capably handled without complete system failure. Errors are output to a text file, errors\_log.txt located in the logs directory.

### Code Structure

When building the system, the decision was made to implement a flexible architecture composed of “pluggable” components. This enabled the adding of new features, such as the shopping cart summary, departments list, customer account/login summary by coding them as separate components and plugging them into the existing application.

In keeping with the three layer architecture of the system the source code developed is separated into a presentation, business and data layer. The presentation layer contains the UI elements of the system and includes all the logic that manages the interaction between the user and the system. This layer is composed of the dynamic web pages (PHP and HTML template files). The business layer receives requests from the presentation layer and returns a result to the presentation layer depending on the

business logic it contains. Almost any event that happens in the presentation layer usually results in the business layer being invoked, except events that can be handled by the presentation layer, for example, simple input validation. For example, if the eMart employee clicks to retrieve the 20 most recent orders in the database, the presentation layer invokes the business tier with a message, ‘Please send me all orders that match this query’. Usually, the business layer needs to call a stored procedure in the data layer for information to be able to respond to the presentation layer’s request. The data layer is responsible for managing the application’s data and sending it to the business layer when requested.

The decision to implement the system using the three layer architecture was made with the view that the overall system is resistant to changes. If for instance something was to be altered in one of the layers, the other layers would generally remain unaffected. For example, if for some reason eMart decide to migrate the back-end of the system, i.e. the database, from MySQL to Oracle, only the data layer and possibly a little of the business layer would need to be updated. This choice is most advantageous to eMart whose company is still in growth and has the potential for an expansion in database needs.

### Tools

The system implemented made use of a template engine, Smarty, which offered a framework for separating the presentation logic from the static HTML templates. Smarty is, to date, the most powerful template engine for PHP. Ideally, it would have proved advantageous when assigning coding ‘jobs’ to members of the team (were there enough bodies), allowing one person to work on the HTML template designs independently from the PHP programmer. This is because the PHP logic can be altered without needing to change the template files, and vice versa.<sup>1</sup>

Communication between the PHP code and the MySQL database was achieved via the PHP Data Objects (PDO) extension tool. PDO offers a uniform way of accessing a variety of data sources (i.e. MySQL, MS SQL, etc). By using PDO, the system’s portability and flexibility is increased since if the database changes, the effects on the code to access the data are kept to a minimum, with, in most cases only a change to the connection string necessary.

### Security

Only customers with registered accounts will be able to place orders. All orders require valid credit card information which is encrypted upon adding to the database. Customer credit card transactions are processed by an online payment gateway, authorize.net. This is fully functional in the product being submitted and processes dummy payments. The algorithm used for encryption of the sensitive information is the Rijndael symmetric key algorithm. A secret key constant is defined along with an initialisation vector. An encryption function encrypts the plain text received as a parameter and returns the result in hexadecimal format. A subsequent decryption function is also used that decrypts the encrypted string.<sup>2</sup>

The system has a SecureCard class that represents an instance of a customer credit card. This class can be supplied with credit card information, which is then accessible in encrypted format. This class can also take encrypted credit card data and supply access to the decrypted information.

All passwords used throughout the system are stored using the sha1 hashing function. It takes as parameters a password string and a prefix and returns a hashed result ready for use. Since a hashing

---

<sup>1</sup> (crash\_course)

<sup>2</sup> (system.security.cryptography.rijndael.aspx) (wsdindex.html) (Advanced\_Encryption\_Standard)

function is irreversible the original password is stored should the event arise that a password needs to be recovered.

Access to the administrative back-end of the system requires authentication. This is done via a PHP login mechanism. Upon successful login, users have access to a GUI control panel that allows them to maintain the ‘shop’, front-end of the system and manage the MySQL database without having to write complex SQL queries. Once logged into the administrative area if the user returns to the storefront without logging out of the back-end they are faced with added extras that allow them to edit individual departments, categories, and products directly via redirection to the appropriate admin page.

### Session Handling

Session handling is a PHP feature that enabled the tracking of variables specific to a particular website visitor. While a customer browses the online store, the session variables are retained by the web server and associated to a unique visitor identifier (stored in the customer’s browser as a cookie). When implementing the paging functionality on the catalog, before requesting the products, the database is queried for the total number of products that are going to be returned, so you can show the customer how many pages of products are available. This number will be saved in the session, so if the customer browses the various pages on the catalog, the database wouldn’t need to be queried multiple times for the number of products on subsequent calls. This adds to the performance of the website. Session variables are used in a similar fashion for the continue shopping links in the shopping cart.

### Search Engine Optimisation

Keyword rich URLs have been implemented with the use of various rewrite rules in the htaccess file and a function in the Link class that uses regular expressions to replace the characters that are not wanted in the URL, such as whitespace, with dashes. For example, the function would transform the string ‘Fresh Merseyside Butter’ with a URL friendly string such as ‘Fresh-Merseyside-Butter’. This is an added extra that provides a great means of search engine optimisation.

### Forgotten Passwords

If a customer and/or back-end user of the system forget their password they can be sent a reminder by clicking on the ‘forgotten password’ link underneath the relevant login form. In both cases along with containing the current password, the email sent recommends that the password gets changed immediately and details the steps required to do this.

### Administering Products

Removing products in the system is not straightforward. A product can either be unassigned from a category by removing the record from the product categories table or remove the product from the product table. Due to the hierarchy of the product catalog, it is necessary to make sure that there are no products that don’t belong to any category. There is a remove from category button that removes the product from a single category and a remove from catalog button that removes the product from the whole database. If the product belongs to multiple categories, the user can use the remove from category button. However, if the product belongs to only a single category, only the remove from catalog button will be available.

### Shopping Cart

The system has a custom built shopping cart. The benefits of this by far outweighed the alternative third party cart facilities, like those supplied by PayPal. Since the project set out to build a scalable

system the ability to save order details was most advantageous. It opens up the prospect of adding features such as a product ranking system which would not be possible were we not to possess any data. The shopping cart has a ‘save for later’ facility. This facility allows a customer to order only a subset of the products in their shopping cart and save the other items ready to be purchased at a later date. When a product is saved for later, it is moved to a separate list of the shopping cart and is not included in the order when the customer checks out.

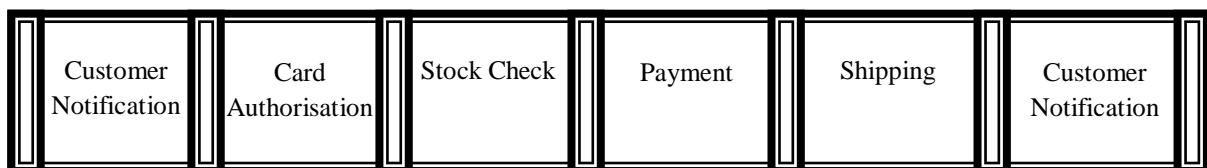
The carts option in the administrative back-end allows the deletion of old shopping cart records in the database. This is beneficial if the shopping cart table was to grow very large. At present shopping carts instances are stored in the customer’s browser for seven days. As such shopping carts older than seven days old can be assumed to be obsolete and can be removed from the database.

### **Customer Account Functionality**

Customer’s have the ability to update and change their details via the storefront upon successful login.

### **Order Process**

Implemented is a set of customer order-processing stages that deals with credit card authorisation (via authorize.net), stock checking, despatching orders, taking payments, and email notification. The order-processing functionality allows the tracking of orders at every stage of the process and provides auditing information that can be referenced at any point in time.



### **Input Validation**

The system includes a number of regular expressions that validate user input. For example, the script ensures that a user registers with a valid syntactical email address.

## **Evaluation**

### **Strengths**

eMart now has a fully functioning e-commerce solution. A customer can browse products, register and place an order. This order will be processed fully (see Testing Document for full details), credit card details are encrypted (see ‘What was produced’) and sent securely to the payment gateway. The customer will receive updates by email when their order has been received and when it has been dispatched. Another useful aspect of the website is that customers can save carts for a later date (see ‘What was produced’ for full technical details),

The web page has a consistent layout that is easy to use and is aesthetically pleasing; which is aided by the use of templates. The high usability of the system means that customers who are maybe inexperienced users of ICT might feel confident to try the eMart website, broadening the potential customer base of the company. The contact form is an excellent tool for customers who wish to get in touch with the company, it is simple and self-explanatory to use.

The company’s integrity is improved by encryption. Encryption is used to secure passwords and payment details. This also ensures the company complies with the Data Protection Act. The website also complies with the BCS Code of Conduct (see Professional Issues).

The web page incorporates a Twitter feed. The use of Twitter acts as a way of advertising as when customers who ‘follow’ eMart on Twitter ‘tweet’ eMart, their ‘followers’ can view this post.

The back end of the system is also fully functional. It allows staff to edit all aspects of the website as well as add suppliers and deliveries to the warehouse.

### **Weaknesses and their Improvements**

There are several weaknesses in the final product which could be improved.

Firstly, when an order is cancelled by the company the customer is not notified. A way to improve this would be to automatically send an email to the customer who placed the order notifying them of the cancellation.

When a user is using the order search filters, at present when a user enters the wrong format for the search it outputs “no order found”. This could be improved by a more relevant message appearing telling the user it is in the wrong format.

The demo feedback suggested that the system could have been improved firstly by including a confirmation check when a user deletes something from the system. Another simple improvement would be to use regular expressions when a user enters a post code.

### **Working as a team**

The team worked excellently together throughout the project. Every team member acted in an efficient manner and accepted constructive criticism from other team members when necessary. Every member made an effort to attend all meetings. Additionally, all members participated in informal discussions which occurred via a private Facebook group, email, and BaseCampHQ. A shared Dropbox (i.e. an online directory) facility was managed effectively by all members which aided sound version control of various project files. The loss of a team member so early on could have easily caused problems within the group but remaining parties pulled together effectively and provided a final submission to be proud of.

### **Extensions**

As with all software development projects, there are many ways in which add-ons can be incorporated into the application to keep the system current as new and exciting technological advances are made.

A primary feature that would prove effective in the current version of the application is a full indexed product search. The search would enable a user to enter keywords in a search box to find the products they wish to locate more efficiently.

Requirements derived from initial discussions with the client included the option of a payroll system however, after much deliberation with the client this was deemed unnecessary and thus was removed from the product design. The same was decided about an online rota module to the application which would have enabled eMart employees to manage and view shift rotas. These extensions would prove an effective extension in providing a more complete business system.

Another potential extension would be a ‘store locator.’ This would enable users to locate their nearest store on entering their postcode and see the distance. It would also be possible to further extend this by incorporating Google Maps to show directions from a customer’s location to the store.

Statistics are a big part of business; they show trends of products which companies can use to order the relevant stock. Therefore, a report creating facility would be a good improvement for eMart. Reports could first of all be produced seasonally and eventually, when the business grows to be produced weekly.

A lot of ecommerce sites have a product review section; this is where customers can give ratings and personal reviews of products they have bought. A ranking system could prove a very worthwhile implementation. This would be a feature that eMart could really utilise. As they are only a small business at present, eMart will rely on old customers to return and bring in loyalty from new customers.

In order to keep customers interested, promotional offers will be introduced. This will require a new table in the database containing the original and discounted price of the product.

A useful feature to be implemented would be a delivery slot booking system. This would allow customers to choose a delivery time and day. Delivery charges could then be automatically calculated depending on the time and day i.e. in peak time or not.

Eventually, eMart will create an automatic stock control system. That is, when products are purchased, rather than the warehouse needing to confirm that the products are in stock, there is a system which automatically decrements the product from the database. When a product reaches a certain stock level, there will be a warning sent to the warehouse to reorder that item.

Finally, a key improvement to the system would be to implement multiple levels of authorisation to the administrative back-end of the system. As in all businesses, different employees have access to different amounts of data. Therefore, assigning different privileges to employees (using their admin area login name) would be a vital aspect to addition to the system.

## Professional Issues

Professional issues were a big aspect to inspect during the project. It is extremely important for eMart that the system follows the BCS Code of Conduct, as well as the Data Protection Act.

After analysing the BCS Code of Conduct it became clear which codes applied to the project:

### Public Interest

*“a. Have due regard for public health, privacy, security and wellbeing of others and the environment”*

Privacy and security were the main issues that needed to be ensured. SHA1 encryption has been used to keep personal details safe.

*“d. Promote equal access”*

The website is accessible by all who have the correct equipment. It does not favour any race, religion or disability.

### Professional Competence and Integrity

*“a. Only undertake to do work or provide a service that is within your professional competence”*

Everything included in the implementation was not out of the groups reach and knowledge.

*“b. NOT claim any level of competence that you do not possess”*

Any code that has been taken from another source has been referenced sufficiently.

*“c. Develop your professional knowledge, skills and competence on a continuing basis”*

Each member of the team has had their knowledge skills and competence developed throughout this project in several areas.

*“d. Ensure that you have the knowledge and understand of Legislation and that you comply with such Legislation, in carrying out your professional responsibilities”*

This was achieved by researching and reviewing the BCS Code of Conduct prior to implementation and keeping it in mind throughout.

*“e. Respect and value alternative viewpoints”*

Throughout implementation the design was constantly shown to friends and family to gather opinions on how it was looking so far; each alternative opinion was taken on board, and at times altered implementation.

## Duty to Relevant Authority

*“b. Seek to avoid any situation that may give arise to a conflict between you and your relevant authority”*

Kisacasa insured that all implementation was completed with regards to authorities and no laws were broken to their knowledge.

*“d. NOT disclose or authorize to be disclosed, or use for personal gain or to benefit third party confidential information except with permissions by authorities”*

All of the sensitive data of customers and staff in the eMart system needs to be kept within the company safely to comply with the Data Protection Act.

*“e. NOT misrepresent or withhold information on the performance of products, systems or services or take advantage of the lack of relevant knowledge or inexperience of others”*

All information on the website is correct at time of implementation.

## Duty to the Profession

*“a. Accept your personal duty to uphold the reputation of the profession and not take any action which could bring the profession into disrepute”*

It was extremely important that the system was implemented in a positive way of eMart protecting their reputation. It was also important to take action such as ensuring all personal details are kept secure as otherwise this could lead eMart into trouble

*“c. Uphold the reputation and good standing of the BCS”*

Following the standards of the BCS throughout the project has insured that this standard has stood.

*“f. Encourage and support fellow members in their professional development”*

Throughout the project the group has worked well together and at times, when one member of the team has struggled on an aspect, another team member has aided them by offering their knowledge to support the other in the completion of their contribution.<sup>3</sup>

## Items not complied with

Items not complied with signify the items that were not relevant to the project. All items relevant to the project were complied with efficiently.

---

<sup>3</sup> (6030)

## Bibliography

- (n.d.). Retrieved 02 14, 2012, from hfgilbert.com: <http://www.hfgilbert.com/>
6030. (n.d.). Retrieved 05 03, 2012, from www.bcs.org: <http://www.bcs.org/category/6030>
- Advanced\_Encryption\_Standard*. (n.d.). Retrieved 05 02, 2012, from en.wikipedia.org: [http://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](http://en.wikipedia.org/wiki/Advanced_Encryption_Standard)
- best-screen-size*. (n.d.). Retrieved 02 15, 2012, from www.hobo-web.co.uk: <http://www.hobo-web.co.uk/best-screen-size/>
- Connolly, T., & Begg, C. (2004). *Database Solutions: A step-by-step guide to building databases*. 2nd ed. Pearson.
- crash\_course*. (n.d.). Retrieved 04 16, 2012, from Smarty: [www.smarty.net/crash\\_course](http://www.smarty.net/crash_course)
- data\_protection*. (n.d.). Retrieved 02 10, 2012, from www.ico.gov.uk: [http://www.ico.gov.uk/for\\_organisations/data\\_protection.aspx](http://www.ico.gov.uk/for_organisations/data_protection.aspx)
- Faulkner, C. (1998). *The Essence of Human Computer Interaction*. Prentice Hall.
- FundamentalsOfRelationalDatabaseDesign*. (n.d.). Retrieved 03 12, 2012, from www.deeptraining.com: <http://www.deeptraining.com/litwin/dbdesign/FundamentalsOfRelationalDatabaseDesign.aspx>
- Nixon, R. (2009). *Learning PHP, MySQL & Javascript*. O'Reilly.
- password-hashing.html*. (n.d.). Retrieved 02 29, 2012, from phpsec.org: <http://phpsec.org/articles/2005/password-hashing.html>
- Preece, Rogers, & Sharp. (2002). *Interaction Design - beyond human-computer interaction*. Wiley.
- Sommerville, I. (2011). *Software Engineering* 9th ed. Pearson.
- suomi, r. *project e-society: building bricks*.
- system.security.cryptography.rijndael.aspx*. (n.d.). Retrieved 05 02, 2012, from msdn.microsoft.com: <http://msdn.microsoft.com/en-us/library/system.security.cryptography.rijndael.aspx>
- terms-and-conditionscontainer*. (n.d.). Retrieved 03 12, 2012, from asda.com: <http://groceries.asda.com/asda-estore/common/static/terms-and-conditionscontainer.jsp>
- termsandconditionsGeneral*. (n.d.). Retrieved 03 12, 2012, from www.tesco.com: <http://www.tesco.com/termsandconditions/termsconditionsGeneral.htm>
- Williams, H. E., & Lane, D. (2004). *Web Database Applications with PHP and MySQL*. O'Reilly.
- wsdindex.html*. (n.d.). Retrieved 05 02, 2012, from csrc.nist.gov: <http://csrc.nist.gov/archive/aes/rijndael/wsdindex.html>

## Appendix

## Appendix A

# REQUIREMENTS DOCUMENT

---

# 1. Introduction

## 1.1 Project Statement

Today, it is increasingly important for organisations to achieve complete synchronisation of their computerised processes. To do so, it is necessary to have database systems that can operate over a network. eMart Management System is a comprehensive database system that stretches across the entire company. The system will consist of a number of comprehensive modules, each targeting a specific area of the organisation's management issues. The system will cover Stock Management, HR Management and eCommerce Management. The Stock Management will pay particular concern to the maintenance of the organisations inventory. It will contain features such as automatic notification of low stock levels etc. The HR Management facility will satisfy the organisations administration of personnel data and will facilitate the undemanding completion of payroll. Finally, the eCommerce Management capability will incorporate a fully-functional web site that enables the secure transaction of consumer orders.

Along with providing the client with a GUI front end for accessing the system on site, personnel with administrative access will be able to access parts of the system remotely via a web based administration tool. Such facilities will allow organisation personnel to log in and inspect their personal rotas in addition to several other services. Data stored can be accessed, manipulated and retrieved very easily. The interface designed will be made user friendly. The result of query functions will be properly optimised to be very quick and efficient.

## 1.2 Feasibility Study

### Implementation:

Given that the programming knowledge of the project team is currently confined to Java, implementing the GUI in this language is the most realistic choice. Alternatives such as C# exist although would not prove practical for the purpose of this particular project.

### Technical:

The system will not require any sophisticated training. It will be designed with key focus on the user interface. The application will be easily employed even by the most novice of user. An introductory training period of no more than 30 minutes will be adequate enough to train the user. The system will feature predominantly menu-driven and (icon) button interaction methods similar to many COTS software on the market today. A help facility can be made available without any trouble.

### Operational:

All support facilities necessary for the ongoing operation and maintenance of the system can be made readily available. A user manual defining all possible obscurities within the system can be made available without difficulty.

### Economic:

Type	Potential Costs	Potential Benefits
Quantitative	Hardware/software upgrades Fully-burdened cost of labour Support costs for the application Expected operational costs	Reduced operating costs Reduced personnel costs from a reduction in employees Increased revenue from additional sales of

	Training costs for users to learn the application	the organisations products Reduced losses due to human error
<b>Qualitative</b>	Increased employee dissatisfaction from fear of change Negative public perception from layoffs as the result of automation	Improved decisions as the result of access to accurate and timely information Increased employee satisfaction from the availability of online shift information Positive public perception that the organisation is an innovator Increased revenue from widening of demographic (more accessible to the unable)

### 1.3 Data Protection and eMart

- eMart will hold and use sensitive information on customers, employees and suppliers.
- According to the Data Protection Act (1998), eMart must keep this information safe, correct and, where necessary, up to date.
- eMart must not store the data longer than necessary. For example, if an employee leaves the company, data on that employee must not be kept for a prolonged period of time.
- eMart must process data in accordance with the DPA. That is, that they must not process data in such a way that it can be interpreted as misleading.
- eMart must ensure that precautions are taken to ensure that data is not processed unlawfully, for example, revealing data to third parties.
- eMart must also ensure that measures are taken to avoid accidental loss or destruction of, or damage to, personal data.
- eMart will not transfer data to any countries which would have inadequate levels of protection rights of data subjects in relation to the processing of personal data.
- If a request is made, eMart will allow personal data to be accessed to the party which it is concerned with. If such a request is made, eMart will respond to a valid subject request within 40 calendar days of receiving it.

## 2. Database Planning

### 2.1 Mission Statement

The purpose of eMart's system is to provide a complete and comprehensive database system with an easy to use interface for an online retailer of grocery goods. The database will collect, store, manage and control access to the data for stock management, customer orders, HR management, and company finances.

### 2.2 Mission Objectives

- To maintain (insert, update, and delete) data on departments.
- To maintain (insert, update, and delete) data on stock.
- To maintain (insert, update, and delete) data on employees.
- To maintain (insert, update, and delete) data on orders.
- To maintain (insert, update, and delete) data on order vans.
- To maintain (insert, update, and delete) data on user login.

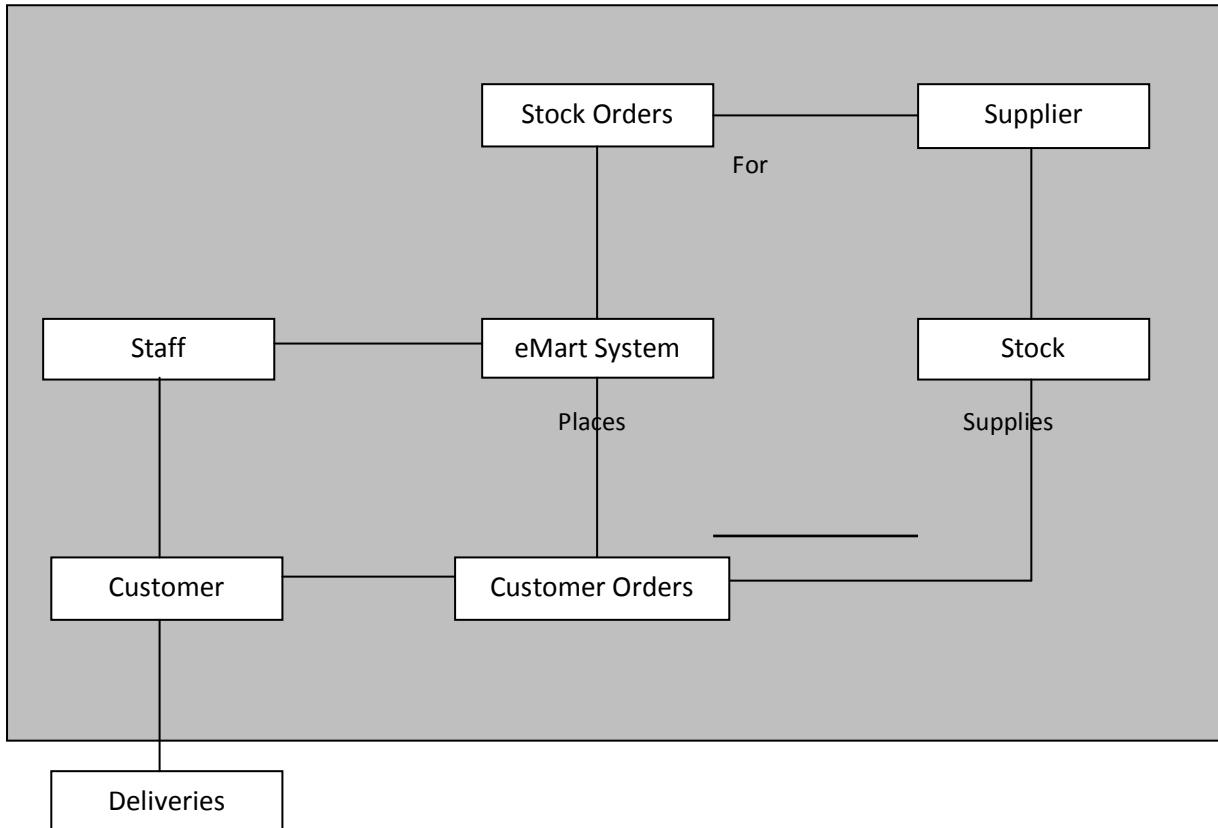
- To maintain (insert, update, and delete) data on employee contact details.
  - To maintain (insert, update, and delete) data on emergency contacts.
  - To maintain (insert, update, and delete) data on customers.
  - To maintain (insert, update, and delete) data on customer accounts.
  - To maintain (insert, update, and delete) data on accounts.
  - To maintain (insert, update, and delete) data on employee accounts.
  - To maintain (insert, update, and delete) data on supplier accounts.
  - To maintain (insert, update, and delete) data on suppliers.
  - To maintain (insert, update, and delete) data on supplier deliveries.
  - To maintain (insert, update, and delete) data on supplier drivers.
- 
- To perform searches on departments.
  - To perform searches on stock.
  - To perform searches on employees.
  - To perform searches on orders.
  - To perform searches on order vans.
  - To perform searches on user logins.
  - To perform searches on employee contact details.
  - To perform searches on emergency contacts.
  - To perform searches on customers.
  - To perform searches on customer accounts.
  - To perform searches on accounts.
  - To perform searches on employee accounts.
  - To perform searches on supplier accounts.
  - To perform searches on suppliers.
  - To perform searches on supplier deliveries.
  - To perform searches on supplier drivers.
- 
- To track the status of stock.
  - To track the status of orders.
  - To track the status of supplier deliveries.
- 
- To report on stock.
  - To report on orders.
  - To report on employees.
  - To report on supplier deliveries.
  - To report on database access (illegal access)
  - To report on system errors
  - To report on customers
  - To report on suppliers
- 
- To show employee information

### 3. System Definition

#### 3.1 System Scope and Boundary

After careful analysis of the data collected so far the following diagram shows the system boundaries of the main data types and how this data is related.

Boundaries of the *eMart* database system:



#### 3.2 User Views

##### Organisation owner

- To maintain (insert, update, and delete) data on accounts.
- To perform searches on accounts.

##### System Administrator

- To maintain (insert, update, and delete) data on the system.
- To maintain (insert, update, and delete) data on user login.
- To perform searches on user login.
- To perform searches on customers.
- To report on database access (illegal access).
- To report on system errors.

##### Manager

- To maintain (insert, update, and delete) data on stock.

- To maintain (insert, update, and delete) data on employees.
- To maintain (insert, update, and delete) data on accounts.
- To perform searches on employees.
- To perform searches on customers.
- To perform searches on accounts.
- To report on stock.
- To report on orders.
- To report on employees.
- To report on customers.
- To report on suppliers.
- To show employee information.

#### **Department Managers**

- To maintain (insert, update, and delete) data on departments.
- To maintain (insert, update, and delete) data on stock.
- To maintain (insert, update, and delete) data on employees.
- To perform searches on departments.
- To perform searches on employees.
- To report on stock.
- To report on employees.
- To show employee information.

#### **Supervisors**

- To maintain (insert, update, and delete) data on orders.
- To perform searches on stock.
- To perform searches on orders.
- To track the status of stock.
- To track the status of orders.
- To report on stock.
- To report on orders.

#### **Merchandiser**

- To maintain (insert, update, and delete) data on suppliers.
- To perform searches on stock.
- To perform searches on suppliers.
- To track the status of stock.
- To track the status of supplier deliveries.
- To report on stock.
- To report on suppliers.

#### **Assistants**

- To maintain (insert, update, and delete) data on orders.
- To perform searches on stock.
- To perform searches on orders.
- To track the status of stock.
- To track the status of orders.

#### **Customers**

- To maintain (insert, update, and delete) data on orders.

- To maintain (insert, update, and delete) data on user login.
- To perform searches on stock.
- To perform searches on orders.
- To report on stock.
- To report on orders.
- To report on system errors

## 4. Requirements Collection

### 4.1 Interviews

#### Interview with the Managing Director of eMart

*What is the purpose of your company?*

eMart is currently a shop based convenience store that would like to expand into an online shop for customers. We are opening a new warehouse for the website creating new employment opportunities, and hopefully in the future can expand to more warehouses throughout the country.

*Why do you feel you need a database?*

We need a database to predominantly manage the stock in and out of the warehouse. We want to store employee details on the database as well as customer information and orders.

*Can you give me an idea of the type of things you'd like to keep track of in the database?*

Within the warehouse we would like to keep track of the stock levels, the employees working in each department, their details and also the customer orders. Also, we need it to hold all of our accounts information for customers, suppliers and employees.

*What type of reports do you generate?*

Generally, I like to generate stock and account reports within our shops. However I think customer order reports are also now necessary with the new database.

*What type of things do you like to keep track of?*

I think it is important to keep track of stock coming in from deliveries, and the stock going out for customer orders. It is also important to keep an eye of the accounts.

*Do we need to be aware of any security issues?*

It is of the upmost important personal customer details are kept secure, and only those who need to view them can. The accounts should also be kept secure.

*Generally, as a company, what service do you provide for your customers?*

We aim to give our shoppers the best prices for the highest quality products.

#### Interview with the Store Manager of eMart

*What is your job description?*

I am the manager of an eMart store in Essex. I employ 76 people and my job is to ensure the shop is run efficiently from day to day. I have to make sure we are receiving enough stock to meet demand. I make sure the stock is available ASAP and the team is working efficiently together to provide the best service to our customers.

*What kind of tasks do you perform on a day to day basis?*

We receive a daily delivery so I update the stock levels manually every morning. I then assign employees to their jobs and ensure they all know what their day entails. I manage the accounts by hand, and place the order for the delivery for the following day.

*Why do you feel you need a database?*

eMart would improve greatly with the introduction of a database as the stock management in stores at the minute is hardly visible. The current system is very time consuming, and not productive at all. With an online database stock levels could be checked automatically and the merchandiser would know what to reorder without me having to tell them.

*Can you give me an idea of the type of things you'd like to keep track of in the database?*

The stock levels and the movement of stock. I would also like to view each department to see which employees are assigned there. It would definitely make the company as a whole more organized.

*What type of reports do you create?*

The most important reports I create are on the stock and orders. However, the account reports are extremely important to monitor the movement of money within the shop. I also generate employee reports to help me assign tasks to the employees each day.

*What do you think about eMart expanding to online shopping?*

I think this is a brilliant idea and very exciting for the company. It brings the company in touch with the 21<sup>st</sup> century and offers our services to potential customers who may have never heard of us before.

*What type of data do you need to hold on employees?*

Personal details mainly, with an emergency contact number too. Also their banks details for paying in their wages which need to be kept secure. I also want to identify each employee with their own unique ID number so we are able to search for them quickly.

*What sort of things do you do with the data on employees?*

Deleting and adding employees to the database will be necessary. I need to be able to change details for when they move address etc. Also I want to print reports that list things like all employee names, all departments etc.

*What transactions run frequently on the database?*

When customers want to search for a specific product they will search through the database to see if we have it in stock. Also with quantities of stock, us as employees will run transactions to find out ideal quantities of stock and how many is currently in the warehouse.

*What transactions are critical to the operation of the business?*

Definitely the stock control transactions. Without any stock we wouldn't be able to offer a service to our customers; and therefore not operate as a business at all.

*When do critical systems run?*

All the time,every day.

*What time of the day has the highest workload period for the critical transactions?*

With our shops we tend to get busier as the day goes on, however I expect with internet purchases the peak time will be early evening for customer orders.

*What type of security do you want for the database system?*

We want the data we will store to be as secure as possible. It is important that it doesn't become available to any third parties without the proper authorisation. We would like passwords to be used to limit this as much as humanly possible.

*Is there any sensitive information that should only be accessed by certain employees?*

The accounts should only be accessed by the accounts employees and the store manager. It is extremely important to keep this information hidden from everybody else

Interview with a regular shop customer of eMart

*How often do you shop at eMart?*

eMart is the only convenience store I visit, their quality and prices cannot be beaten in the local area.

*Would you use an online shopping service from eMart?*

Yes definitely, all of my friends shop regularly online for their weekly shop with other competitive companies and it has been something I have always wanted to try however I always hear horror stories about online fraud so I never know whether to trust a website.

*So, if eMart could create a simple website and guarantee your security would you purchase from eMart at home?*

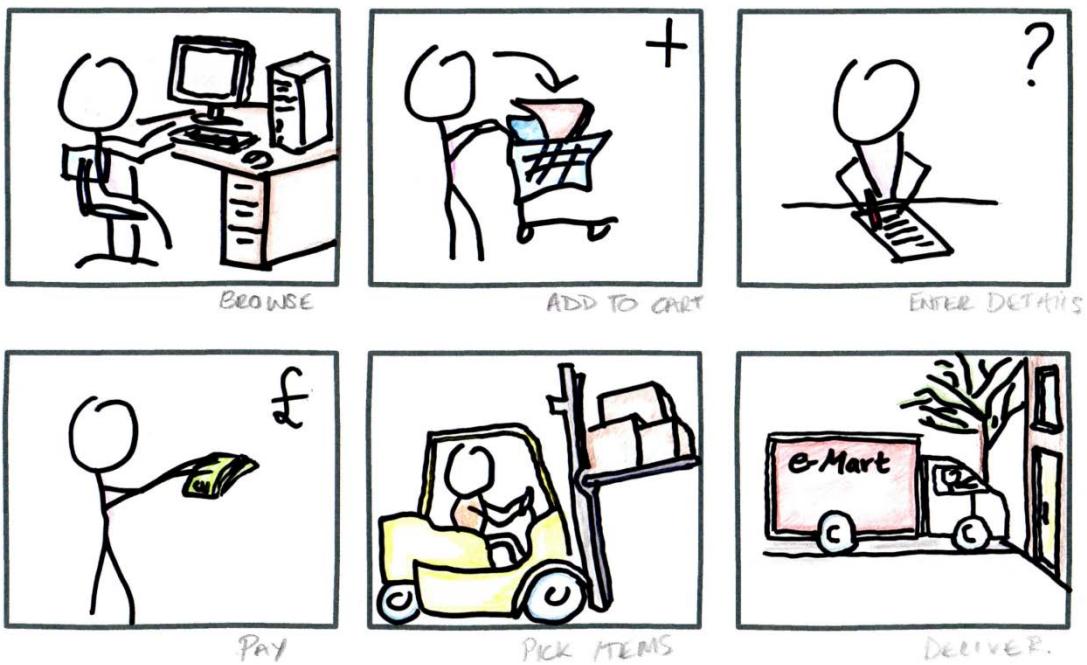
Yes, definitely. As long as the website was easy to use, and it had the same choice as in store it would be of a great benefit to me.

## 4.2 Ecommerce Walkthrough

In an attempt to gather the relevant requirement information (i.e. the data to be used and how the data will be used) for the customer view of the system, research was carried out. Several live ecommerce sites were visited in an attempt to gain a thorough understanding of the process. A depiction of the key steps involved in the process of a customer making an order is included below.

1. Browse through the product catalogue.
2. Click 'Add to cart' button on desired products.
3. When shopping is complete, Click 'Checkout'. User informed to enter personal details and select delivery option.
4. Select a payment method and submit.
5. Order information reported to warehouse for sorting.
6. Items delivered to customer on selected day.

## E-COMMERCE WALKTHROUGH.



Further phases were identified that were significant in the process. These included:

1. Customer visits e-commerce website and browses through the product catalogue.
2. A connection with the database is established.
3. Customer places their order.
4. Secure Socket Layer.
5. Checkout commenced.
6. Credit card verification process.
7. Customer credit card account.
8. Order Confirmed.
9. Order information reported to warehouse for sorting.
10. Order is fulfilled and delivered.

### 4.3 Domain Analysis

#### Overview

This document provides background information about how similar systems implement a stock management system. This information is to be used to guide the development of software.

#### Stock Management System

##### *General Knowledge about the Domain*

- Stock levels can only be updated during working hours (incoming stock).

- Stock levels can decrease 24/7 (through customer orders).
- New stock can be added to the system during working hours.
- Old stock can be deleted from the system during working hours.
- Stock prices can be updated on the system during working hours.
- Stock images can be updated during working hours.

#### *Clients and Users*

- Technicians typically install software that must be available to all users.

#### *Environment*

- Several PC's will be located throughout the warehouse for assistants to use.
- PC's will be located in the offices for managers, accounts employees and system administrators to use.

#### *Tasks and Procedures Currently Performed*

- Stock levels for shop based eMart is currently updated manually by the store manager.

#### **Payroll**

##### *General Knowledge about the Domain*

- Employee's contract hours can be edited(added, changed, deleted) on the system.
- Employee's holidays can be edited(added, changed) on the system.

#### *Clients and Users*

- Clients will be any sized companies that employ people and need to implement a payroll system.
- Technicians typically install software that must be available to all users.

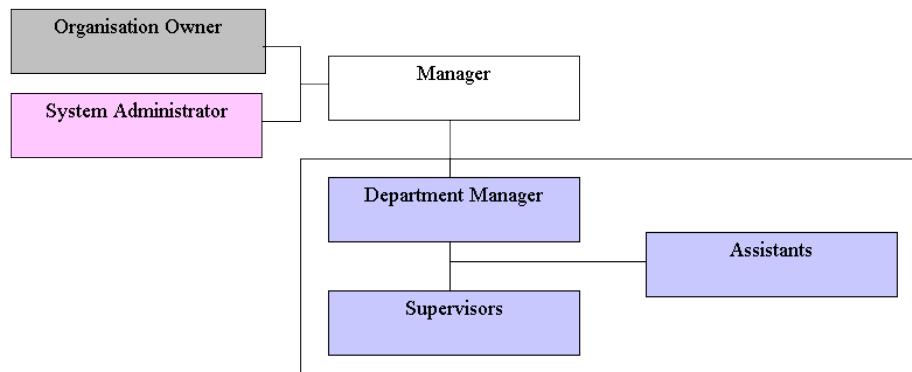
#### *Environment*

- Store manager and accounts employees have PC's with access to the payroll section of the system within their offices.

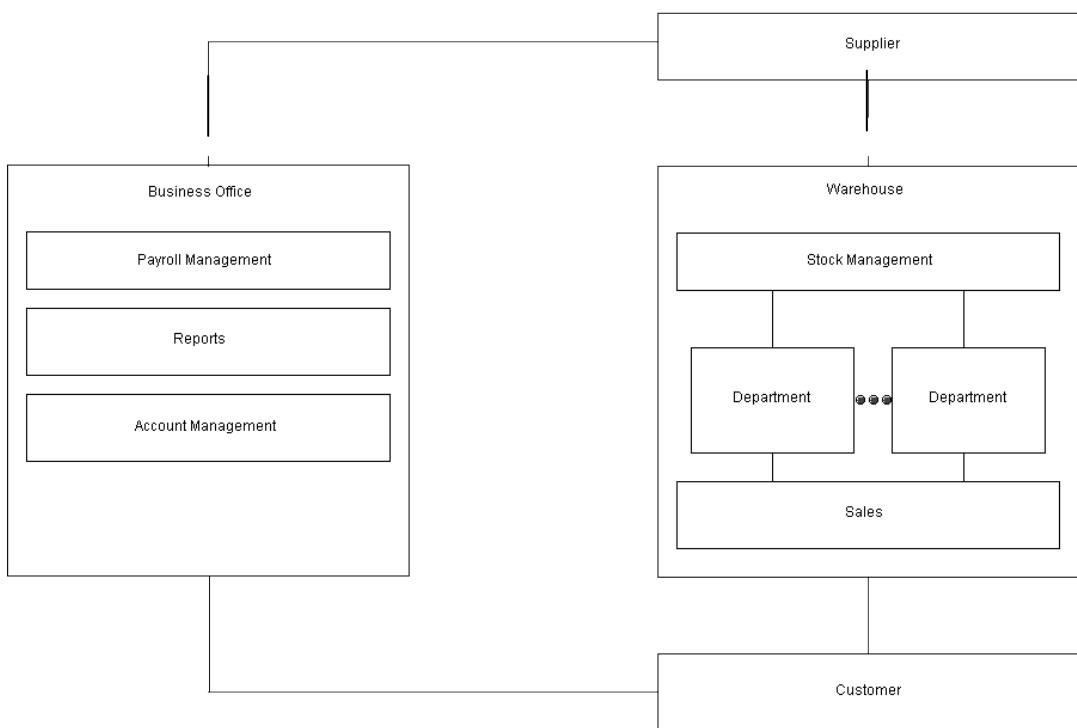
#### *Tasks and Procedures Currently Performed*

- Store manager does it manually using clocking in cards and enters information into a spreadsheet.

#### 4.4 Organisational Structure



## 4.5 Context Diagram



# Requirements Analysis

## 5.1 Stakeholders

The following are the stakeholders in the application. Each has a vested interest in the working of the database system. It is important to note that the application must be as easy to use as possible for each stakeholder.

- Organisation owner
- Employees (Manager, Supervisors, Department Managers, Merchandiser, Assistants)
- Software developer
- System Administrator
- Customers

## 5.2 Actors and Goals

The table lists all the actors, their various goals and the related use cases.

Actor	Role	Description of GOAL	Associated U/C
<b>System Administrator</b>	Initiating	goal is to control all access to the system, view/insert/modify/remove access rights, and track error and access logs	UC1, UC2, UC3, UC4, UC5
<b>Owner</b>	Initiating	goal is to access various account data	UC45, UC46, UC47, UC 48
<b>Manager</b>	Initiating	goal is to access various accounts, complete employee payroll, view/insert/modify/remove employees and rota data. view/modify/remove customers and view orders. insert/view/modify/remove items and insert/view departments.	UC6, UC7, UC8, UC9, UC10, UC11, UC12, UC13, UC14, UC15, UC17, UC19, UC28, UC37, UC38, UC39, UC40, UC45, UC46, UC47, UC48, UC49, UC50
<b>Supervisors</b>	Initiating	view customers, insert/view/modify/remove orders, ship orders and view items	UC17, UC28, UC29, UC30, UC31, UC31, UC38
<b>Department Manager</b>	Initiating	view/insert/modify/remove employees and rota data. view customers and view orders. insert/view/modify/remove items and insert/view departments	UC6, UC7, UC8, UC9, UC10, UC11, UC12, UC13, UC17, UC28, UC37, UC38, UC39, UC40, UC49, UC50
<b>Merchandiser</b>	Initiating	goal is to	UC33, UC34, UC35,

		insert/view/modify/remove suppliers, view/insert/modify/delete item data, and insert/view/modify/remove supplier delivery's	UC36, UC337, UC38, UC39, UC40, UC51, UC52, UC53, UC54
<b>All Employees</b>	Participating	goal is to view rota and payroll reports data, to login and logout of the system	UC13, UC15, UC55, UC56
<b>Assistants</b>	Initiating	goal is to update order status, view items, ship orders	UC28, UC29, UC32, UC38
<b>Customers</b>	Initiating	goal is to order goods online and pay for them, to view/modify personal data	UC16, UC18, UC20, UC21, UC22, UC23, UC24, UC25, UC26, UC27
<b>Inventory Database</b>	Initiating	goal is to report on stock level and stock expiry	UC43, UC44
<b>System</b>	Initiating	goal is to log errors and database access	UC40, UC41

## 5.3 Use Case Descriptions

### **UC1: CREATE NEW USER**

(“System administrator creates a new user for the system.”)

**Primary Actor:** System administrator

**Goal in Context:** To access the computer terminal for the system, create a new user and grant access rights to them.

**Scope:** System Management

**Precondition:** System administrator decides to create a new user. System administrator is granted access. Network connection established.

**Happy Path:**

1. System administrator accesses terminal
2. System administrator selects user table option
3. System administrator selects to add new user
4. System administrator inserts new data
5. System administrator assigns permissions to new user
6. System administrator confirms new account
7. System adds new account

**Alternate Pathways:** None

**Exception Pathway:** The system administrator tries to create a new user with an already

existing user identifier (i.e. username).

**Postcondition:** New user is created and access rights are given to them by the system administrator.

### **UC2: UPDATE PASSWORD**

(“System administrator changes the password for a user account”)

**Primary Actor:** System administrator

**Goal in Context:** To access the computer terminal for the system and change the password for a particular user account.

**Scope:** System Management

**Precondition:** System administrator decides to change a password. System administrator is granted access. Network connection established.

**Happy Path:**

1. System administrator accesses terminal
2. System administrator selects user table option
3. System administrator applies filters
4. System administrator changes the password for a particular user account
5. System administrator confirms password change
6. System updates new password

**Alternate Pathways:** None

**Exception Pathway:** The system administrator tries to change the password to a simple (easy to decrypt) set of characters.

**Postcondition:** Password is changed and new password given to the relevant user.

### **UC3: DELETE USER**

(“System administrator deletes the account of a user”)

**Primary Actor:** System administrator

**Goal in Context:** To access the computer terminal for the system and delete a particular user account.

**Scope:** System Management

**Precondition:** System administrator decides to delete an account. System administrator is granted access. Network connection established.

**Happy Path:**

1. System administrator accesses terminal
2. System administrator selects user table option
3. System administrator attempts to delete account

4. System administrator confirms account deletion
5. System deletes user account

**Alternate Pathways:** None

**Exception Pathway:** The system administrator tries to delete an account but fails to enter correct password to confirm deletion.

**Postcondition:** User is deleted from the system.

#### **UC4: ACCESS ‘ERROR LOG’**

(“System administrator views the system error log”)

**Primary Actor:** System administrator

**Goal in Context:** To access the computer terminal for the system and view the system error log.

**Scope:** System Management

**Precondition:** System administrator decides to check the error log. System administrator is granted access. Network connection established.

- Happy Path:**
1. System administrator accesses terminal
  2. System administrator selects logs option
  3. System administrator selects error log option
  4. System administrator filters by date
  5. System administrator is shown error log

**Alternate Pathways:** None

**Exception Pathway:** None

**Postcondition:** Error log is shown and can be printed.

#### **UC5: ACCESS ‘ACCESS LOG’**

(“System administrator views the system access log”)

**Primary Actor:** System administrator

**Goal in Context:** To access the computer terminal for the system and view the system access log.

**Scope:** System Management

**Precondition:** System administrator decides to check the database access log. System administrator is granted access. Network connection established.

- Happy Path:**
1. System administrator accesses terminal
  2. System administrator selects logs option

3. System administrator selects access log option
4. System administrator filters by date
5. System administrator is shown access log

**Alternate Pathways:** None

**Exception Pathway:** None

**Postcondition:** Access log is shown and can be printed.

#### **UC6: ADD EMPLOYEE**

(“A new employee is added to the database”)

**Primary Actor:** Manager, Department manager

**Goal in Context:** To access the computer terminal for the system and insert data for new employee.

**Scope:** HR Management

**Precondition:** Manager or department manager decides to add data for new employee. Actor is granted access. Network connection established.

- Happy Path:**
1. Manager or department manager accesses terminal
  2. Manager or department manager selects employee option
  3. System displays unfiltered employee list
  4. Manager or department manager selects to add new employee
  5. Manager or department manager adds new employee data and creates new payroll instance
  6. Manager or department manager confirms new addition
  7. System adds new employee account

**Alternate Pathways:** None

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** New employee is added to the system and can be used immediately.

#### **UC7: LOCATE EMPLOYEE(S)**

(“Database is filtered for employee(s)”)

**Primary Actor:** Manager, Department manager

**Goal in Context:** To search for an employee record or group of records.

**Scope:** HR Management

**Precondition:** Manager or department manager decides to filter employees. Actor is granted access. Network connection established.

- Happy Path:**
1. Manager or department manager accesses terminal
  2. Manager or department manager selects employee option
  3. System displays unfiltered employee list
  4. Manager or department manager filters list
  5. System displays filtered employee list

**Alternate Pathways:** None

**Exception Pathway:** None.

**Postcondition:** Filtered set of results displayed.

### **UC8: UPDATE EMPLOYEE**

(“A modification is made to existing employee data”)

**Primary Actor:** Manager, Department manager

**Goal in Context:** To access the computer terminal for the system and update the data for an existing employee account.

**Scope:** HR Management

**Include:** UC7

**Precondition:** Manager or department manager decides to update employee data. Actor is granted access. Network connection established.

- Happy Path:**
1. Manager or department manager accesses terminal
  2. Manager or department manager locates employee (UC7)
  3. Manager or department manager selects to make modification
  4. Manager or department manager makes modifications
  5. Manager or department manager confirms update
  6. System updates employee account

**Alternate Pathways:** None

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Employee data is up-to-date.

### **UC9: DELETE EMPLOYEE**

(“An existing employee account is deleted from the system”)

**Primary Actor:** Manager, Department manager

**Goal in Context:** To access the computer terminal for the system and delete the data for an existing employee account.

**Scope:** HR Management

<b>Include:</b>	UC7
<b>Precondition:</b>	Manager or department manager decides to delete unnecessary employee data. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Manager or department manager accesses terminal</li> <li>2. Manager or department manager locates employee (UC7)</li> <li>3. Manager or department manager selects to delete employee</li> <li>4. Manager or department manager confirms deletion</li> <li>5. System deletes employee account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Unnecessary employee account is deleted from the database.

#### **UC10: ADD ROTA DATA**

(“The monthly rota data is inserted into the database”)

<b>Primary Actor:</b>	Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and insert the rota data for a month.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager or department manager decides to add rota data. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Manager or department manager accesses terminal</li> <li>2. Manager or department manager selects rota option</li> <li>3. Manager or department manager selects to add new month</li> <li>4. Manager or department manager adds data</li> <li>5. Manager or department manager confirms insertion</li> <li>6. System adds new rota data</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Rota data for month duration is added.

#### **UC11: UPDATE ROTA**

(“A modification is made to existing rota data”)

<b>Primary Actor:</b>	Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and update the data for

existing rota instance.

<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager or department manager decides to update rota data. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Manager or department manager accesses terminal</li><li>2. Manager or department manager selects rota option</li><li>3. Manager or department manager searches for rota</li><li>4. Manager or department manager selects to make modification</li><li>5. Manager or department manager makes modifications</li><li>6. Manager or department manager confirms update</li><li>7. System updates rota data</li></ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Rota data is up-to-date.

#### **UC12: DELETE ROTA DATA**

(“An existing rota instance is deleted from the system”)

<b>Primary Actor:</b>	Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and deletes existing rota data.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager or department manager decides to delete rota data. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Manager or department manager accesses terminal</li><li>2. Manager or department manager selects rota option</li><li>3. Manager or department manager searches for rota</li><li>4. Manager or department manager selects to delete rota</li><li>5. Manager or department manager confirms deletion</li><li>6. System deletes rota data</li></ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Rota data is deleted from the database.

#### **UC13: SHOW ROTA DATA**

(“A report for a rota is created”)

<b>Primary Actor:</b>	Manager, Department manager
<b>Secondary Actor:</b>	All Employees
<b>Goal in Context:</b>	To access the computer terminal for the system and view rota data.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	<p>1. Manager or department manager decides to view a rota report.</p> <p>2. All employees decide to view a rota report.</p> <p>Actor is granted access. Network connection established.</p>
<b>Happy Path:</b>	<p>1.1 Manager or department manager accesses terminal</p> <p>1.2 Manager or department manager selects rota option</p> <p>1.3 Manager or department manager searches for rota</p> <p>1.4 Manager or department manager selects to view report</p> <p>1.5 System displays rota report</p> <p>3.1 Employee accesses terminal</p> <p>3.2 Employee selects rota option</p> <p>3.3 System shows rota for current month</p> <p>3.4 Employee selects to view report</p> <p>3.5 System displays rota report</p>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Customised report is created based on authorisation level and option to print report made available.

#### **UC14: RUN PAYROLL**

(“Calculate the payroll”)

<b>Primary Actor:</b>	Manager
<b>Goal in Context:</b>	To access the computer terminal for the system and run the automatic calculation of payroll.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager decides to run payroll calculation. Manager is granted access. Network connection established.
<b>Happy Path:</b>	<p>1. Manager accesses terminal</p> <p>2. Manager selects payroll option</p> <p>3. Manager selects to run payroll</p> <p>4. Manager selects payroll period</p> <p>5. Manager confirms process</p> <p>6. System calculates payroll for period</p>

<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation. Process attempted for incomplete period.
<b>Postcondition:</b>	Payroll is up-to-date.

### **UC15: CREATE PAYROLL REPORT**

(“Payroll report on period (i.e. total hours worked, net pay etc) ”)

<b>Primary Actor:</b>	Manager
<b>Secondary Actor:</b>	All employees
<b>Goal in Context:</b>	To access the computer terminal for the system and view payroll report.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager decides to view payroll report. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ul style="list-style-type: none"> <li>1.1 Manager accesses terminal</li> <li>1.2 Manager selects payroll option</li> <li>1.3 Manager selects report option</li> <li>1.4 Manager searches for payroll period</li> <li>1.5 System displays payroll report</li> <li>2.1 Employee accesses terminal</li> <li>2.2 Employee selects payroll option</li> <li>2.3 Employee selects report option</li> <li>2.4 Employee searches for payroll period</li> <li>2.5 System displays individual payroll report</li> </ul>

<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Process attempted for incomplete period.
<b>Postcondition:</b>	Payroll report is created and can be printed.

### **UC16: REGISTER CUSTOMER**

(“A new customer is added to the database”)

<b>Primary Actor:</b>	Customer
<b>Goal in Context:</b>	To add new customer data to the database.
<b>Scope:</b>	Order Management
<b>Precondition:</b>	A new customer registers their data. Network connection established.
<b>Happy Path:</b>	<ul style="list-style-type: none"> <li>1. A customer enters their data on the website</li> </ul>

2. The data is validated
3. System adds new data to database

**Alternate Pathways:** None.

**Exception Pathway:** Data entered fails validation (e.g. post code syntax error). Email address already exists in the system.

**Postcondition:** New customer is added to the system.

**UC17: LOCATE CUSTOMER(S)**  
*(“Database is filtered for customer(s)”)*

**Primary Actor:** Manager, Department manager, Supervisors

**Goal in Context:** To search for a customer record or group of records.

**Scope:** Order Management

**Precondition:** Actor decides to filter customers. Actor is granted access. Network connection established.

- Happy Path:**
1. Actor accesses terminal
  6. Actor selects customer option
  7. System displays unfiltered customer list
  8. Actor filters list
  9. System displays filtered customer list

**Alternate Pathways:** None

**Exception Pathway:** None.

**Postcondition:** Filtered set of results displayed.

**UC18: UPDATE CUSTOMER**  
*(“An existing customer’s data is updated”)*

**Primary Actor:** Customer

**Secondary Actor:** Manager

**Goal in Context:** To update customer data in the database.

**Scope:** Order Management

**Include:** UC17

**Precondition:** An existing customer decides to change their data. Actor is granted access. Network connection established.

<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. A customer logs into their account on the website</li> <li>2. Log in credentials are validated</li> <li>3. Customer selects modify personal data</li> <li>4. Customer changes data</li> <li>5. Customer confirms change</li> <li>6. The data is validated</li> <li>7. System changes data</li> </ol>
<b>Alternate Pathways:</b>	A manager changes the customer data after informed to do so by customer. (UC17)
<b>Exception Pathway:</b>	Data entered fails validation (e.g. post code syntax error).
<b>Postcondition:</b>	Customer data is up-to-date.

### ***UC19: DELETE CUSTOMER***

(“An existing customer’s data is deleted”)

<b>Primary Actor:</b>	Manager
<b>Goal in Context:</b>	To access the computer terminal for the system and delete customer data from the database.
<b>Scope:</b>	Order Management
<b>Include:</b>	UC17
<b>Precondition:</b>	An existing customer decides they want their data deleted from the system. Manager is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Manager accesses the terminal</li> <li>2. Manager searches for customer (UC17)</li> <li>3. Customer selects delete option</li> <li>4. Customer confirms deletion</li> <li>5. System deletes customer account</li> </ol>
<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	Illegal password entered during confirmation. Customer does not exist.
<b>Postcondition:</b>	Customer account is deleted from the system.

### ***UC20: BROWSE PRODUCTS***

(“Customer browses current inventory”)

<b>Primary Actor:</b>	Customer
<b>Goal in Context:</b>	To browse through the current inventory.

<b>Scope:</b>	Order Management
<b>Precondition:</b>	A customer decides they want to browse the inventory. Network connection established. Have store homepage open.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Customer selects a product category</li> <li>2. System displays inventory for selected category</li> </ol>
<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Category specific inventory is displayed to customer.

**UC21: SEARCH FOR ITEM**  
*(“Customer searches for a particular item”)*

<b>Primary Actor:</b>	Customer
<b>Goal in Context:</b>	To view data for particular item.
<b>Scope:</b>	Order Management
<b>Precondition:</b>	A customer decides they want to view data for an item currently in the inventory.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Customer enters product search parameters and requests a product search</li> <li>2. System searches through the inventory database</li> <li>3. System displays matching results</li> </ol>
<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	Search term not recognised. Notification displayed.
<b>Postcondition:</b>	System displays a list of matching items to the customer.

**UC22: ADD TO CART**  
*(“A customer adds an item to their shopping cart (order)”)*

<b>Primary Actor:</b>	Customer
<b>Goal in Context:</b>	To add individual item to a virtual shopping cart.
<b>Scope:</b>	Order Management
<b>Precondition:</b>	A customer decides they want to add an item to their shopping cart. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Customer selects to add item to cart</li> </ol>

- System stores item in the shopping cart instance

**Alternate Pathways:** None.

**Exception Pathway:** Connection lost. Shopping cart data lost.

**Postcondition:** New item is added to the shopping cart instance.

### **UC23: VIEW CART**

(“A customers shopping cart is displayed”)

**Primary Actor:** Customer

**Goal in Context:** To view current state of shopping cart.

**Scope:** Order Management

**Precondition:** A customer decides they want to view the items in their shopping cart.  
Network connection established.

**Happy Path:**

- Customer selects to view shopping cart
- System displays current state of shopping cart

**Alternate Pathways:** No shopping cart instance created. Empty cart message displayed.

**Exception Pathway:** Connection lost. Shopping cart data lost.

**Postcondition:** Current shopping cart state is displayed with up-to-date unit price and total price.

### **UC24: UPDATE QUANTITY**

(“A customer updates the quantity for an item in their cart”)

**Primary Actor:** Customer

**Goal in Context:** To update state of shopping cart.

**Scope:** Order Management

**Precondition:** A customer decides they want to update the quantity of an item in their shopping cart. Network connection established.

**Happy Path:**

- Customer selects to view shopping cart
- System displays current state of shopping cart
- Customer modifies the quantity of a particular item (or several items)
- Customer clicks ‘update cart’
- System displays updated state of the shopping cart

**Alternate Pathways:** None.

<b>Exception Pathway:</b>	Connection lost.
<b>Postcondition:</b>	Current shopping cart state is displayed with up-to-date quantity, unit price and total price.

### **UC25: REMOVE ITEM FROM CART**

(“A customer removes an item from the shopping cart”)

<b>Primary Actor:</b>	Customer
<b>Goal in Context:</b>	To remove an item from the shopping cart.
<b>Scope:</b>	Order Management
<b>Precondition:</b>	A customer decides they want to remove an item from their shopping cart. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Customer selects to view shopping cart</li> <li>2. System displays current state of shopping cart</li> <li>3. Customer clicks ‘remove from cart’</li> <li>4. System displays updated state of the shopping cart</li> </ol>
<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	Connection lost.
<b>Postcondition:</b>	Current shopping cart state is displayed with up-to-date items and total price.

### **UC26: PLACE ORDER**

(“A new order is made”)

<b>Primary Actor:</b>	Customer
<b>Secondary Actor:</b>	Supervisor
<b>Goal in Context:</b>	To submit a completed order to the system.
<b>Scope:</b>	Order Management
<b>Include:</b>	UC16
<b>Precondition:</b>	A customer decides they want to “checkout”. Items exist in the customer’s shopping basket. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Customer selects checkout option</li> <li>2.1 If customer is registered they are forwarded to PayPal for payment processing.</li> <li>2.2 Else customer is asked to enter registration data (UC16)</li> <li>3.1 If PayPal processing is successful order form is processed by the</li> </ol>

system  
3.2 Else process is cancelled and customer is returned to their shopping cart with a notification message displayed

**Alternate Pathways:** Supervisor processes order when instructed to do so on a customer's behalf.

**Exception Pathway:** Payment processing unsuccessful. Checkout process incomplete by customer.

**Postcondition:** New order is created and added to the system. Notification is sent.

### **UC27: VIEW ORDER HISTORY**

(“A customer order history is displayed”)

**Primary Actor:** Customer

**Goal in Context:** To browse recent orders.

**Scope:** Order Management

**Precondition:** A customer is logged into the system. Customer decides they want to search their recent order history (30days-). Actor is granted access. Network connection established.

**Happy Path:**  
1. Customer selects recent orders  
2. System displays order data related to customer within past 30 days

**Alternate Pathways:** None.

**Exception Pathway:** No order history.

**Postcondition:** Brief descriptions of recent orders are displayed to the customer with the most recent first. Customer can choose to view more data on particular order by selecting the order.

### **UC28: LOCATE ORDER(S)**

(“Database is filtered for order(s)”)

**Primary Actor:** Manager, Department manager, Supervisor, Assistants

**Goal in Context:** To search for an order record or group of records.

**Scope:** Order Management

**Precondition:** Actor decides to filter employees. Actor is granted access. Network connection established.

**Happy Path:**  
6. Actor accesses terminal

7. Actor selects orders option
8. System displays unfiltered orders list
9. Actor filters list
10. System displays filtered orders list

**Alternate Pathways:** None

**Exception Pathway:** None.

**Postcondition:** Filtered set of results displayed.

### **UC29: UPDATE ORDER STATUS**

(“A modification is made to an order’s status in the system”)

**Primary Actor:** Supervisor, Assistants

**Goal in Context:** To update the status of an order in the system.

**Scope:** Order Management

**Include:** UC28

**Precondition:** A customer has submitted an order. Actor is granted access. Status of order needs changing Network connection established.

- Happy Path:**
1. Supervisor or assistant accesses terminal
  2. Supervisor or assistant searches for order (UC28)
  3. Supervisor or assistant select to update status
  4. Supervisor or assistant update the status of the order
  5. Supervisor or assistant confirms update
  6. System updates the order data

**Alternate Pathways:** None.

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Order data is up-to-date.

### **UC30: UPDATE ORDER**

(“A modification is made to an order in the system (e.g. item not in stock)”)

**Primary Actor:** Supervisor

**Goal in Context:** To update order data in the system.

**Scope:** Order Management

**Include:** UC28

**Precondition:** A customer has submitted an order. A problem exists with an item or

items in the system. Supervisor is granted access. Network connection established.

- Happy Path:**
1. Supervisor accesses terminal
  2. Supervisor searches for order (UC28)
  3. Supervisor selects to modify order
  4. Supervisor modifies the order data
  5. Supervisor confirms modification
  6. System updates the order data

**Alternate Pathways:** None.

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Order data is up-to-date.

### **UC31: DELETE ORDER**

(“An order is deleted from the database”)

**Primary Actor:** Supervisor

**Goal in Context:** To delete order data from the database.

**Scope:** Order Management

**Include:** UC28

**Precondition:** Supervisor is granted access. Network connection established.

- Happy Path:**
1. Supervisor accesses terminal
  2. Supervisor searches for order (UC28)
  3. Supervisor selects to delete order
  4. Supervisor confirms deletion
  5. System deletes the order data

**Alternate Pathways:** System administrator deletes obsolete data from the system after back up.

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Order data is deleted.

### **UC32: SHIP ORDER**

(“An order in the system is fulfilled”)

**Primary Actor:** Assistants

**Secondary Actor:** Supervisor

**Goal in Context:** To ship an order.

<b>Scope:</b>	Order Management
<b>Include:</b>	UC29
<b>Precondition:</b>	An order has been fulfilled and is ready to be despatched. Actor is granted access.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Assistant gets order ready for despatch</li> <li>2. Order is shipped to customer</li> <li>3. Assistant updates orders status (UC29)</li> </ol>
<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Customer receives their order.

#### ***UC33: ADD SUPPLIER***

(“Add new supplier data to the system”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To access the computer terminal for the system and insert data for new supplier.
<b>Scope:</b>	Stock Management
<b>Precondition:</b>	Merchandiser decides to add data for new supplier. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser selects supplier option</li> <li>3. System displays unfiltered supplier list</li> <li>4. Merchandiser selects to add new supplier</li> <li>5. Merchandiser adds new supplier data</li> <li>6. Merchandiser confirms new addition</li> <li>7. System adds new supplier account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	New supplier is added to the system and can be used immediately.

#### ***UC34: LOCATE SUPPLIER(S)***

(“Database is filtered for supplier(s)”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To search for a supplier record or group of records.

<b>Scope:</b>	Stock Management
<b>Precondition:</b>	Merchandiser decides to filter suppliers. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser selects supplier option</li> <li>3. System displays unfiltered supplier list</li> <li>4. Merchandiser filters list</li> <li>5. System displays filtered supplier list</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Filtered set of results displayed.

**UC35: UPDATE SUPPLIER**  
 (“Update existing supplier data in the system”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To access the computer terminal for the system and update data for an existing supplier.
<b>Scope:</b>	Stock Management
<b>Include:</b>	UC34
<b>Precondition:</b>	Merchandiser decides to update data for an existing supplier. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser filters supplier list (UC34)</li> <li>3. Merchandiser selects to modify data</li> <li>4. Merchandiser makes modifications data</li> <li>5. Merchandiser confirms modifications</li> <li>6. System updates supplier account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Supplier data is up-to-date.

**UC36: DELETE SUPPLIER**  
 (“An existing supplier account is deleted from the system”)

<b>Primary Actor:</b>	Merchandiser
-----------------------	--------------

<b>Goal in Context:</b>	To access the computer terminal for the system and delete the data for an existing supplier account.
<b>Scope:</b>	Stock Management
<b>Include:</b>	UC34
<b>Precondition:</b>	Merchandiser decides to delete unnecessary supplier data. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser locates supplier (UC34)</li> <li>3. Merchandiser selects to delete supplier</li> <li>4. Merchandiser confirms deletion</li> <li>5. System deletes supplier account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Unnecessary supplier account is deleted from the database.

#### **UC37: ADD ITEM** ("Add new item data to the system")

<b>Primary Actor:</b>	Merchandiser, Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and insert data for new item.
<b>Scope:</b>	Stock Management
<b>Precondition:</b>	Actor decides to add data for new item. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Actor accesses terminal</li> <li>2. Actor selects stock option</li> <li>3. System displays unfiltered stock list</li> <li>4. Actor selects to add new item</li> <li>5. Actor adds new item data</li> <li>6. Actor confirms new addition</li> <li>7. System adds new item account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	New item is added to the system and can be used immediately.

### **UC38: LOCATE ITEM(S)**

(“Database is filtered for item(s)”)

<b>Primary Actor:</b>	Merchandiser, Manager, Department manager, Supervisors, Assistants
<b>Goal in Context:</b>	To search for an item record or group of records.
<b>Scope:</b>	Stock Management
<b>Precondition:</b>	Actor decides to filter item. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Actor accesses terminal</li><li>2. Actor selects stock option</li><li>3. System displays unfiltered item list</li><li>4. Actor filters list</li><li>5. System displays filtered item list</li></ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Filtered set of results displayed.

### **UC39: UPDATE ITEM**

(“Update existing item data in the system”)

<b>Primary Actor:</b>	Merchandiser, Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and update data for an existing item.
<b>Scope:</b>	Stock Management
<b>Include:</b>	UC38
<b>Precondition:</b>	Actor decides to update data for an existing item. Actor is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Actor accesses terminal</li><li>2. Actor filters item list (UC38)</li><li>3. Actor selects to modify data</li><li>4. Actor makes modifications data</li><li>5. Actor confirms modifications</li><li>6. System updates item data</li></ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.

**Postcondition:** Item data is up-to-date.

#### **UC40: DELETE ITEM**

(“An existing item record is deleted from the system”)

**Primary Actor:** Merchandiser, Manager, Department manager

**Goal in Context:** To access the computer terminal for the system and delete the data for an existing item.

**Scope:** Stock Management

**Include:** UC38

**Precondition:** Actor decides to delete unnecessary item data. Actor is granted access. Network connection established.

- Happy Path:**
1. Actor accesses terminal
  2. Actor locates item (UC38)
  3. Actor selects to delete item
  4. Actor confirms deletion
  5. System deletes item data

**Alternate Pathways:** None

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Unnecessary item data is deleted from the database.

#### **UC41: UPDATE ACCESS LOG**

(“Any sequence (3 times) of unsuccessful log in attempts are logged in the access log”)

**Primary Actor:** System

**Goal in Context:** To log any sequence of unsuccessful log in attempts. IP address, date and time will be logged.

**Scope:** System Management

**Precondition:** Access log has been created. Network connection established.

- Happy Path:**
1. System writes to log, the IP address of the access attempt, the current date and the time.
  2. Log is updated

**Alternate Pathways:** System Administrator manually alters the log.

**Exception Pathway:** None.

**Postcondition:** Log is up-to-date.

### **UC42: UPDATE ERROR LOG**

(“Any exceptions are logged in the error log”)

**Primary Actor:** System

**Goal in Context:** To log any exceptions thrown. Error type, date and time will be logged.

**Scope:** System Management

**Precondition:** Error log has been created. Network connection established.

**Happy Path:**

1. System writes to log, error type, the current date and time.
2. Log is updated

**Alternate Pathways:** System Administrator manually alters the log.

**Exception Pathway:** None.

**Postcondition:** Log is up-to-date.

### **UC43: FIRE ITEM EXPIRY WARNING**

(“A notification is fired if any items in the inventory are due to expire”)

**Primary Actor:** Inventory System

**Goal in Context:** To notify of any expiring items.

**Scope:** Stock Management

**Precondition:** Item is due to expire. Network connection established.

**Happy Path:**

1. System fires notification for expiring item

**Alternate Pathways:** None.

**Exception Pathway:** None.

**Postcondition:** Relevant personnel see the notification.

### **UC44: FIRE LOW STOCK LEVEL WARNING**

(“A notification is fired if the stock level of any items reaches stated level”)

**Primary Actor:** Inventory System

**Goal in Context:** To notify of low stock levels.

**Scope:** Stock Management

**Precondition:** Stock level reaches stated level. Network connection established.

**Happy Path:**

1. System fires notification for low stock level

<b>Alternate Pathways:</b>	None.
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Relevant personnel see the notification.
<b>UC45: ADD ACCOUNT</b> ("Add new account data to the system")	
<b>Primary Actor:</b> Owner, Manager	
<b>Goal in Context:</b>	To access the computer terminal for the system and insert data for a new account.
<b>Scope:</b>	Account Management
<b>Precondition:</b>	Owner or manager decides to add data for new account. Owner or manager is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Owner or manager accesses terminal</li> <li>2. Owner or manager selects accounts option</li> <li>3. System displays unfiltered account list</li> <li>4. Owner or manager selects to add new account</li> <li>5. Owner or manager adds new account data</li> <li>6. Owner or manager confirms new addition</li> <li>7. System adds new account</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	New account is added to the system and can be used immediately.

<b>Primary Actor:</b>	Owner, Manager
<b>Goal in Context:</b>	To search for an account record or group of records.
<b>Scope:</b>	Account Management
<b>Precondition:</b> Owner or manager decides to filter account(s). Owner or manager is granted access. Network connection established.	
<b>Happy Path:</b> <ol style="list-style-type: none"> <li>1. Owner or manager accesses terminal</li> <li>7. Owner or manager selects accounts option</li> <li>8. System displays unfiltered item list</li> <li>9. Owner or manager filters list</li> </ol>	

10. System displays filtered item list

**Alternate Pathways:** None

**Exception Pathway:** None.

**Postcondition:** Filtered set of results displayed.

#### **UC47: UPDATE ACCOUNT**

(“Update existing account data in the system”)

**Primary Actor:** Owner, Manager

**Goal in Context:** To access the computer terminal for the system and update data for an existing account.

**Scope:** Account Management

**Include:** UC46

**Precondition:** Owner or manager decides to update data for an existing item. Owner or manager is granted access. Network connection established.

- Happy Path:**
1. Owner or manager accesses terminal
  2. Owner or manager filters accounts list (UC46)
  3. Owner or manager selects to modify data
  4. Owner or manager makes modifications data
  5. Owner or manager confirms modifications
  6. System updates account data

**Alternate Pathways:** None

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** Account data is up-to-date.

#### **UC48: DELETE ACCOUNT**

(“An existing account record is deleted from the system”)

**Primary Actor:** Owner, Manager

**Goal in Context:** To access the computer terminal for the system and delete the data for an existing account.

**Scope:** Account Management

**Include:** UC46

**Precondition:** Owner or manager decides to delete unnecessary account data. Owner or manager is granted access. Network connection established.

<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Owner or manager accesses terminal</li> <li>2. Owner or manager locates account (UC46)</li> <li>3. Owner or manager selects to delete account</li> <li>4. Owner or manager confirms deletion</li> <li>5. System deletes account data</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Unnecessary account data is deleted from the database.
<b>UC49: ADD DEPARTMENT</b> ("Add new department data to the system")	
<b>Primary Actor:</b>	Manager, Department manager
<b>Goal in Context:</b>	To access the computer terminal for the system and insert data for a new department.
<b>Scope:</b>	HR Management
<b>Precondition:</b>	Manager or department manager decides to add data for new department. Manager or department manager is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Manager or department manager accesses terminal</li> <li>2. Manager or department manager selects departments option</li> <li>3. System displays unfiltered department list</li> <li>4. Manager or department manager selects to add new department</li> <li>5. Manager or department manager adds new department data</li> <li>6. Manager or department manager confirms new addition</li> <li>7. System adds new department</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	New department is added to the system and can be used immediately.
<b>UC50: LOCATE DEPARTMENT(S)</b> ("Database is filtered for department(s)")	
<b>Primary Actor:</b>	Manager, Department manager
<b>Goal in Context:</b>	To search for a department record or group of records.
<b>Scope:</b>	HR Management

**Precondition:** Manager or department manager decides to filter department(s). Manager or department manager is granted access. Network connection established.

- Happy Path:**
1. Manager or department manager accesses terminal
  2. Manager or department manager selects departments option
  3. System displays unfiltered department list
  4. Manager or department manager filters list
  5. System displays filtered department list

**Alternate Pathways:** None

**Exception Pathway:** None.

**Postcondition:** Filtered set of results displayed.

### **UC51: ADD SUPPLIER DELIVERY** (“Add new supplier delivery data to the system”)

**Primary Actor:** Merchandiser

**Goal in Context:** To access the computer terminal for the system and insert data for a new supplier delivery.

**Scope:** Stock Management

**Precondition:** Merchandiser decides to add data for new supplier delivery. Merchandiser is granted access. Network connection established.

- Happy Path:**
1. Merchandiser accesses terminal
  2. Merchandiser selects stock option
  3. System displays unfiltered stock list
  4. Merchandiser selects supplier delivery option
  5. System displays unfiltered supplier delivery list
  6. Merchandiser selects to add new supplier delivery
  7. Merchandiser adds new supplier delivery data
  8. Merchandiser confirms new addition
  9. System adds new supplier delivery

**Alternate Pathways:** None

**Exception Pathway:** Illegal password entered during confirmation.

**Postcondition:** New supplier delivery is added to the system and can be used immediately.

### **UC52: LOCATE SUPPLIER DELIVERY(S)** (“Database is filtered for supplier delivery(s)”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To search for a supplier delivery record or group of records.
<b>Scope:</b>	Stock Management
<b>Precondition:</b>	Merchandiser decides to filter supplier delivery. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser selects stock option</li> <li>3. System displays unfiltered stock list</li> <li>4. Merchandiser selects supplier delivery option</li> <li>5. System displays unfiltered supplier delivery list</li> <li>6. Merchandiser filters list</li> <li>7. System displays filtered item list</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	None.
<b>Postcondition:</b>	Filtered set of results displayed.

### **UC53: UPDATE SUPPLIER DELIVERY**

(“Update existing supplier delivery data in the system”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To access the computer terminal for the system and update data for an existing supplier delivery.
<b>Scope:</b>	Stock Management
<b>Include:</b>	UC52
<b>Precondition:</b>	Merchandiser decides to update data for an existing supplier delivery. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"> <li>1. Merchandiser accesses terminal</li> <li>2. Merchandiser filters supplier delivery list (UC52)</li> <li>3. Merchandiser selects to modify data</li> <li>4. Merchandiser makes modifications to data</li> <li>5. Merchandiser confirms modifications</li> <li>6. System updates supplier delivery data</li> </ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Supplier delivery data is up-to-date.

## **UC54: DELETE SUPPLIER DELIVERY**

(“An existing supplier delivery record is deleted from the system”)

<b>Primary Actor:</b>	Merchandiser
<b>Goal in Context:</b>	To access the computer terminal for the system and delete the data for an existing supplier delivery.
<b>Scope:</b>	Stock Management
<b>Include:</b>	UC52
<b>Precondition:</b>	Merchandiser decides to delete unnecessary supplier delivery data. Merchandiser is granted access. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Merchandiser accesses terminal</li><li>2. Merchandiser locates supplier delivery(UC52)</li><li>3. Merchandiser selects to delete supplier delivery</li><li>4. Merchandiser confirms deletion</li><li>5. System deletes account data</li></ol>
<b>Alternate Pathways:</b>	None
<b>Exception Pathway:</b>	Illegal password entered during confirmation.
<b>Postcondition:</b>	Unnecessary supplier delivery data is deleted from the database.

## **UC55: LOG IN**

(“A user logs into to the system”)

<b>Primary Actor:</b>	All Employees
<b>Goal in Context:</b>	To access the computer terminal for the system and gain authorised access.
<b>Scope:</b>	System Management
<b>Precondition:</b>	Actor decides to access the system. Network connection established.
<b>Happy Path:</b>	<ol style="list-style-type: none"><li>1. Actor accesses terminal</li><li>2. System displays log in facility</li><li>3. Actor enters username</li><li>4. Actor enters password</li><li>5. Actor clicks ‘enter’</li><li>6. Actor confirms password</li><li>7. System determines whether access it to be granted (authorisation)</li><li>8. Actor has access to database facilities</li></ol>
<b>Alternate Pathways:</b>	System administrator resets system and with it all passwords.

**Exception Pathway:** Illegal username and password combination entered.

**Postcondition:** User has access to various parts of the database.

### **UC56: LOG OUT**

(“A user logs out of the system”)

**Primary Actor:** All Employees

**Goal in Context:** To log out of the system and leave it in a secure state.

**Scope:** System Management

**Precondition:** Actor is logged in. Actor decides to log out of the system. Network connection established.

- Happy Path:**
1. Actor accesses terminal
  2. System displays log out capability
  3. Actor clicks ‘log out’
  4. System returns to log in facility
  5. System is secure

**Alternate Pathways:** System administrator resets system and with it all passwords. System crashes. Connection timeout.

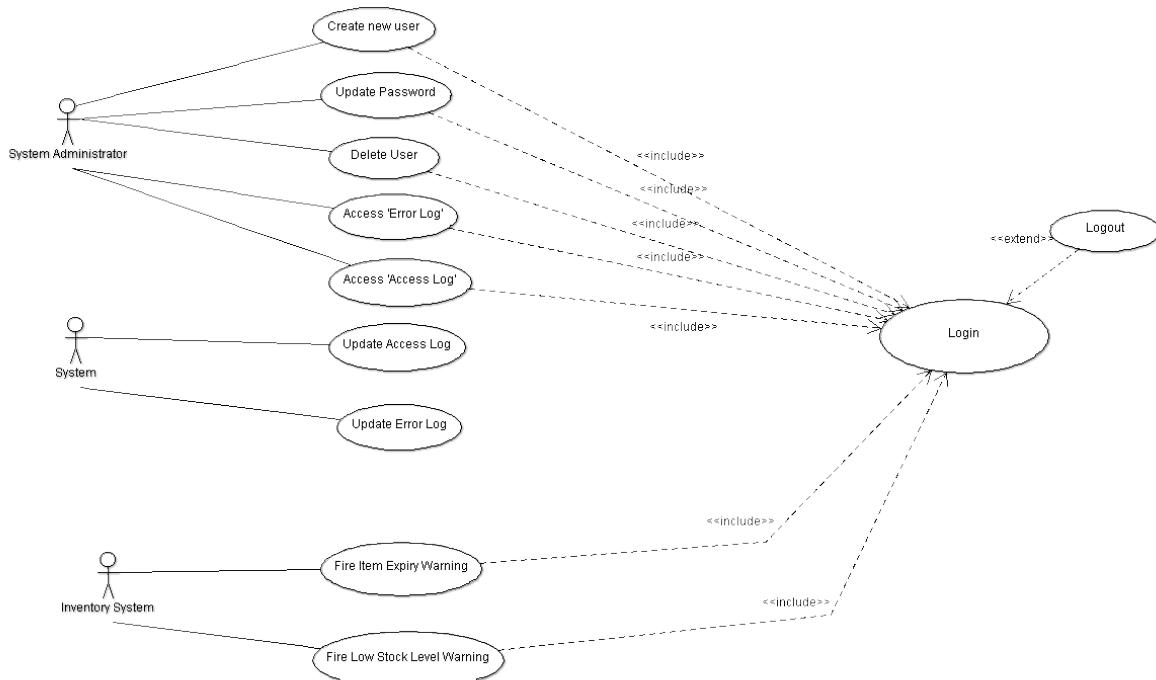
**Postcondition:** System is in a secure state.

## 5.4 Use Case Diagrams

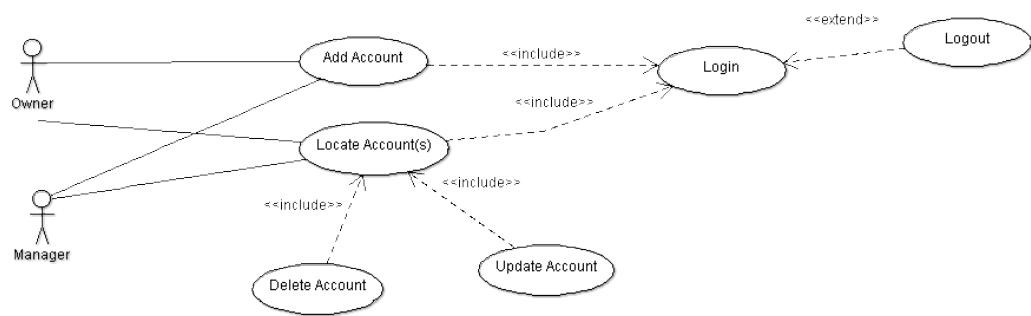
### Orders



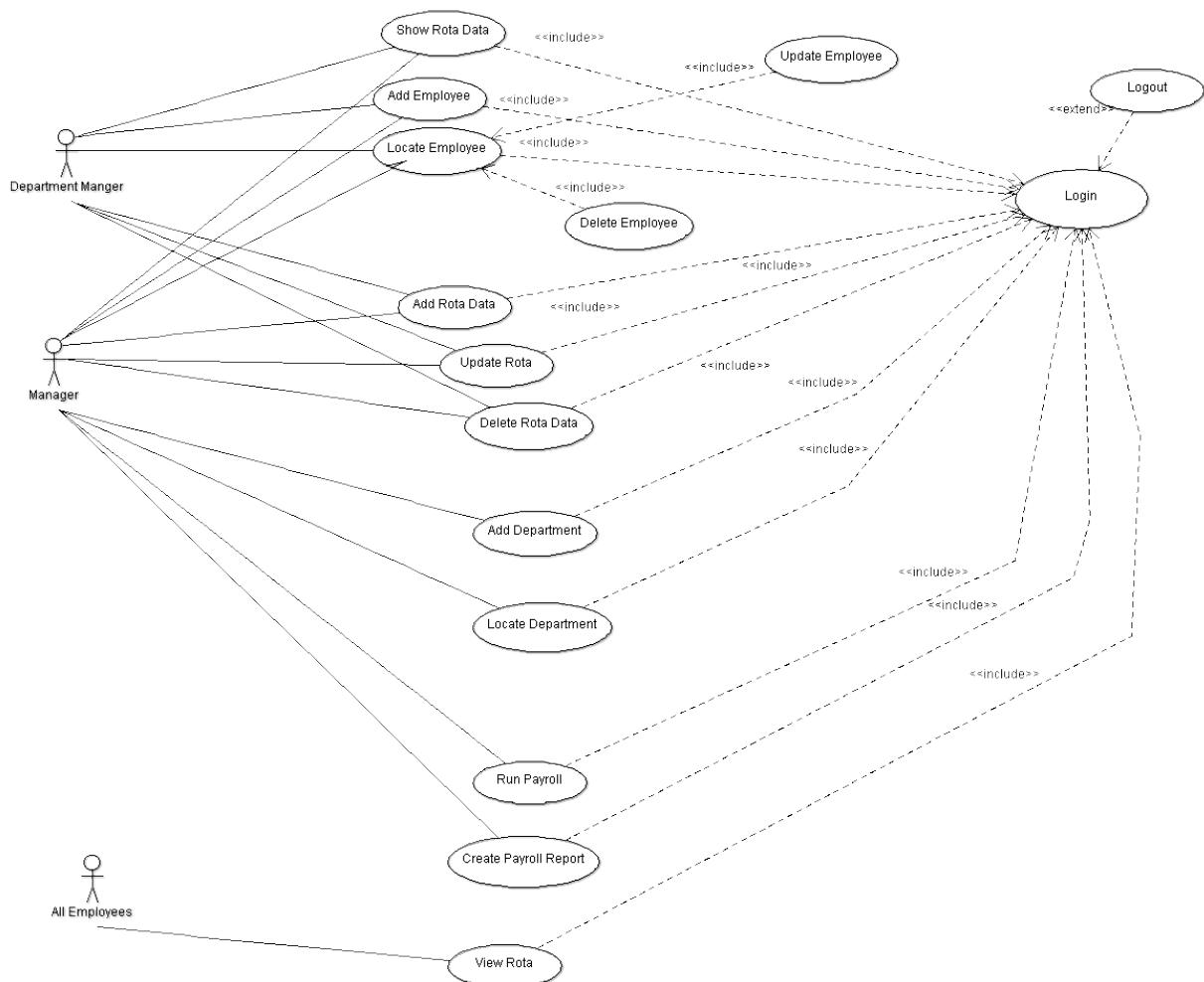
### System



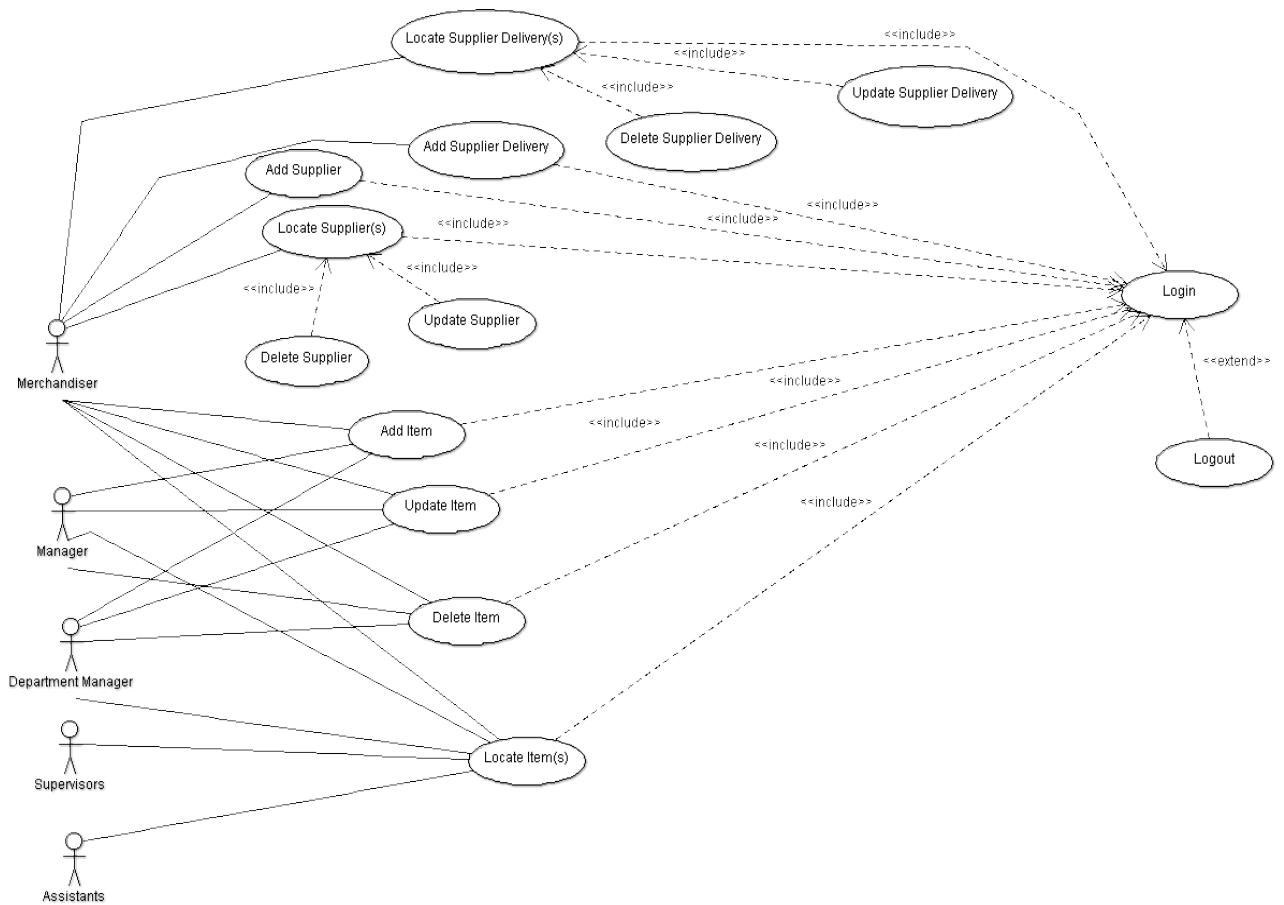
### Accounts



## HR Management



## Stock



## 5.5 Transaction Requirements

### Data Entry

- Enter the details of departments
- Enter details of new staff
- Enter details of new item
- Enter details of ideal quantities of items
- Enter details of new customer
- Enter details of new supplier
- Enter details of a new order (i.e. if supervisor is taking order over the phone)
- Enter details of new account

### Data update/deletion

- Update/delete the details of a department
- Update/delete the details of a member of staff
- Update/delete the details of a customer
- Update/delete the details of an item
- Update/delete an order
- Update/delete an account
- Update/delete a supplier

### Date queries

*A sample of queries which the database will support*

- List the details of departments
- List the name, position and salary of staff
- List the department, price, ideal stock level, current stock level and supplier of an item
- List customers and their orders
- List the name and contact details of customers
- List staff contact details
- List staff bank details
- List all suppliers

## 5.6 System Specifications

### Initial database size

- There are approximately 100 brands, 2000 products with a total stock level of approximately 50,000.
- There are currently 76 employees.
- There are currently 5 suppliers.

### Database rate of growth

- Approximately 10 new products and 450 items of each product are added to the database each month.
- On average 1 employee leaves or is employed by the company each month. The records of the employee that has left are kept for 2 years before they are deleted from the system.
- We predict that an average of 200 new customers will register per month for the first 6 months, then 125 each month thereafter. Records of customers will be deleted after a period of 5 years of inactivity.

### The types and average number of record searches

- Searching details of an employee on average 6 per day.
- Searching details of a given product, on average 2000 per day.
- Searching details of a specific product, on average 500 per day.
- Searching details of a given customer, on average 50 per day.
- Searching details of a given supplier, on average 2 per day.
- Searching details of a given order, on average 300 per day.

### Networking and shared access requirements

- The eMart system must be securely networked to a centralized database located off-site at a location not yet determined, with a proxy server located on site.
- The system should allow for at least 500 users to be active concurrently.

### Performance

- The system is expected to run within a 3 second response for single searches at peak times (7am-11:59pm) and 1 second at non-peak times.
- The system is expected to run within a 5 second response for multiple searches at peak times (7am-11:59pm) and 3 second at non-peak times.
- The system is expected to run within a 3 second response for all updates/saves during peak times (7am-11:59pm) and 1 second at non-peak times.

### Security

- The database should be password protected.
- Employees will be assigned database access related to a appropriate user view Organisation owner, Manager, Supervisors, Department Managers, Merchandiser, Assistants, Software developer, System administrator.
- Customers will be assigned a unique user-name with a password consisting of 6-12 case-sensitive characters (numbers, letters and underscore only).
- All monetary transactions to be provided by a secure online payment scheme which will be provided by paypal.com.
- All users of the system should only see data that is appropriate to their task.

## User interface

- The user interface will be menu-driven, with lots of useful hints to help novice users navigate the site with ease.

## 5.7 Functional Requirements

### 5.7.1 The system must provide a means for a customer to access the website (shop front).

**Objective:** *To provide a website that customers can access without having to create an account with eMart.*

**Priority:** High

#### 5.7.1.1 The system must provide a means for a customer to browse products.

**Objective:** *The customer must be able to view the eMart website and browse products that are currently in stock.*

**Priority:** High

#### 5.7.1.2 The system must provide a means for a customer to search for an item.

**Objective:** *The customer must be able to type keywords into the search engine and search for a specific product.*

**Priority:** Low

#### 5.7.1.3 The system must provide a means for a customer to add items to a cart.

**Objective:** *The customer must be able to add product(s) to the cart when they find items they want.*

**Priority:** High

#### 5.7.1.4 The system must provide a means for a customer to view a cart.

**Objective:** *The customer must be able to view the items that they've added to the cart.*

**Priority:** High

#### 5.7.1.4.1 The system must provide a means for a customer to delete an item from a cart.

**Objective:** *The customer must be able to delete any items from the cart which they no longer need.*

**Priority:** High

#### 5.7.1.4.2 The system must provide a means for a customer to update the quantity of any items in a cart.

**Objective:** *The customer must be able update the quantities in the cart whilst viewing.*

**Priority:** Low

### 5.7.2 The system must provide a means for a customer to sign in to the website

**Objective:** *To allow the customer to sign in and access other features of the website.*

**Priority:** High

#### 5.7.2.1 The system must provide a means for a customer to register their details with the organisation.

**Objective:** *The customer must be able to register with eMart.*

**Priority:** High

- 5.7.2.2 **The system must provide a means for a customer to place an order.**  
**Objective:** *The customer must be able to place an order with eMart.*  
**Priority:** High
- 5.7.2.3 **The system must provide a means for a customer to view their order history.**  
**Objective:** *The customer shall be able to view any previous orders that they have placed with eMart.*  
**Priority:** Low
- 5.7.2.4 **The system must provide a means for a customer to update their details.**  
**Objective:** *The customer shall be able to update their personal details after they've registered with eMart.*  
**Priority:** Medium
- 5.7.3 **The system must provide a means for specified employees to access customer details.**  
**Objective:** *To allow certain employees with the appropriate access rights to access any customer details.*  
**Priority:** High
- 5.7.3.1 **The system must provide a means for specified employees to locate customer(s).**  
**Objective:** *To allow certain employees with the appropriate access rights to locate customers and view their details.*  
**Priority:** Medium
- 5.7.3.1.1 **The system must provide a means for the manager to update customer details.**  
**Objective:** *The manager must be able to update any customer details when necessary.*  
**Priority:** Medium
- 5.7.3.1.2 **The system must provide a means for the manager to delete customer details.**  
**Objective:** *The manager must be able to delete any customer from the eMart system.*  
**Priority:** Medium
- 5.7.4 **The system must provide a means for specified employees to access the orders database.**  
**Objective:** *To allow employees with the appropriate access rights to access the orders database.*  
**Priority:** High
- 5.7.4.1 **The system must provide a means for supervisors to place orders.**  
**Objective:** *Supervisors must be able to place customer orders if the customer is unable to do so through the website.*  
**Priority:** Medium

**5.7.4.2 The system must provide a means for specified employees to locate order(s).**

**Objective:** *Employees with the appropriate access rights must be able to locate any order(s).*

**Priority:** Medium

**5.7.4.2.1 The system must provide a means for supervisors to update orders.**

**Objective:** *Supervisors must be able to update any order information.*

**Priority:** High

**5.7.4.2.2 The system must provide a means for supervisors to delete orders.**

**Objective:** *Supervisors must be able to delete orders when they are cancelled*

**Priority:** Low

**5.7.4.3 The system must provide a means for specified employees to update the status of an order.**

**Objective:** *To allow certain employees with the appropriate access rights to update an order's status when an order gets further along in the process.*

**Priority:** Medium

**5.7.5 The system must provide a means for employees to access the HR Management database.**

**Objective:** *To allow all employees to access the HR Management database.*

**Priority:** High

**5.7.5.1 The system must provide a means for the manager and the department managers to access the employee database.**

**Objective:** *The manager and the department managers must be able to access eMart's employee database.*

**Priority:** High

**5.7.5.1.1 The system must provide a means for the manager and the department managers to add employees.**

**Objective:** *The manager and department managers must be able to add new employees of eMart to the HR management database.*

**Priority:** High

**5.7.5.1.2 The system must provide a means for the manager and the department managers to locate employees.**

**Objective:** *The manager and department managers must be able to locate any employees of eMart and view their information.*

**Priority:** Medium

**5.7.5.1.2.1 The system must provide a means for the manager and the department managers to update employee details.**

**Objective:** *The manager and department managers must be able to update any record on the eMart employee database.*

**Priority:** Medium

- 5.7.5.1.2.2 **The system must provide a means for the manager and the department managers to delete employees.**  
**Objective:** *The manager and department managers must be able to delete any record in the eMart employee database.*  
**Priority:** Medium
- 5.7.6 **The system must provide a means for employees to access the rota database.**  
**Objective:** *Employees must be able to access eMart's employee rota.*  
**Priority:** Low
- 5.7.6.1 **The system must provide a means for employees to view the rota data.**  
**Objective:** *All employees of eMart must be able to access the HR Management database to view their individual rota.*  
**Priority:** Low
- 5.7.6.2 **The system must provide a means for the manager and the department managers to add rota data.**  
**Objective:** *The manager and department managers must be able to add data to the eMart employee rota.*  
**Priority:** Medium
- 5.7.6.3 **The system must provide a means for the manager and the department managers to update rota data.**  
**Objective:** *The manager and department managers must be able to update the eMart employee rota.*  
**Priority:** Medium
- 5.7.6.4 **The system must provide a means for the manager and the department managers to delete rota data.**  
**Objective:** *The manager and department managers must be able to delete the eMart employee rota.*  
**Priority:** Low
- 5.7.7 **The system must provide a means for the manager to access the department database.**  
**Objective:** *The manager must be able to access the eMart department database.*  
**Priority:** Low
- 5.7.7.1 **The system must provide a means for the manager to add a department.**  
**Objective:** *The manager must be able to add any new department to eMart's department database.*  
**Priority:** Low
- 5.7.7.2 **The system must provide a means for the manager to locate a department.**  
**Objective:** *The manager must be able to locate any department in eMart's department database.*  
**Priority:** Low
- 5.7.8 **The system must provide a means for the manager to access the payroll.**  
**Objective:** *The manager must be able to access eMart's payroll database.*

**Priority:** High

- 5.7.8.1 **The system must provide a means for the manager to run the payroll.**

**Objective:** *The manager must be able to run eMart's payroll, which automatically calculates the wages for employees.*

**Priority:** High

- 5.7.8.2 **The system must provide a means for the manager to create a payroll report.**

**Objective:** *The manager must be able to create a payroll report for any stated period of time.*

**Priority:** Low

- 5.7.9 **The system must provide a means for the system administrator to access the user database.**

**Objective:** *The system administrator must be able to access the eMart user database.*

**Priority:** High

- 5.7.9.1 **The system must provide a means for the system administrator to create a new user.**

**Objective:** *The system administrator must be able to create a new user for eMart's user database.*

**Priority:** High

- 5.7.9.2 **The system must provide a means for the system administrator to update a password.**

**Objective:** *The system administrator must be able to update an eMart's user's password for the user database.*

**Priority:** Low

- 5.7.9.3 **The system must provide a means for the system administrator to delete a user.**

**Objective:** *The system administrator must be able to delete a user from eMart's user database.*

**Priority:** Low

- 5.7.10 **The system must be able to keep accurate log files.**

**Objective:** *The system must log complete and accurate information regularly.*

**Priority:** Medium

- 5.7.10.1 **The system must update the 'access' log when required.**

**Objective:** *The system must log any unauthorized access attempts to the database to track any malicious behaviour by users.*

**Priority:** Medium

- 5.7.10.2 **The system must be able to update the 'error' log.**

**Objective:** *The system must log any exceptions caught during runtime.*

**Priority:** Medium

- 5.7.10.3 **The system administrator must be able to access the database 'access' log.**

**Objective:** *The system administrator must be able to view the entries in the 'access' log.*

**Priority:** Medium

5.7.10.4 The system administrator must be able to access the ‘error’ log.

**Objective:** *The system administrator must be able to access the systems ‘error’ log.*

**Priority:** Medium

**5.7.11 The system must accurately keep a record of stock.**

**Objective:** *To provide an improvement to the current manual system in use by the organization.*

**Priority:** High

**5.7.11.1 The system must fire an item expiry date warning.**

**Objective:** *The system must be able to initiate a warning when an item is close to its expiry date.*

**Priority:** High

**5.7.11.2 The system must fire a low stock level warning.**

**Objective:** *The system must be able to initiate a warning when an item falls below a certain stock level so that efforts can be made to replenish stock.*

**Priority:** High

**5.7.11.3 The system must include a bar-code scanning facility which can decode the information of any product that needs entering into the database.**

**Objective:** *All incoming and outgoing stock will be recognized by the system after its barcode is scanned so as to increase ease and efficiency.*

**Priority:** Medium

**5.7.12 The system must provide a means for the merchandiser to access the supplier delivery database.**

**Objective:** *The merchandiser must be able to access eMart’s database that contains all the information for deliveries from suppliers.*

**Priority:** High

**5.7.12.1 The system must provide a means for the merchandiser to add a new supplier delivery.**

**Objective:** *The merchandiser must be able to add any new deliveries that eMart are receiving from suppliers.*

**Priority:** High

**5.7.12.2 The system must provide a means for the merchandiser to locate a supplier delivery.**

**Objective:** *The merchandiser must be able to locate any supplier deliveries in the database.*

**Priority:** Medium

**5.7.12.2.1 The system must provide a means for the merchandiser to update a supplier delivery.**

**Objective:** *The merchandiser must be able to update any details on a supplier delivery.*

**Priority:** Medium

5.7.12.2.2 **The system must provide a means for the merchandiser to delete a supplier delivery.**

**Objective:** *The merchandiser must be able to delete any supplier delivery.*

**Priority:** Low

5.7.13 **The system must provide a means for the merchandiser to access the supplier database.**

**Objective:** *The merchandiser must be able to access eMart's database that contains the information for all the suppliers.*

**Priority:** High

5.7.13.1 **The system must provide a means for the merchandiser to add a new supplier.**

**Objective:** *The merchandiser must be able to add any new suppliers that eMart are using.*

**Priority:** High

5.7.13.2 **The system must provide a means for the merchandiser to locate a supplier.**

**Objective:** *The merchandiser must be able to locate any supplier in the database.*

**Priority:** Medium

5.7.13.2.1 **The merchandiser must be able to update a supplier.**

**Objective:** *The merchandiser must be able to update any details on a supplier.*

**Priority:** High

5.7.13.2.2 **The merchandiser must be able to delete a supplier.**

**Objective:** *The merchandiser must be able to delete any supplier from the database.*

**Priority:** Medium

5.7.14 **The system must provide a means for specified employees to access the stock database.**

**Objective:** *To allow employees with the appropriate access rights to access the database containing details of all items that eMart stocks.*

**Priority:** High

5.7.14.1 **Employees with the correct access rights must be able to add item(s).**

**Objective:** *Certain employees must be able to add details of new items that eMart stocks.*

**Priority:** High

5.7.14.2 **The system must provide a means for specified employees to locate item(s).**

**Objective:** *To allow employees with the appropriate access rights to locate and view details of items that eMart stocks.*

**Priority:** Medium

5.7.14.2.1 **Employees with the correct access rights must be able to update item(s).**

**Objective:** *Certain employees must be able to update details of items that eMart stocks.*

**Priority:** High

5.7.14.2.2 **Employees with the correct access rights must be able to delete item(s).**

**Objective:** Certain employees must be able to delete items that eMart no longer stocks.

**Priority:** Medium

5.7.15 **The system must provide a means for the owner and manager to access the account database.**

**Objective:** *The manager and the owner must be able to access the database containing details of all eMart's accounts.*

**Priority:** High

5.7.15.1 **The owner and manager must be able to add account(s).**

**Objective:** *The owner and the manager must be able to add details of new accounts of eMart.*

**Priority:** High

5.7.15.2 **The owner and manager must be able to locate account(s).**

**Objective:** *The owner and the manager must be able to locate and view details of eMart's accounts.*

**Priority:** Medium

5.7.15.2.1 **The owner and manager must be able to update account(s).**

**Objective:** *The owner and the manager must be able to update details of eMart's accounts.*

**Priority:** High

5.7.15.2.2 **The owner and manager must be able to delete account(s).**

**Objective:** *The owner and the manager must be able to delete details of eMart's accounts which are no longer in use.*

**Priority:** Medium

5.7.16 **The system must provide a secure means of logging in.**

**Objective:** *All employees of eMart must be able to login into the system.*

**Priority:** High

5.7.17 **The system must provide a means of logging out of every part of the system.**

**Objective:** *All employees of eMart must be able to logout of the system.*

**Priority:** High

5.7.18 **The system must provide a means of processing refunds.**

**Objective:** *To allow the efficient and accurate processing of customer refunds.*

**Priority:** High

**5.7.19 The system must provide a suitable means of data input.**

**Objective:** *The means of input should be simple enough to learn with 30 minutes training.*

**Priority:** High

**5.7.19.1 The system should hide the complexities of the operating system and should provide an easy to use interface.**

**Objective:** To make it unnecessary for employees to learn how to use the workings of the operating system in addition to the database system.

**Priority:** High

**5.7.20 The system must try to prevent human error with the stock control and have a way to correct errors if they are made.**

**Objective:** *To keep the system accurate and minimize human errors as much as possible.*

**Priority:** Medium

**5.7.21 The system must provide a means to print any reports created.**

**Objective:** *The reports can be printed and used however the user needs.*

**Priority:** Low

**5.7.22 The system should provide a backup of the database, so that after a system failure, data can be recovered almost immediately with minimal loss of data.**

**Objective:** *A backup is required to recover from system failure where data is lost. A backup should be made daily, so any loss is minimised.*

**Priority:** High

## 5.8 Non-functional Requirements

### 5.8.1 Audit-ability

5.8.1.1 Record in the database log when a user or employee enters their password incorrectly more than three times.

**Priority:** Medium

5.8.1.2 Automatically send audit of database log to the system administrator every day to check for inconsistencies.

**Priority:** Medium

### 5.8.2 Availability

5.8.2.1 The system shall be available to customers 24/7. Maximum downtime allowance of five seconds during off peak times.

**Priority:** High

5.8.2.2 The system shall be available to employees during working hours (8.00-19.00). Downtime shall not exceed five seconds during this period.

**Priority:** High

- 5.8.2.3 Routine system maintenance is conducted every Sunday, between 1:00 and 3:00.  
**Priority:** Medium
- 5.8.2.4 The software shall only be available to authorized users of the machine within the warehouse.  
**Priority:** High

#### 5.8.3 Back-Up and Restore

- 5.8.3.1 The system administrator should perform data backups daily at 1:00 using back-up software; Bacula.  
**Priority:** Medium
- 5.8.3.2 Use hot back up to back up as it is a multi-user system that is actively accessible.  
**Priority:** Medium
- 5.8.3.3 Only the system administrator shall be allowed access to retrieve restoration after getting authorization from the store manager to perform back up or restoration.  
**Priority:** High

#### 5.8.4 Capacity

- 5.8.4.1 The database has a maximum capacity of 1TB.  
**Priority:** Medium
- 5.8.4.2 A maximum of 200 users may access the database at one given time.  
**Priority:** Medium
- 5.8.4.3 No more than 50 customers may place orders simultaneously online.  
**Priority:** Medium

#### 5.8.5 Data Integrity

- 5.8.5.1 Stock updates must be altered to customers view in less than 2 seconds.  
**Priority:** High
- 5.8.5.2 Stock updates and prices should be kept accurate and updated within 1 hour of an employee acknowledging changes are needed.  
**Priority:** High

#### 5.8.6 Data Retention

- 5.8.6.1 Database log files are saved for 5 years from date of archive.  
**Priority:** Low
- 5.8.6.2 There is 2 TB memory allocated for archives to be stored.  
**Priority:** Medium

#### 5.8.7 Development environment

- 5.8.7.1 All sections of warehouse that need access to database shall have a new PC installed running Windows 7.  
**Priority:** Low
- 5.8.7.2 All PC's shall have the latest version of Java SDK installed.  
**Priority:** High
- 5.8.7.3 All PC's shall have Bacula back-up software installed.  
**Priority:** Medium
- 5.8.7.4 All PC's shall have up to date anti-virus and firewall software installed.  
**Priority:** High
- 5.8.7.5 All PC's shall have Wi-Fi installed.

**Priority:** Low

#### 5.8.8 Documentation

- 5.8.8.1 KISACASA delivers all software modules produced for the project to e-Mart on final client sign-off of the project.

**Priority:** High

- 5.8.8.2 KISACASA should provide a PDF copy and hardcopy user guide manual for use by all employees.

**Priority:** Low

#### 5.8.9 Efficiency

- 5.8.9.1 Employees shall be able to use all of their allocated system functions after 30 minutes of training. After this training, the average number of errors made by experienced users shall not exceed one per hour.

**Priority:** High

- 5.8.9.2 Each customer order shall be processed from the client's PC to the warehouse database in less than 5 seconds.

**Priority:** Low

#### 5.8.10 Hardware

- 5.8.10.1 eMart shall provide electricity and internet access to the system 24/7.

**Priority:** Medium

- 5.8.10.2 eMart is responsible for ensuring all utilities and hardware is installed by 1<sup>st</sup> May 2012.

**Priority:** Low

- 5.8.10.3 eMart is responsible for providing Wi-Fi to the warehouse by 1<sup>st</sup> May 2012.

**Priority:** Low

#### 5.8.11 Install-ability

- 5.8.11.1 Software should be installed on each system in maximum 20 minutes.

**Priority:** Low

- 5.8.11.2 All software should be installed manually by eMart's system administrator.

**Priority:** High

- 5.8.11.3 Future installations should be installed automatically during out of work hours.

**Priority:** Low

- 5.8.11.4 The software should be installed and tested initially on 15 PC's within the warehouse.

**Priority:** Low

- 5.8.11.5 Anti-virus software to be installed and tested by the system administrator before the software is used by employees.

**Priority:** Medium

- 5.8.11.6 The client software installation instructions are written for Windows 7 environment.

**Priority:** Medium

#### 5.8.12 Legal

- 5.8.12.1 The application complies with the Data Protection Act 1998 (*see section 1.3*)

**Priority:** High

- 5.8.12.2 All employee bank details shall be encrypted within the system.  
**Priority:** High
- 5.8.12.3 All employee personal details shall be available to as few employees as possible.  
**Priority:** High
- 5.8.12.4 A license shall be obtained when the system is finalised and KISACASA shall own the rights to the software.  
**Priority:** High
- 5.8.12.5 eMart will ensure a local building inspectors company should inspect all wiring installations by 5<sup>th</sup> May.  
**Priority:** Low

#### 5.8.13 Maintainability

- 5.8.13.1 All hardware problems are reported and repaired by system administrator.  
**Priority:** High
- 5.8.13.2 Anti-virus and firewall protection is updated automatically on every PC on the network.  
**Priority:** Medium
- 5.8.13.3 When the system is down it should be repaired by system administrator within two hours during working hours. Outside of working hours it should be repaired within twelve hours.  
**Priority:** Medium

#### 5.8.14 Operation Environment

- 5.8.14.1 An alert is triggered when the utilization of the server exceeds 85%.  
**Priority:** Low
- 5.8.14.2 In the event of the unavailability of the Web server or database server an alert email is sent to the system administrator.  
**Priority:** High
- 5.8.14.3 The following environment is required to support the system Web pages:
  - 5.8.14.3.1 Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7  
**Priority:** High
  - 5.8.14.3.2 Internet Explorer 5.x or above  
**Priority:** High
  - 5.8.14.3.3 Optimized for 1024x768  
**Priority:** High
  - 5.8.14.3.4 A development package such as Adobe Dreamweaver to support the update of web page content.  
**Priority:** Low

#### 5.8.15 Performance

- 5.8.15.1 The maximum response time during peak periods (16:00-20:00) should be 10 seconds whilst processing orders.  
**Priority:** Medium
- 5.8.15.2 All customers in Britain should have the same response time on the system regardless of location.  
**Priority:** Low
- 5.8.15.3 The system should respond to user inputs almost instantly within the warehouse.

**Priority:** Medium

- 5.8.15.4 The system administrator shall monitor the network load for two weeks after implementation.

**Priority:** High

- 5.8.15.5 Almost every operation involves a confirmation check to prevent inaccuracy.  
**Priority:** High

## 5.8.16 Privacy

- 5.8.16.1 (See section 5.8.12)

## 5.8.17 Reliability

- 5.8.17.1 User input should be limited so that it is impossible for invalid or unnecessary inputs to be processed.

**Priority:** High

- 5.8.17.2 Each user shall have a unique login and password to remove the risk of unauthorized access.

**Priority:** High

- 5.8.17.3 Users can only access areas allocated to their username to ensure reliability.

**Priority:** High

- 5.8.17.4 No more than 1 failure per week is expected.

**Priority:** High

- 5.8.17.5 The acceptable recovery time for failures is 30 minutes.

**Priority:** High

## 5.8.18 Recoverability

- 5.8.18.1 In the event of a power failure all systems shall be running efficiently again within 2 hours.

**Priority:** Medium

- 5.8.18.2 In the event of a product failure all systems should be running efficiently again within 4 hours.

**Priority:** Medium

- 5.8.18.3 In the event of lost data all systems should be running efficiently again within 1 day.

**Priority:** Medium

- 5.8.18.4 In the event of sabotage all systems should be running efficiently again within 2 days.

**Priority:** Medium

- 5.8.18.5 In the event of acts of nature all systems should be running efficiently again within 3 days.

**Priority:** Medium

- 5.8.18.6 In the event of acts of war, terrorism or vandalism all systems should be running efficiently again within 7 days.

**Priority:** Medium

- 5.8.18.7 Recoverability is the responsibility of the system administrator.

**Priority:** High

## 5.8.19 Robustness

- 5.8.19.1 Data validation error messages are explained in detail in the user guide to help the user determine the problem.

**Priority:** Medium

- 5.8.19.2 All system failures shall be logged and tracked until resolved.  
**Priority:** Medium
- 5.8.19.3 For severe failures the system administrator shall be sent an email notification immediately.  
**Priority:** Low
- 5.8.19.4 For minor failures the system administrator shall be sent an email notification automatically.  
**Priority:** Low

#### 5.8.20 Security

- 5.8.20.1 The software shall be handled only by the administrator and authorized users.  
**Priority:** High
- 5.8.20.2 Only the system administrator has the power to assign permission, by creating accounts and generating the access to each account.  
**Priority:** High
- 5.8.20.3 Each user shall be allocated a personal 6 digit ID number.  
**Priority:** High
- 5.8.20.4 Each user shall choose a personal 6 digit password.  
**Priority:** High
- 5.8.20.5 Each user shall update their password every 60 days by visiting the system administrator; the new password must not have been used previously.  
**Priority:** Low
- 5.8.20.6 No employee other than the system administrator should have access to the back-up system.  
**Priority:** High

#### 5.8.21 Training

- 5.8.21.1 (See section 5.8.9)

#### 5.8.22 Usability

- 5.8.22.1 The user interface shall be very simple and self-explanatory.  
**Priority:** High
- 5.8.22.2 All employees shall interact with the system using keyboards and mouse; with output displayed on LCD screens.  
**Priority:** High
- 5.8.22.3 Each keyboard should be fitted with a gel wrist rest pad to reduce RSI.  
**Priority:** Low
- 5.8.22.4 Near each PC a user guide with step by step instructions should be placed.  
**Priority:** Low
- 5.8.22.5 A novice computer user should be able to use the website after two minutes of browsing time efficiently. An intermediate to expert computer user should be able to use the website to its full capacity immediately.  
**Priority:** High

### 5.9 Prioritisation of Requirements

The requirements have been given a rating of high, medium or low. Descriptions of what these ratings mean follows:

- **High** – the system will fail if this requirement is not implemented.
- **Medium** – there is clear added value to having this in the requirement.

- **Low** – this is a nice added extra to the system but definitely not vital to the functioning of the system.

## 5.10 Success Criteria for Requirements

In order for the system to be considered a success, the following criteria must be met:

- **The system should increase productivity**
  - It should make stock operations quicker than the current manual system
- **The system should be easy to use**
  - All employees should be able to use it within 30 minutes of training
  - No additional employees need to be employed
- **The system should be secure**
  - A username and password must be entered to access and use the system
  - Different users should have different access rights
- **The system should be reliable**
  - The system should not crash more than once a month
  - The system should keep a backup database so no information is lost
- **The system should be fast**
  - It should take no more than 30 seconds to start up
  - It should take less than 3 seconds to return a search result from the database

**Appendix B**

# **DESIGN DOCUMENT**

---

## 6. Conceptual Database Design

There is a lot more to an e-commerce solution than just the user side of things (website). There are also all the aspects of managing the warehouse and inventory. The right tools to manage the e-commerce solution are as critical as the right tools to engage the site visitor. The backbone of any interactive website is a well-designed database.

The first stage of the design process will be the design of the high-level relational data schema for the system. Subsequently, the field structure for each table will be designed in-depth.

### 6.1 Business Rules

#### *Orders and Stock*

- Delivery time and date can only be during business hours.
- New orders cannot be created for discontinued menu items.
- Order quantities cannot be greater than current stock level or less than one or greater than 15.
- New orders cannot be created for discontinued items.
- Each order contains one or many ordered items and each ordered item belongs to only one order.
- When current stock level goes below the ‘reorder’ status, email the merchandiser to reorder stock.
- Stock levels are altered once order is finalised.
- Stock is split into relevant categories containing no more than 5000 separate products of stock.

#### *Customers*

- Customers must be over 18 years of age and have a valid Paypal account.
- It is the customer’s responsibility to provide correct payment and delivery details.
- It is the customer’s responsibility to check details of the confirmatory email are correct as soon as possible and print a copy to keep for themselves.

#### *Staff*

- All staff must be over 16 years of age.
- As few as possible system administrators shall control the eMart System.
- Only the system administrator shall have the privilege to create staff accounts which shall be created using the eMart system.
- Only the system administrator shall have the privileges to be able to GRANT privileges to staff.
- Only staff with granted privilege shall have access to customer account.
  - All staff with privilege shall sign a disclaimer form stating that they are aware of the Data Protection Act and will only use customer information for relevant tasks.
- Only staff with granted privilege shall have the ability to update orders.
- Only staff with granted privilege shall have the ability to ship orders.
- Each staff member is a part of one or several departments
- One member of staff shall handle one order.

#### *Promotional Goods*

- No customer may have more than 6 quantity of promotion stock.
- All offers are subject to availability and whilst stocks last.

#### *Pricing*

- If an error is discovered on the price of any goods which a customer has order they will be informed as soon as possible and given the option of reconfirming their order or cancelling it.

- If we are unable to contact the customer the order will be cancelled. If payment has already been processed a full refund will be issued.
- Where applicable, prices are inclusive of VAT at the appropriate rate to the country of receipt.

### *Redeliveries*

- If an item is missing from a delivery the customer should contact eMart as soon as possible.

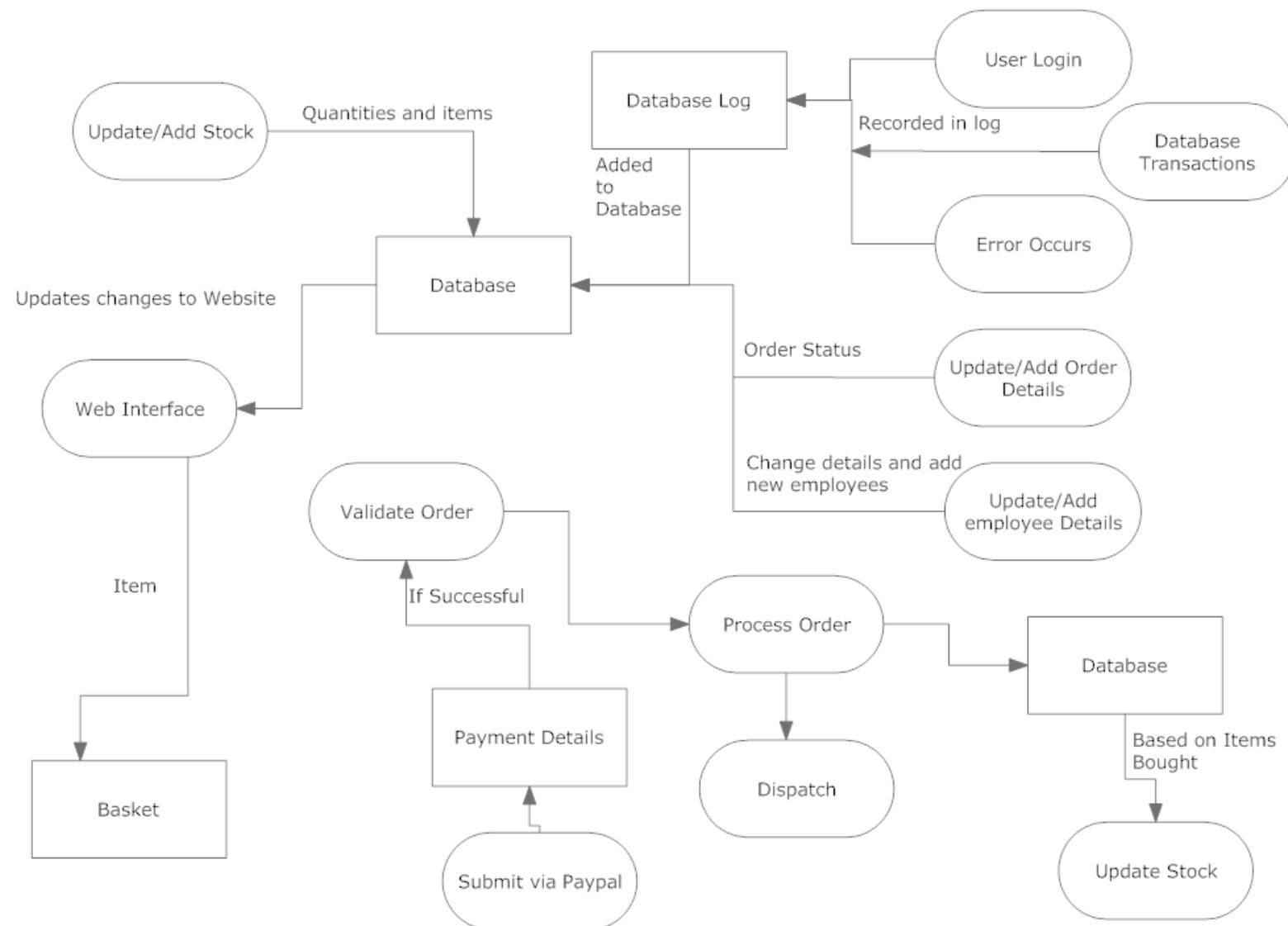
### *Customer Delivery*

- eMart shall always aim to deliver your goods within the customers designated delivery slot. However eMart does not accept liability if we deliver goods outside of this slot or don't deliver all or any of the goods in your order.
- eMart will only deliver to an address stored in a customer's account in the 'Your Account' section of the website.
- Deliveries are only available within British shores and within a certain range.

### *Refunds*

- Other than items with a noted mark on their page (being non-returnable by which their nature may not be returned; statutory rights are not affected) any item may be returned for a full refund within 14 days.
- On receipt of the returned item stock levels shall be readjusted by staff manually and a full refund for item shall be issued immediately.
- Item return delivery is organised and paid for by the customer.
- eMart shall not pay any unpaid delivery costs for returned items.
- Refunds for promotional stock will be based upon the terms of the promotional price. Statutory rights are not affected.
- If a customer receives goods which were not ordered or goods that are damaged or defective or a different quality than expected eMart shall make good any shortage or non-delivery, replace or repair any damaged or defective goods or refund to the account via Paypal for the amount paid for the goods.

## 6.2 Data Flow Diagram



### 6.3 Functional Descriptions

#### *Stock Control*

In order for the system to know that items of stock need reordering, there needs to be level definitions for each item, which also takes into consideration the time it takes for the supplier to deliver the item. As eMart specialises in local produce, this system needs to be particularly accurate as local suppliers might not have as much stock available at short notice as larger commercial suppliers. The system will have 4 statuses, “Stocked”, “Medium”, “Reorder”, and “OutOfStock”. “Stocked” means that the quantity of items available is ample and has most likely recently been restocked. “Medium” will mean that the product is above the ideal quantity level but it is getting increasingly closer to it. “Reorder” means the item has just dropped below the ideal quantity level and the system will contact the merchandiser (via email) to reorder the item from the relevant supplier. In theory, the item should not deplete to the “OutOfStock” status since the system should flag the notification within sufficient time so that the item can be reordered. However, if the merchandiser has problems replenishing stock levels then products will become low in stock. This will hopefully be a rare occasion.

#### *Expiry Date Warning System*

In order to ensure the ultimate freshness of eMart goods, a system has to be made to ensure that all goods which are delivered to customers homes are within a suitable time frame of their expiry date in order to consume. This system will increase customer satisfaction. eMart’s expiry policy will be that if fresh goods come within 2 days of their expiry date then a message will be sent to the department manager to remove these items and to transfer them to the shop where if they are not sold they will be put as ‘reduced’ until their expiry date where they will consequently be disposed of. For preserved goods, e.g. tins, they will be transferred to the shop when they are within 1 week of their expiry date. The department manager will make sure that the goods are decremented accordingly in the stock database so that stock control consistency is maintained.

*NOTE: not all goods in the physical shop are surplus from the warehouse. They would also have their own stock control system.*

## 7. Logical Database Design

### 7.1 ER Model

#### 7.1.1 Entity Identification

Entities are identified through use cases and user views

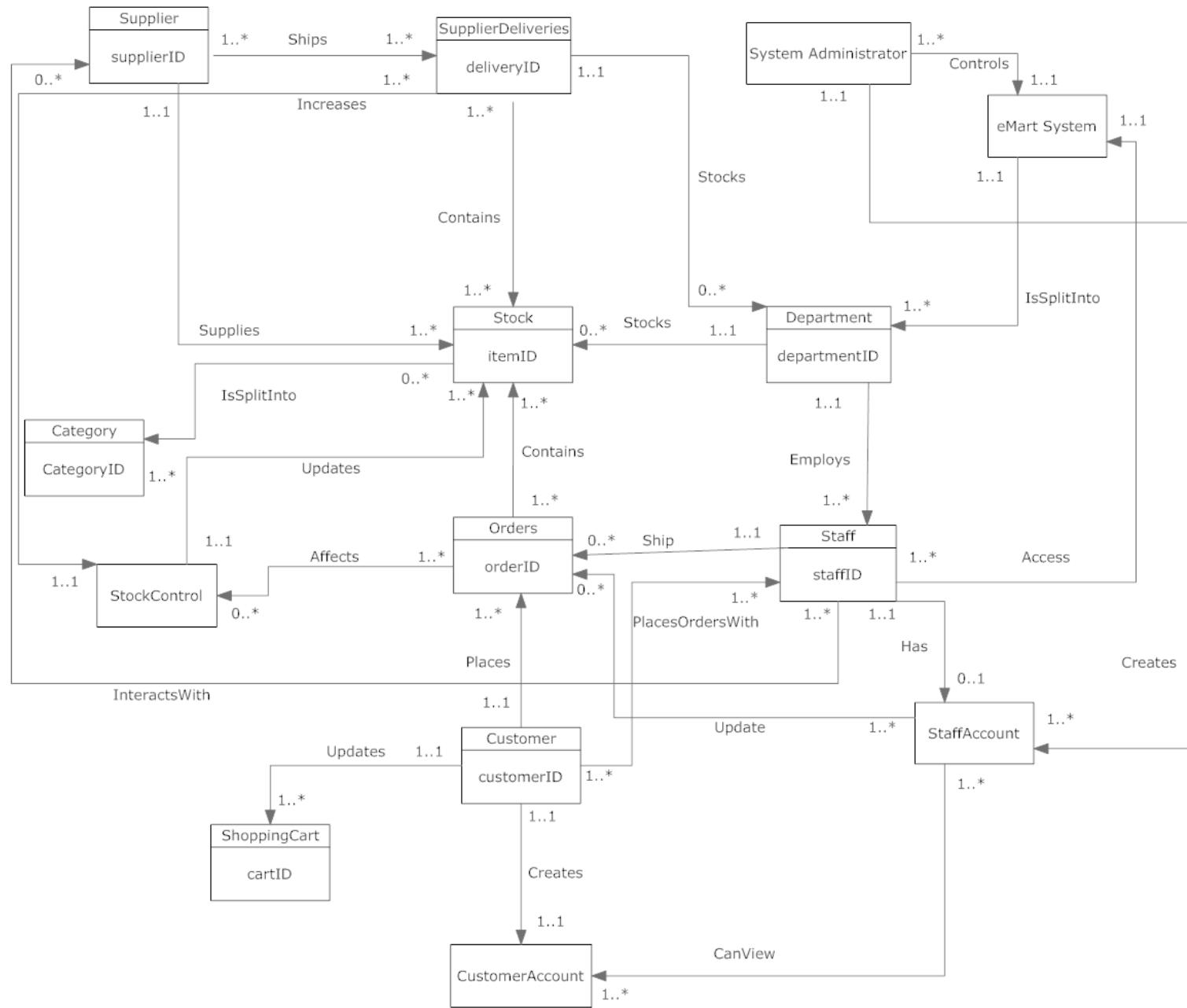
- System Administrator
- eMart System
- Department
- Staff
- Staff Account
- Customer
- Customer Account
- Shopping Cart
- Orders
- Stock Control
- Stock
- Category

- Supplier
- Supplier Deliveries

### 7.1.2 Relationship Identification

Entity	Multiplicity	Relationship	Multiplicity	Entity
<b>System Administrator</b>	1..*	Controls	1..1	eMart System
<b>System Administrator</b>	1..1	Creates	1..*	Staff Accounts
<b>eMart System</b>	1..1	IsSplitInto	1..*	Department
<b>Department</b>	1..*	Stocks	0..*	Stock
<b>Department</b>	1..1	Employs	1..*	Staff
<b>Stock</b>	1..*	IsSplitInto	1..1	Catagory
<b>Supplier</b>	1..1	Supplies	1...*	Stock
<b>Supplier</b>	0..*	Ships	1..*	Supplier Deliveries
<b>Supplier Deliveries</b>	1..*	Stocks	0..*	Department
<b>Orders</b>	1..*	Contains	1..*	Stocks
<b>Orders</b>	1..*	Affects	0..*	Stock Control
<b>Stock Control</b>	1..1	Updates	1..*	Stock
<b>Customer</b>	1..*	Places	1..1	Orders
<b>Customer</b>	1..1	Creates	1..1	Customer Account
<b>Customer</b>	1..*	PlacesOrdersWith	1..*	Staff
<b>Customer</b>	1..1	Updates	1..*	ShoppingCart
<b>Staff</b>	1..1	Ship	0..*	Orders
<b>Staff</b>	1..*	Access	1..1	eMart System
<b>Staff</b>	1..1	Has	1..1	Staff Accounts
<b>Staff Accounts</b>	1..*	CanView	1..*	Customer Accounts
<b>Staff</b>	1..*	InteractsWith	0..*	Suppliers
<b>Supplier Deliveries</b>	1..*	Increases	1..1	Stock Control
<b>Staff Accounts</b>	1..*	Updates	0..*	Orders

### 7.1.3 ER Diagram



#### 7.1.4 Interaction Diagrams

##### **Activity Diagram for customer login and update**

Allows a customer to login and update user details after a successful login attempt.

### **Activity Diagram for customer searching/browsing activity**

Customer can search for a specific item or items in a specific product category. The item can be viewed if available, if not available, the customer is able to search again for other items/products.

### **Activity Diagram for customer checkout**

The customer can enter the address for delivery of items, the customer can then pay for items by either entering credit card details or retrieving details already stored, the details are then verified.

### 7.1.5 Sequence Diagrams

The following diagrams show how and in what order processes interact with each other in the eMart system. Each diagram shows an actor and the goal they want to achieve. In order to achieve their goal, they must send a number of messages through the eMart system between themselves, the database and the server. Each message is annotated to show what is happening at a particular time.















## 7.2 Table Structures

**Department**(departmentID, departmentName, departmentManager, telephoneNumber)

Primary Key departmentID

Foreign key departmentManager references Staff

**UserLogin**(userID, staffID, password, lastLogDate)

Primary Key userID

Foreign Key staffID references Staff

**Staff**(staffID, firstName, surname, jobDescription)

Primary Key staffID

**Stock**(itemID, name, briefDescription, longDescription, supplierID, subcategoryID, price, attributeID, thumbImage, bigImage)

Primary Key itemID

Foreign Key supplierID references Suppliers

Foreign Key subcategoryID references Subcategory

Foreign Key attributeID references Attribute

**StockControl**(itemID, itemExpiryDate, idealQuantity, currentQuantity, status, departmentID)

Primary Key itemID

Foreign Key departmentID references Department

**Category**(categoryID, categoryName)

Primary Key categoryID

**Subcategory**(subcategoryID, categoryID, subcategoryName, description, image)

Primary Key subcategoryID

Foreign Key categoryID references Category

**Suppliers**(supplierID, supplierName, emailAddress, contactID)

Primary Key supplierID

Foreign Key contactID references ContactDetails

**SupplierDeliveries**(deliveryID, supplierID, dateOfDelivery, timeOfDelivery, itemID)

Primary Key deliveryID

Foreign Key supplierID references Suppliers

Foreign Key itemID references Stock

**Customer**(customerID, firstName, surname, registrationDate, emailAddress, contactID, password)

Primary Key customerID

Alternate Key emailAddress

Foreign Key contactID references ContactDetails

**ContactDetails**(contactID, addressLineOne, town, city, county, postcode, telephoneNumber)

Primary Key contactID

Alternate Key postcode

**Orders**(orderID, customerID, paymentDate, orderStatus, promoCode, orderNotification)

Primary Key orderID

Foreign Key customerID references Customer

Foreign Key promoCode references PromoDiscounts

**ShoppingCart**(cartID, itemID, cartSessionID, cartQuantity, cartDate, promoCode)

Primary Key cartID

Foreign Key itemID references Stock

Foreign Key promoCode references PromoDiscounts

**AttributeCategory**(attributeCategoryID, attributeGroupName)

Primary Key attributeCategoryID

**Attribute**(attributeID, attributeCategoryID, attributeName)

Primary Key attribute ID

Foreign Key attributeCategoryID references AttributeCategory

**OrderItem**(orderID, itemID, orderQuantity, promoCode)

Primary Key (orderID, itemID)

Foreign Key orderID references Orders

Foreign Key itemID references Stock

Foreign Key promoCode references PromoDiscounts

**ShopConfig**(name, contactID, emailAddress, shippingCost)

Foreign Key contactID references ContactDetails

**PromotionDiscounts**(promoCode, promoReason, discountPercentage, startDate, expiryDate)

Primary Key promoCode

**ErrorLog**(errorCode, errorReason, connID, logTime, logDate)

Primary Key (errorCode, connID, logTime, logDate)

**TransactionLog**(transactionID, transactionDescription, logTime, logDate, userID)

Primary Key (transactionID, logTime, logDate, userID)

Foreign Key userID references UserLogin

### 7.3 Data Dictionary

The figure below is eMart's data dictionary. It contains all tables and their attributes necessary for the eMart database and website to function. In the data dictionary, it states the maximum amount of characters allowed in the field and whether they are fixed or variable (the type of data is stated in the base tables, see section 9.2). It also states what type of key each attribute is (if any), a brief description of the attribute and whether it can be null or not. The data dictionary is a vital part of the design process as it clearly states all of the attributes ready for the implementation phase.

ENTITY	ATTRIBUTES	LENGTH	KEY	DESCRIPTION	NULL
<b>Department</b>					
<i>This is a table containing all the departments of eMart and their details</i>					
	departmentID	5 fixed characters	PK	The primary key of the department table	No
	departmentName	50 variable characters		The name of the department	No
	departmentManager	5 fixed characters	FK	The user ID of the manager of the department	Yes
	phoneNumber	Text		The phone number of the department	No
<b>UserLogin</b>					
<i>This is the employees information for them to log into the eMart system</i>					
	userID	5 fixed characters	PK	The employee's unique user login	No
	staffID	5 fixed characters	FK	The unique ID of the employee	No
	password	12 variable characters		The password of the user (will be encrypted)	No
	lastLogDate	10 fixed characters		The date at which the employee last logged into the system	Yes
<b>Staff</b>					
<i>This is the basic information of the employees</i>					
	staffID	5 fixed characters	PK	The unique ID of the employee	No
	firstName	30 variable characters		The employees first name	No
	surname	30 variable characters		The employee surname	No
	jobDescription	50 variable characters		The employees job role	No
<b>Stock</b>					
<i>This is the information of all the items which eMart stocks</i>					

	itemID	5 fixed characters	PK	The primary key of the stock table	No
	name	30 variable characters		This is the name of the item	No
	briefDescription	Text		This is a brief description of the item	Yes
	longDescription	Text		This is a long description of the item	Yes
	supplierID	5 fixed characters	FK	This is the ID of the supplier that eMart orders the item from	No
	subcategoryID	5 fixed characters	FK	The subcategory which the product is in for ecommerce purposes	No
	price	6 variable characters		The price at which eMart sells the product	No
	attributeID	5 fixed characters	FK	The attribute ID of the product	No
	thumbImage	Image		A small image of the product	Yes
	bigImage	Image		A large image of the product	Yes

### StockControl

*This is the table which monitors the stock levels*

	itemID	5 fixed characters	PK FK	The item ID of the product	No
	itemExpiryDate	10 fixed characters		The date at which the product does out of date	No
	idealQuantity	5 variable characters		The quantity which eMart wants the product to be at	No
	currentQuantity	5 variable characters		The quantity which the product is currently at	No
	status	10 variable characters		A flag of the level which the item is at	No
	departmentID	5 fixed characters	FK	This is the ID of the department which stores the item	No

### Category

*This is the parent category table for the subcategories of products in the website*

	categoryID	5 fixed characters	PK	The category ID of the category	No
	categoryName	30 variable characters		The name of the category	No

<b>Subcategory</b>					
<i>This is the child subcategory table for categories of products in the website</i>					
	subCategoryID	5 fixed characters	PK	The subcategory ID of the subcategory	No
	categoryID	5 fixed characters	FK	The category ID of the category	No
	subcategoryName	30 variable characters		The name of the subcategory	No
	description	Text		A brief description of the subcategory	Yes
	image	Image		An image of the subcategory	Yes
<b>Suppliers</b>					
<i>This is the basic information of the suppliers</i>					
	supplierID	5 fixed characters	PK	This is the primary key of the supplier table	No
	supplierName	30 variable characters		This is the name of the supplier	No
	emailAddress	80 variable characters		This is the email address of the contact	No
	contactID	5 fixed characters	FK	This is the ID of the supplier's contact details	No
<b>Supplier Deliveries</b>					
<i>This table contains the details of the deliveries from the suppliers</i>					
	deliveryID	5 fixed characters	PK	This is the primary key of the supplier delivery table	No
	supplierID	5 fixed characters	FK	This is the foreign key which links this table to the supplier table	No
	dateOfDelivery	10 fixed characters		This is the date at which the delivery will happen	No
	timeOfDelivery	5 fixed characters		This is the time at which the delivery will come	Yes
	itemID	5 fixed characters	FK	This is a foreign key which says which items are in the delivery	No

<b>Customer</b>					
<i>This is the basic information of the customer</i>					
	customerID	5 fixed characters	PK	This is the primary key of the customer table	No
	firstName	30 variable characters		This is the customers first name	No
	surname	30 variable characters		This is the customers surname	No
	registrationDate	10 fixed character		This is the date on which the customer registered with eMart	No
	emailAddress	80 variable characters	AK	This is the email address of the contact	No
	contactID	5 fixed characters	FK	This is the contact ID for the customer	No
	password	12 variable characters		The password which the customer uses to log in the website	No
<b>Contact Details</b>					
<i>This contains the contact details of any persons which eMart needs to contact</i>					
	contactID	5 fixed characters	PK	This is the primary key of the customer address table	No
	addressLineOne	50 variable characters		This is the first line of the contacts address	No
	town	50 variable characters		This is the town of the contact	Yes
	city	50 variable characters		This is the city of the contact	Yes
	county	50 variable characters		This is the county of the contact	Yes
	postcode	8 variable characters		This is the postcode of the contact	No
	telephoneNumber	Text		This is the telephone number of the contact	No
<b>Orders</b>					
<i>This is the basic information of all of the customer orders</i>					
	orderID	5 fixed characters	PK	This is the primary key of the order's table	No
	customerID	5 fixed characters	FK	This is a foreign key which links this table to the customer table	No
	paymentDate	10 fixed characters		This is the payment date of the order	No

	orderStatus	15 variable characters		This is the current status of the order	No
	promoCode	5 fixed characters	FK	This is the promo code of the order	Yes
	orderNotification	Boolean		This is whether the customer has had a notification of the status	No

### ShoppingCart

*This is the shopping cart data*

	cartID	10 fixed characters	PK	This is the primary key of the shopping cart table	No
	itemID	5 fixed characters	FK	This is the item ID(s) of what is in the shopping cart	No
	cartSessionID	15 variable characters		The session ID number for the cart	No
	cartQuantity	3 variable characters		This is the quantity of each item in the shopping cart	No
	cartDate	10 fixed characters		This is the date of the cart	No
	promoCode	5 fixed characters	FK	This is the promo code used in the cart	Yes

### AttributeCategory

*The parent groups name of any item attributes e.g. size*

	AttributeCategoryID	5 fixed characters	PK	The ID of the attribute group	No
	AttributeGroupName	30 variable characters		The name of the attribute group	No

### Attribute

*The table of any item attribute types e.g. small, medium, large*

	attributeID	5 fixed characters	PK	The ID of the attribute	No
	attributeCategoryID	5 fixed characters	FK	The ID of the attribute group	No
	attributeName	30 variable characters		The name of the attribute	No

<b>OrderItem</b>					
<i>This contains the information of which order each item is in</i>					
	orderID	5 fixed characters	PK	This is the order number	No
	itemID	5 fixed characters	PK	This is the item ID(s) of what is in the order	No
	orderQuantity	3 variable characters		This is the quantity of the item in the order	No
	promoCode	5 fixed characters	FK	This is the promo code of the item	Yes
<b>ShopConfig</b>					
<i>The shop information that can be used dynamically on the website</i>					
	name	30 variable characters		This is the name of the company	No
	contactID	5 fixed characters		The contact information of the company	No
	emailAddress	80 variable characters	PK	The email address which the customer uses to log in the website	No
	shippingCost	6 variable characters		The cost of delivery	No
<b>PromotionDiscounts</b>					
<i>This is the details of any promotions which eMart are running</i>					
	promoCode	5 fixed characters	PK	This is the promo code of the offer	No
	promoReason	30 variable characters		This is the reason for the promo	No
	discountPercentage	5 variable characters		This is the percentage of the promo	No
	startDate	10 fixed characters		This is when the promotion begins	No
	expiryDate	10 fixed characters		This is when the promotion ends	No
<b>Error Log</b>					
<i>This is a log of any errors that occur in the system</i>					

	errorCode	5 fixed characters	PK	This is the code of the error	No
	errorReason	30 variable characters		This is a description of the error	No
	connID	30 variable characters	PK	This is the address ID of the error origination	No
	logTime	8 fixed characters	PK	This is the time of the error	No
	logDate	10 fixed characters	PK	This is the date of the error	No

### TransactionLog

*This is a log of any access to the eMart system*

	transactionID	15 variable characters		This is the ID of the transaction	No
	transactionDescription	50 variable characters		This is a description of the transaction	Yes
	logTime	8 fixed characters		This is the time of the transaction	No
	logDate	10 fixed characters		This is the date of the transaction	No
	userID	5 fixed characters		This is the user ID of the employee who did the transaction	No

## 7.4 Transaction Matrix

The list of use cases for the system has been systematically evaluated and added to the transaction matrix below. The transaction matrix can help identify the tables that are most frequently accessed.

I = Insert, R = Read, U = Update, D = Delete.

Table Name	UC1				UC2				UC3				UC4				UC5			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
<b>Department</b>																				
<b>UserLogin</b>	X	X				X	X			X		X								
<b>Staff</b>		X				X				X										
<b>Stock</b>																				
<b>StockControl</b>																				
<b>Category</b>																				
<b>Subcategory</b>																				
<b>Suppliers</b>																				
<b>SupplierDeliveries</b>																				
<b>Customer</b>																				
<b>ContactDetails</b>																				
<b>Orders</b>																				
<b>ShoppingCart</b>																				
<b>AttributeCategory</b>																				
<b>Attribute</b>																				
<b>OrderItem</b>																				
<b>PromotionDiscount</b>																				
<b>ErrorLog</b>																X				
<b>TransactionLog</b>	X				X				X									X		

Table Name	UC6				UC7				UC8				UC9				UC16			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
<b>Department</b>																				
<b>UserLogin</b>																				
<b>Staff</b>	X	X				X				X	X			X		X				
<b>Stock</b>																				
<b>StockControl</b>																				
<b>Category</b>																				
<b>Subcategory</b>																				
<b>Suppliers</b>																				
<b>SupplierDeliveries</b>																				
<b>Customer</b>																	X	X		
<b>ContactDetails</b>																	X			
<b>Orders</b>																				
<b>ShoppingCart</b>																				
<b>AttributeCategory</b>																				
<b>Attribute</b>																				
<b>OrderItem</b>																				
<b>PromotionDiscount</b>																				
<b>ErrorLog</b>																				
<b>TransactionLog</b>	X				X				X				X		X		X		X	

Table Name	UC17				UC18				UC19				UC20				UC21			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department																				
UserLogin																				
Staff																				
Stock																	X			X
StockControl																	X			X
Category																	X			X
Subcategory																	X			X
Suppliers																				
SupplierDeliveries																				
Customer		X					X			X		X								
ContactDetails		X					X			X		X								
Orders																				
ShoppingCart																				
AttributeCategory																	X			X
Attribute																	X			X
OrderItem																				
PromotionDiscount																				
ErrorLog																				
TransactionLog	X					X				X										

Table Name	UC22				UC23				UC24				UC25				UC26			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department																				
UserLogin																				
Staff																				
Stock		X									X									X
StockControl		X									X							X	X	
Category																				
Subcategory																				
Suppliers																				
SupplierDeliveries																				
Customer																			X	
ContactDetails																			X	
Orders																			X	
ShoppingCart	X	X	X			X			X				X	X	X			X		
AttributeCategory		X																		
Attribute		X																		
OrderItem																			X	
PromotionDiscount		X																	X	
ErrorLog																				
TransactionLog																				

Table Name	UC27				UC28				UC29				UC30				UC31			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department																				
UserLogin											x				x				x	
Staff																				
Stock															x					
StockControl														x	x			x	x	
Category																				
Subcategory																				
Suppliers																				
SupplierDeliveries																				
Customer		x																		
ContactDetails	x																			
Orders	x				x				x	x				x	x		x		x	x
ShoppingCart																				
AttributeCategory																				
Attribute																				
OrderItem	x																x		x	x
PromotionDiscount																				
ErrorLog														x			x			
TransactionLog									x				x				x			

Table Name	UC32				UC33				UC34				UC35				UC36			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department																				
UserLogin	x				x				x				x				x			
Staff																				
Stock																				
StockControl																				
Category																				
Subcategory						x	x			x			x	x			x		x	
Suppliers						x	x			x			x	x			x		x	
SupplierDeliveries																				
Customer	x					x			x				x	x			x		x	x
ContactDetails	x				x				x				x	x			x		x	x
Orders	x	x																		
ShoppingCart																				
AttributeCategory																				
Attribute																				
OrderItem	x																			
PromotionDiscount																				
ErrorLog																				
TransactionLog	x				x				x				x				x			

Table Name	UC37				UC38				UC39				UC40				UC41			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department		x																		
UserLogin	x				x				x				x							
Staff																				
Stock	x	x				x				x	x		x		x					
StockControl	x	x											x		x					
Category	x	x																		
Subcategory	x	x																		
Suppliers		x																		
SupplierDeliveries																				
Customer																				
ContactDetails																				
Orders																				
ShoppingCart																				
AttributeCategory	x	x																		
Attribute	x	x																		
OrderItem																				
PromotionDiscount																				
ErrorLog																				
TransactionLog	x				x				x				x				x	x	x	x

Table Name	UC42				UC43				UC44				UC49				UC50			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Department													x	x			x			
UserLogin													x				x			
Staff																				
Stock																				
StockControl					x	x				x	x									
Category																				
Subcategory																				
Suppliers																				
SupplierDeliveries																				
Customer																				
ContactDetails																				
Orders																				
ShoppingCart																				
AttributeCategory																				
Attribute																				
OrderItem																				
PromotionDiscount																				
ErrorLog	x	x	x	x					x				x			x			x	
TransactionLog	x				x				x				x			x				

Table Name	UC51				UC52				UC53				UC54				UC55			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
<b>Department</b>																				
<b>UserLogin</b>		x				x				x								x	x	
<b>Staff</b>																				
<b>Stock</b>	x																			
<b>StockControl</b>	x	x											x	x						
<b>Category</b>																				
<b>Subcategory</b>																				
<b>Suppliers</b>		x				x														
<b>SupplierDeliveries</b>	x	x				x				x	x			x		x				
<b>Customer</b>																				
<b>ContactDetails</b>																				
<b>Orders</b>																				
<b>ShoppingCart</b>																				
<b>AttributeCategory</b>																				
<b>Attribute</b>																				
<b>OrderItem</b>																				
<b>PromotionDiscount</b>																				
<b>ErrorLog</b>																				
<b>TransactionLog</b>	x				x				x				x			x		x		

Table Name	UC56				UC57			
	I	R	U	D	I	R	U	D
<b>Department</b>								
<b>UserLogin</b>		x	x					
<b>Staff</b>								
<b>Stock</b>								
<b>StockControl</b>								
<b>Category</b>								
<b>Subcategory</b>								
<b>Suppliers</b>								
<b>SupplierDeliveries</b>								
<b>Customer</b>						x	x	
<b>ContactDetails</b>								
<b>Orders</b>								
<b>ShoppingCart</b>								
<b>AttributeCategory</b>								
<b>Attribute</b>								
<b>OrderItem</b>								
<b>PromotionDiscount</b>								
<b>ErrorLog</b>								
<b>TransactionLog</b>								

## 8. Physical Database Design

To ensure referential integrity all FOREIGN KEY's will be created with the ON UPDATE CASCADE and ON DELETE CASCADE. This is most worthwhile for all tables.

### 8.1 Base Tables

#### 8.1.1 Department Table

```
CREATE TABLE Department(
```

```
    departmentID INT(5) NOT NULL AUTO_INCREMENT,  
    departmentName VARCHAR(50) NOT NULL,  
    departmentManager INT(5) DEFAULT NULL,  
    telephoneNumber TEXT NOT NULL,  
    PRIMARY KEY (departmentID),  
    UNIQUE KEY departmentName(departmentName),  
    FOREIGN KEY departmentManager REFERENCES Staff(staffID))  
    AUTO_INCREMENT=00001;
```

#### 8.1.2 UserLogin Table

```
CREATE TABLE UserLogin(
```

```
    userID INT(5) NOT NULL AUTO_INCREMENT,  
    staffID INT(5) NOT NULL,  
    password VARCHAR(12) NOT NULL,  
    lastLogDate DATE DEFAULT NULL,  
    PRIMARY KEY(userID),  
    UNIQUE KEY staffID(staffID),  
    FOREIGN KEY staffID REFERENCES Staff(StaffID))  
    AUTO_INCREMENT=00001;
```

#### 8.1.3 Staff Table

```
CREATE TABLE Staff(
```

```
    staffID INT(5) NOT NULL AUTO_INCREMENT,  
    firstName VARCHAR(30) NOT NULL,
```

```

surname VARCHAR(30) NOT NULL,
jobDescription VARCHAR(50) NOT NULL,
PRIMARY KEY(staffID))
AUTO_INCREMENT=00001;

```

#### 8.1.4 Stock Table

```

CREATE TABLE Stock(
itemID INT(5) NOT NULL AUTO_INCREMENT,
name VARCHAR(30) NOT NULL,
briefDescription TEXT DEFAULT NULL,
longDescription TEXT DEFAULT NULL,
supplierID INT(5) NOT NULL,
subcategoryID INT(5) NOT NULL,
price FLOAT NOT NULL,
attributeID INT(5) NOT NULL,
thumbImage LONGBLOB,
bigImage LONGBLOB,
PRIMARY KEY(itemID),
UNIQUE KEY name(name),
FOREIGN KEY supplierID REFERENCES Supplier(supplierID),
FOREIGN KEY subcategoryID REFERENCES Subcategory(subcategoryID),
FOREIGN KEY attributeID REFERENCES Attribute(attributeID))
AUTO_INCREMENT=00001;

```

#### 8.1.5 StockControl Table

```

CREATE TABLE StockControl(
itemID INT(5) NOT NULL,
itemExpiryDate DATE NOT NULL,
idealQuantity INT(5) NOT NULL,
currentQuantity INT(5) NOT NULL,

```

```

status ENUM('Stocked','Medium','Reorder','OutOfStock') NOT NULL,
departmentID INT(5) NOT NULL,
PRIMARY KEY (itemID),
FOREIGN KEY itemID REFERENCES Stock(itemID));

```

### 8.1.6 Category Table

```

CREATE TABLE Category(
categoryID INT(5) NOT NULL AUTO_INCREMENT,
categoryName VARCHAR(30) NOT NULL,
PRIMARY KEY (categoryID),
UNIQUE KEY categoryName (categoryName)
) AUTO_INCREMENT=00001;

```

### 8.1.7 Subcategory Table

```

CREATE TABLE Subcategory(
subcategoryID INT(5) NOT NULL AUTO_INCREMENT,
categoryID INT(5) NOT NULL,
subcategoryName VARCHAR(30) NOT NULL,
description TEXT DEFAULT NULL,
image longblob DEFAULT NULL,
PRIMARY KEY(subcategoryID),
UNIQUE KEY name(name),
FOREIGN KEY categoryID REFERENCES Category(categoryID)
) AUTO_INCREMENT=00001;

```

### 8.1.8 Suppliers Table

```

CREATE TABLE Suppliers(
supplierID INT(5) NOT NULL AUTO_INCREMENT,
supplierName VARCHAR(30) NOT NULL,
emailAddress VARCHAR(80) NOT NULL,
contactID INT(5) NOT NULL,

```

```
PRIMARY KEY (supplierID),  
FOREIGN KEY contactID REFERENCES ContactDetails(ContactID)  
) AUTO_INCREMENT=00001;
```

### 8.1.9 SupplierDeliveries Table

```
CREATE TABLE SupplierDeliveries(  
    deliveryID INT(5) NOT NULL AUTO_INCREMENT,  
    supplierID INT(5) NOT NULL,  
    dateOfDelivery DATE NOT NULL,  
    timeOfDelivery TIME DEFAULT NULL,  
    itemID INT(5) NOT NULL,  
    PRIMARY KEY (deliveryID),  
    FOREIGN KEY supplierID REFERENCES Suppliers(supplierID),  
    FOREIGN KEY itemID REFERENCES Stock(itemID)  
) AUTO_INCREMENT=00001;
```

### 8.1.10 Customer Table

```
CREATE TABLE Customer(  
    customerID INT(5) NOT NULL AUTO_INCREMENT,  
    firstName VARCHAR(30) NOT NULL,  
    surname VARCHAR(30) NOT NULL,  
    registrationDate DATE NOT NULL,  
    emailAddress VARCHAR(80) NOT NULL,  
    contactID INT(5) NOT NULL,  
    password VARCHAR(12) NOT NULL,  
    PRIMARY KEY(customerID),  
    UNIQUE KEY emailAddress(emailAddress),  
    FOREIGN KEY contactID REFERENCES ContactDetails(contactID)  
) AUTO_INCREMENT=00001;
```

### 8.1.11 ContactDetails Table

```

CREATE TABLE ContactDetails(
    contactID INT(5) NOT NULL AUTO_INCREMENT,
    addressLineOne VARCHAR(50) NOT NULL,
    town VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    county VARCHAR(50) NOT NULL,
    postcode VARCHAR(8) NOT NULL,
    telephoneNumber TEXT NOT NULL,
    PRIMARY KEY(contactID)
) AUTO_INCREMENT=00001;

```

#### 8.1.12 Orders Table

```

CREATE TABLE Orders(
    orderID INT(5) NOT NULL AUTO_INCREMENT,
    customerID INT(5) NOT NULL,
    paymentDate DATE NOT NULL,
    orderStatus ENUM('Received','Processed','Despatched') NOT NULL,
    promoCode VARCHAR(5) DEFAULT NULL,
    orderNotification BOOLEAN NOT NULL,
    PRIMARY KEY(orderID),
    FOREIGN KEY customerID REFERENCES Customer(customerID),
    FOREIGN KEY promoCode REFERENCES PromoDiscounts(promoCode)
) AUTO_INCREMENT=00001;

```

#### 8.1.13 ShoppingCart Table

```

CREATE TABLE ShoppingCart(
    cartID INT(10) NOT NULL AUTO_INCREMENT,
    itemID INT(5) NOT NULL,
    cartSessionID VARCHAR(15) NOT NULL,
    cartQuantity INT(3) NOT NULL,

```

```

        cartDate DATE NOT NULL,
        promoCode VARCHAR(5) DEFAULT NULL,
        PRIMARY KEY(cartID),
        UNIQUE KEY cartSessionID(cartSessionID)
    ) AUTO_INCREMENT=0000000001;

```

#### 8.1.14 AttributeCategory Table

```

CREATE TABLE AttributeCategory(
    attributeCategoryID INT(5) NOT NULL AUTO_INCREMENT,
    attributeGroupName VARCHAR(30) NOT NULL,
    PRIMARY KEY(attributeCategoryID),
    UNIQUE KEY attributeGroupName(attributeGroupName)
) AUTO_INCREMENT=00001;

```

#### 8.1.15 Attribute Table

```

CREATE TABLE Attribute(
    attributeID INT(5) NOT NULL AUTO_INCREMENT,
    attributeCategoryID INT(5) NOT NULL,
    attributeName VARCHAR(30) NOT NULL,
    PRIMARY KEY (attributeID),
    FOREIGN KEY attributeCategoryID REFERENCES
    AttributeCategory(attributeCategoryID)
) AUTO_INCREMENT=00001;

```

#### 8.1.16 OrderItem Table

```

CREATE TABLE OrderItem(
    orderID INT(5) NOT NULL,
    itemID INT(5) NOT NULL,
    orderQuantity INT(3) NOT NULL,
    promoCode VARCHAR(5) DEFAULT NULL,
    PRIMARY KEY (orderID,itemID),

```

```
    FOREIGN KEY orderID REFERENCES Orders(orderID),
    FOREIGN KEY itemID REFERENCES Stock(itemID));
```

#### 8.1.17 ShopConfig Table

```
CREATE TABLE ShopConfig(
    name VARCHAR(30) NOT NULL,
    contactID INT(5) NOT NULL,
    emailAddress VARCHAR(80) NOT NULL,
    shippingCost FLOAT NOT NULL,
    PRIMARY KEY(emailAddress),
    FOREIGN KEY contactID REFERENCES ContactDetails(contactID));
```

#### 8.1.18 PromotionDiscounts Table

```
CREATE TABLE PromotionDiscounts(
    promoCode VARCHAR(5) NOT NULL,
    promoReason VARCHAR(30) NOT NULL,
    discountPercentage FLOAT NOT NULL,
    PRIMARY KEY(promoCode),
    CHECK(0<discountPercentage<51));
```

#### 8.1.19 ErrorLog Table

```
CREATE TABLE ErrorLog(
    errorCode VARCHAR(5) NOT NULL,
    errorReason VARCHAR(30) NOT NULL,
    connID VARCHAR(30) NOT NULL,
    logTime TIME NOT NULL,
    logDate DATE NOT NULL,
    PRIMARY KEY(errorCode,connID,logTime,logDate));
```

#### 8.1.20 TransactionLog Table

```
CREATE TABLE TransactionLog(
```

```

transactionID VARCHAR(15) NOT NULL,
transactionDescription VARCHAR(50) NOT NULL,
logTime TIME NOT NULL,
logDate DATE NOT NULL,
userID INT(5) NOT NULL,
PRIMARY KEY(transactionID,logTime,logDate(userID)),
FOREIGN KEY userID REFERENCES UserLogin(userID));

```

## 8.2 Stored Procedures

A number of stored procedures will be implemented into the database design to aid in maximising performance for queries that are run frequently. MySQL will generate an execution plan for the queries in the stored procedure when it is first executed, and reuse the same plan on subsequent executions of the procedure. Using stored procedures permits easier implementation of the three-tier architecture (the database stored procedures forming the data tier). Using stored procedures also allows the security to be better controlled since, MySQL permits setting different security permissions for each stored procedure whereas SQL queries created make-shift in the PHP code are more vulnerable to SQL injection attacks.

## 8.3 Security Mechanisms

Initially, the system will be set up with a single global user. This user will be responsible for creating new user accounts and will grant the relevant privileges to each. For the purpose of eMart this user will be the System Administrator.

The System Administrator will then be able to create accounts for individual members of staff; they will also then be able to grant permissions. As mentioned previously, each tier of the hierarchy (see requirements stage 4.4) will have different levels of access to the eMart system.

eMart will have protocols in order to keep information on the company, its employees and its customers safe and secure. The most important issue to address will be passwords. Employees and customers alike will have passwords to access the eMart system or eMart's ecommerce solution. All passwords will be encrypted so that no one except for the account holder will know it. If a member of staff forgets their password, they will have to go to the system administrator to have it changed.

In order for the company to ensure the security of its system, employees will have to change their password every 90 days. This will be a strict company policy. Also, passwords will have to be between 8 and 12 characters long and will contain numbers, letters and symbols. They will be a mixture of upper and lower casing. There will be guidelines in place to help staff create a secure password, i.e. it would be preferable that staff do not use any part of their name or address as their password for obvious reasons.

For customers who forget their password, on the login page there will be a 'forgot password' option. When clicking this option, they will be directed to a page where they will have to enter their email

address and a password reset link will be sent to their email account. They will be advised to check they spam folder in case the account does not recognise the eMart email service as being safe.

The information which eMart stores on its staff and customers would not be considered as sensitive. The company keeps only basic information on its employees (first name and surname). For its customers, eMart holds information on how to contact them but no payment details (as this is going to be handled by third party ePayment corporation, PayPal). To the customer, their contact details will be thought of as sensitive therefore, it is the company's duty to ensure the customer feels safe giving eMart their personal information. eMart will not give any customer information to any third parties unless the customer has opted in to doing so. This is important not only to the customer but also to the company because if customers don't feel comfortable giving eMart their information, then they will use competitor sites.

## 8.4 Performance

There are many basic database design options available to improve performance. In the basic implementation one can strategically utilize indexes to speed up queries on a heavily read-oriented database. In the case of the eMart, a first step could be to split the product database from the shopping cart and order database. That would make one of the servers heavily read-oriented and the other much more write-oriented. And one will have successfully split the transactions across two servers. If even that isn't enough, then one needs to consider strategies for splitting the customers into different parts based on postcode, having multiple cached reading servers, and reviewing the code architecture. Since eMart is a relatively small business in terms of its current customer base, performance shouldn't be too much of a problem with the server capable of handling the forecast traffic.

## 8.5 Network Protocol

Since the ecommerce front-end will be implemented as a web site, and thus on a single server, the principal communication standard will be the standard HTTP. However, the processing of the customer's financial information must be secure. For this, the system will implement the PayPal Transaction Protocol. The payment protocol begins after the customer decides to pay for an item via their shopping cart. The process starts by the customer selecting to "pay with PayPal". The website will subsequently redirect the customer to PayPal where a secure connection using SSL (Secure Socket Layer) is used. The payment is processed and the customer is redirected back to the eMart web site. Evidently all the sensitive information of the transaction is handled via PayPal.

PayPal creates a secure connection to its server through SSL. SSL is a protocol used to manage the security of the message in a transmission of data across a public network. SSL integrates security services in the session layer for applications as encrypted and integrity over Hyper Text Transfer Protocol (HTTP).

Additionally, TCP/IP is, of course, the standard networking protocol used across the Internet that allows computer to communicate with each other.

## 8.6 Pseudo code for Key Algorithms

### 8.6.1 Password Hashing

Password hashing is a method of encrypting a password before it's stored as to limit the damage should the database happen to fall into the wrong hands. The eMart eCommerce solution will implement password hashing. Once a password has been hashed it is incredibly difficult to transform

that password back into its original form. This raises the question as to how you're going to be able to tell whether a password entered by a user is correct or not. This question is answered by generating a hash of the user-supplied password and comparing the fingerprint of that hash with the hash stored in the user tuple. The SH1 hashing algorithm will be used to hash passwords. SH1 returns a 160-bit hash (40 hexadecimal characters).

```
/* Store new user details */

START
    Retrieve user entered password from the form
        apply SH1 hashing algorithm to the password
        store result in a variable x
    Insert new user data with variable x as the password
END

/* Check user details */

START
    Retrieve user entered credentials from the form
        apply SH1 hashing algorithm to the password
        store result in a variable y
    Retrieve credentials stored in table for the user credentials with password y
        IF the user is not found
            then return NO_SUCH_USER
        ELSE
            grant user access
    END
```

The SH1 hashing algorithm does however have a weakness. This weakness in the hashing strategy can be overcome by making a small addition to the hashing algorithm. Before generating the hash we create a random string of characters of a predetermined length, and append this string to the start of the plain text password. Provided the string (called a "salt") is of sufficient length - and of course sufficiently random - the resulting hash will almost certainly be different each time we execute the function. Of course we must also store the salt we've used in the database along with our hash but this is generally no more of an issue than extending the width of the field by a few characters. This addition is not really necessary for the purpose of eMart, however the algorithm is illustrated should it become desirable to implement at a later stage.

```

/* Append salt string to password */

START

Define a constant 'LENGTH' of size 9

GenerateHash(password, salt_variable=null)

IF salt_variable = null

    then set salt_variable to

        substr(sh1(uniqid(rand(), true)), 0, LENGTH)

ELSE

    set salt_variable to

        substr(salt_variable, 0, LENGTH)

RETURN salt_variable concatenated with sha1(salt_variable concatenated with the original
password)

END

```

### 8.6.2 Calculating cart totals

For each item in the cart a unit price exists and a quantity. When the cart is processed the correct total cost has to be calculated. The following algorithm will be used.

```

/* Calculate line price */

START

CalcLine(itemID and quantity)

    retrieve item price from the database using (SELECT price FROM Stock WHERE itemID =
the ID of the line item)

    Multiply the price by the quantity

RETURN the totalLinePrice

END

```

```

/* Calculate subtotal price */

START

Define variable subtotal=0

CalcSubtotal(Hashtable h <itemID, quantity>)

Define variable x=0

```

```

FOREACH item in h
    x = CalcLine(h[].itemID, h[].quantity)
    subtotal += x
RETURN subtotal
END

/* Calculating total order cost */

START
Define variable total=subtotal
CalcTotal()

    Retrieve shippingCost from the database (using SELECT shippingCost FROM ShopConfig
    WHERE emailAddress = admin@emart.com)

    total += shippingCost

RETURN total
END

```

## 8.7 Global Control Flow

### 8.7.1 Execution Order

eMart database system is event-driven since it is almost completely based on user interaction. The events of the system are determined by user behaviour, what options they select etc.

### 8.7.2 Time Dependency

The eMart system implements a timer in the system. This timer is the 60 mins inactive log-in time allowed. If a customer logs into his/her account and remains inactive for 60 mins then the system will log him/her out.

### 8.7.3 Concurrency

For the application interface of the system, if two users want to access the same information (stock data) at the same time, we would need multiple threads (e.g. a supervisor and assistant, both log on to update a order status at the same time, on two different accounts). To prevent this problem, we will need to use locks around critical sections where data is being written to and read, to prevent the dirty read problem.

## **9. Interface Design**

### **9.1 Web Site Design**

#### **9.1.1 Web Site Overview**

The store front of this software solution will be a website that is accessed by customers, enabling them to purchase goods over the Internet. Data from eMart's product catalogue is maintained in the relational database, and pages requiring this data are generated dynamically.

Included is an illustration of the architecture and navigation scheme of the website, in addition to the identification of key content and the graphical design. Accessibility and usability factors have been addressed with close regard to the target audience derived from extensive research carried out. The primary objective of the design is to develop a design that will connect with the target demographic on both an emotional and value level.

When designing the interface for the ecommerce website there are several factors that the design should meet in order to be deemed successful. The following depicts the key aims of the design.

- Accumulate elevated traffic
- Increase product awareness
- Generate more sales
- Offer an efficient e-commerce solution
- Accessibility

#### **9.1.2 Target Audience**

##### *Attitude/Psycho-persona*

The attitude of the target audience has been carefully scrutinized and a number of important factors have been determined. It is thought that the site visitors will be moderately difficult to please when it comes to the overall functionality of the site. This is because the majority of site visitors will want their experience to be as efficient as possible. The site will need to provide 'correct' information as quickly as possible.

We are living in a throw away world. People are pressed for time and urgently need to find a few more hours in a day, therefore an increasing number of grocery shoppers are turning to the internet to place orders conveniently. Consumers are busier and more stressed, and have little patience for tasks that take time. For this reason, eMart must provide an efficient solution, which will be less time consuming than doing a weekly shop in the local grocery store.

The target audience is predominantly large families with children. The profile most common is a young to middle aged married, divorced or single mother, looking to make their daily errands easier in any way possible. For example, shopping online a consumer can simply click on their favourites and automatically add things to the shopping cart.

Another growing group of online shoppers are the elderly and physically disabled, as these people might find it difficult or impossible to make it to the grocery stores. Although this group is growing, older consumers still tend to like to place orders via telephone and targeting that audience is a costly, trivial venture.

It can be assumed that the target audience are not particularly aesthetically conscious individuals thus as long as the site layout and navigation is most excellent then the general aesthetics of the site

doesn't require any scrupulous attention. They will expect to be able to swiftly navigate throughout the site as efficiently as possible.

### *Demography*

The target demographic of the site is considered to be females of age group 25-40. However, it is likely that a wide range of age groups from both sexes will visit the site. This will be accommodated in the design.

### *Accessible content*

The text on the website will be adjustable to meet the needs of the audience. Those with bad eyesight and those older visitors, who may be dissuaded by small compact text, can simply click the button which increases and decreases the size of the font.

The language of the site will meet the standards of the eMart's origin, England. The sole reason for this is that all visitors will have to be a resident of the UK to process an order.

The website will have multiple methods of navigation to meet the needs of the visitors with the more interactive approach, and those which like to swiftly and efficiently navigate via shortcuts.

#### **9.1.3 Site Style and Appearance**

The design must conform to the following few design points:

- Warm, friendly, and comfortable – like shopping at the local store.
- Organic style/colour palette with recognition that many visitors will be female.
- Inviting, simple, and clean this signifies that dynamic features such as animations are inappropriate.
- Effortless accessibility - fast, well organized navigation.
- Must also appear professional, trustworthy, and stable.

### *Site Features*

The design will incorporate the following features:

- 15 to 20 pages as appropriate.
- To be built using original page templates.
- Three quarter of pages (75%) devoted to each displaying product inventory.
- On-line registration to accommodate complete customer and order data.
- E-commerce gateway to allow charging order fees. This will be PayPal.
- Routine maintenance possible by eMart staff using Administrative back-end.
- Site hosting to be determined by eMart prior to going live.

### *Site References*

It is helpful when developing a new site to look at what competitors and others have done.

Although the site we are developing will be stylistically and operationally unique, the following sites have some of the features and style we believe should be incorporated into the eMart site.

- [www.tesco.co.uk](http://www.tesco.co.uk) – favourite products feature.
- [www.ocado.com](http://www.ocado.com) – organic colour scheme, fresh simple style.
- [www.waitrose.co.uk](http://www.waitrose.co.uk) – neat navigation.
- [www.naturalgrocery.co.uk](http://www.naturalgrocery.co.uk) – warm, inviting design.

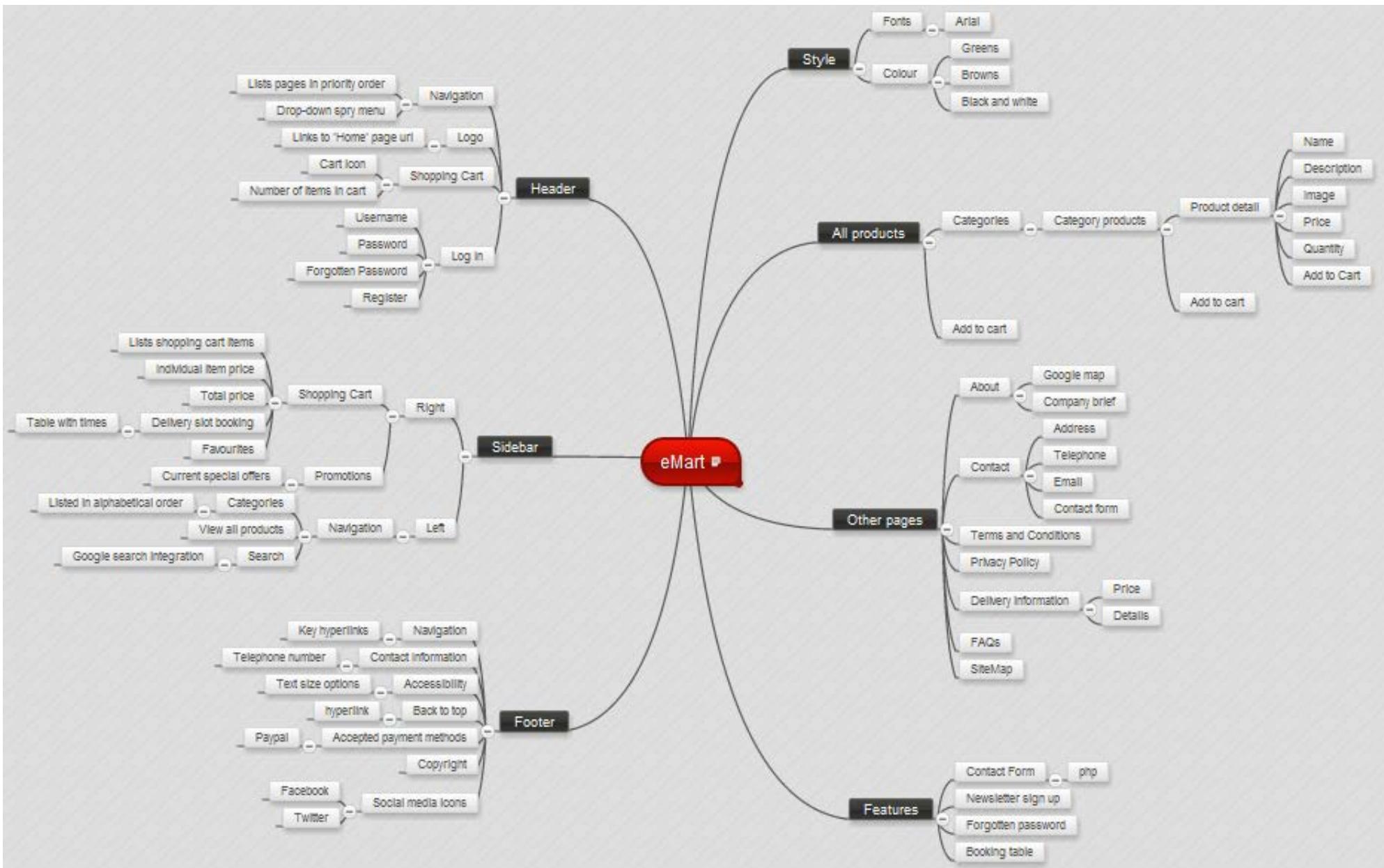
### 7.2.1 Promotion

The promotion of the web site on the internet, both in terms of getting it on the various search engines and also building links with other web sites, are vitally important to the continued success of the site. Additionally, the site will be supported by an off line strategy of marketing and advertising. The various methods of promotion will include the following:

- Affiliate marketing
- Search engine optimisation and submission
- Email marketing – newsletters
- Banner advertisements on high traffic volume web sites
- Social media marketing
- Brochures
- Advertising campaign
- Complimentary gifts, discount vouchers
- Building links with suppliers/partners

### 9.1.4 Idea Map

Idea mapping is very useful when designing a website. It will aide in getting the brainstorming ideas written down in a means that eases comprehension when it comes to sketching the site out. Mapping the ideas also helps in organising site content. The idea map for eMart is included below.



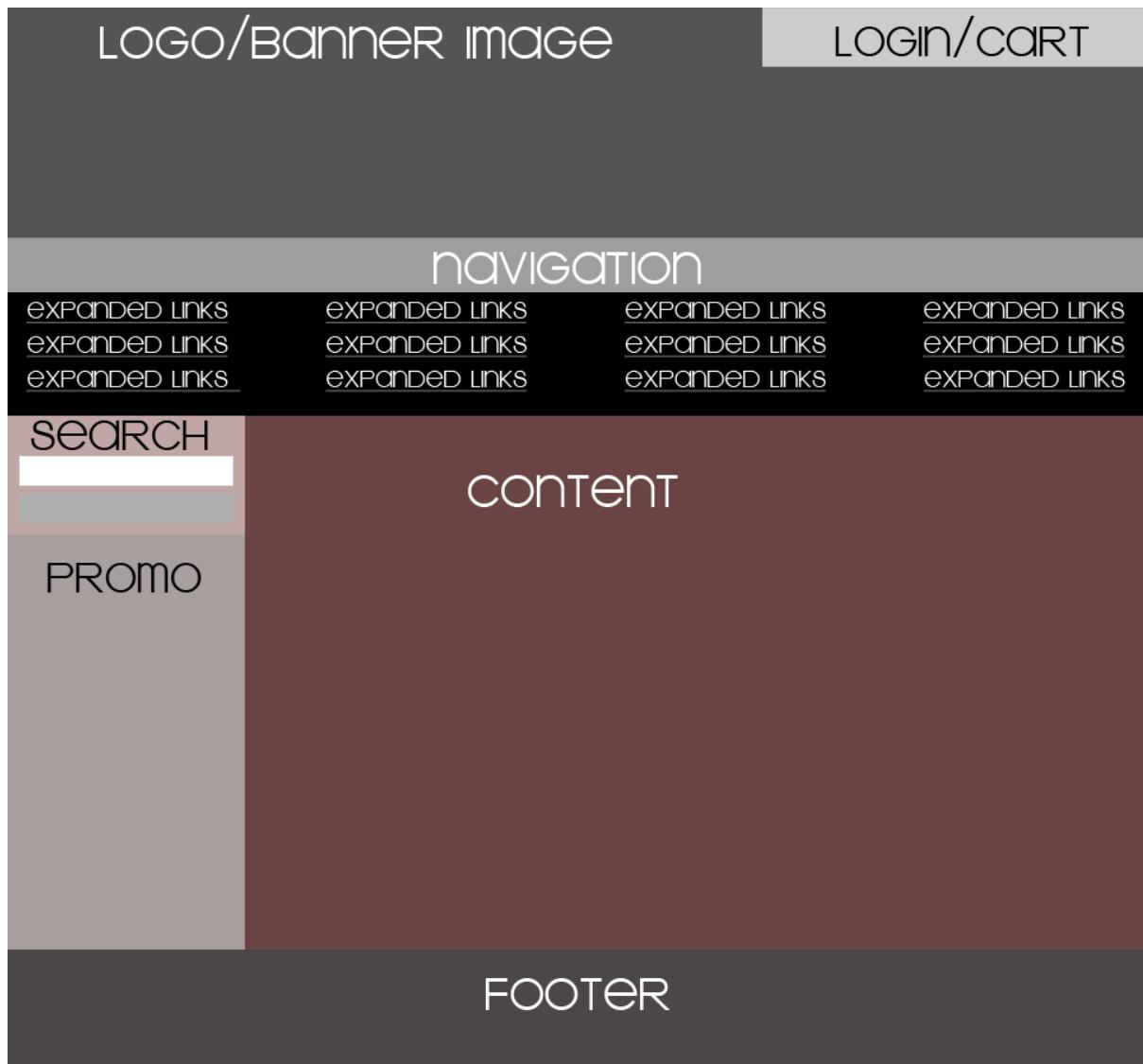
### 9.1.5 Layout Sketches

Idea sketching is a key process in web design. It is a method of sketching any web design layouts that could be ideal for the current project, experimenting with colours and style. Once the idea sketching is complete, the best solution for the purpose of eMart can be decided upon and developed into concept sketches. Sketches for the key pages of the site can be seen below.

#### *Idea 1:*

Below is the first in a series of sketches, illustrating a possible web page layout. As you can see from the image, each navigation link would have a drop-down box showing many different subpages (links).

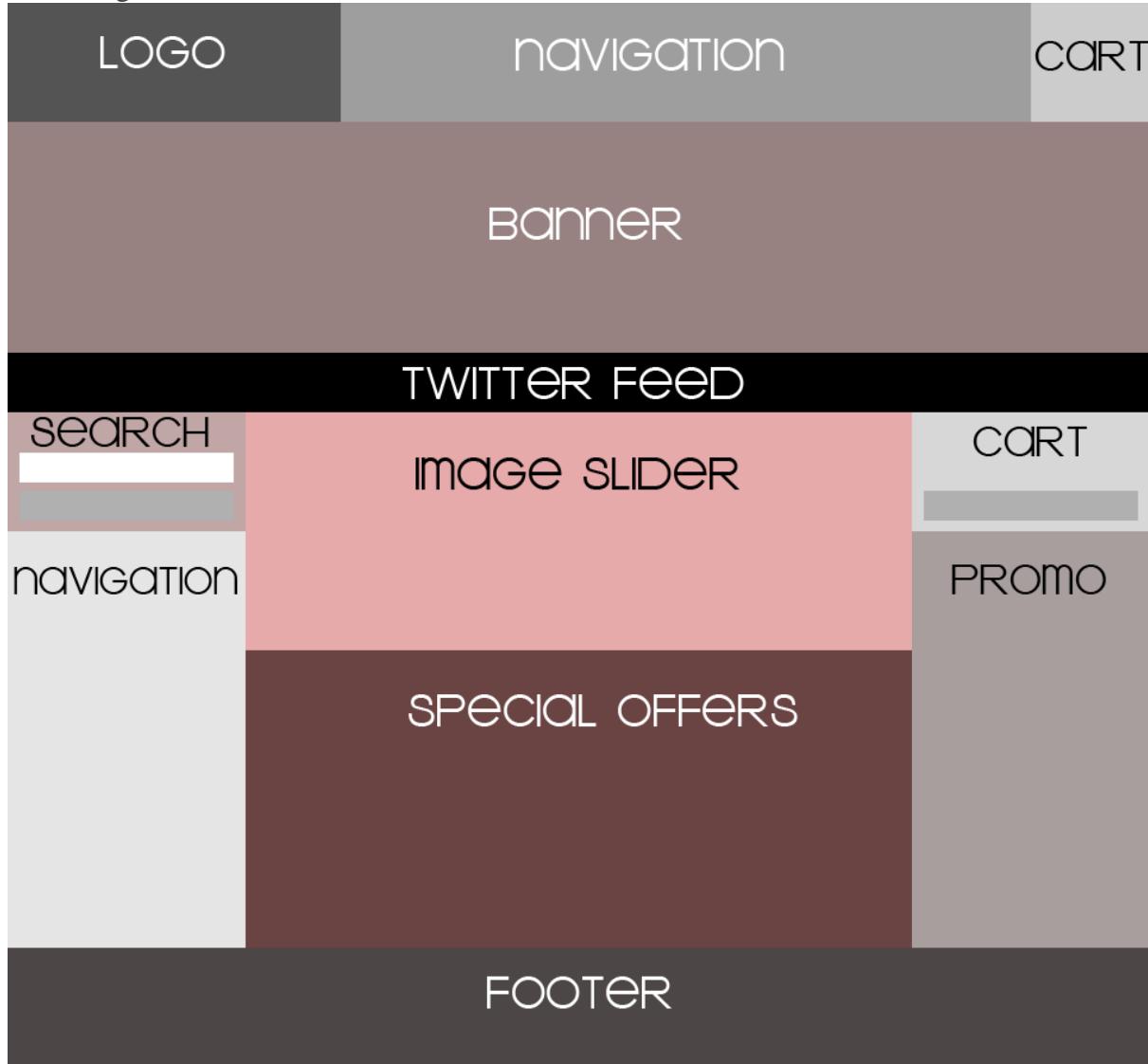
Home Page



*Idea 2:*

The layout below, implements a more comprehensive layout. It contains both high-level navigation at the top for hierarchical navigation and a sidebar navigation that will provide efficient movement throughout particular categories of products. It also provides an area to include a live twitter feed.

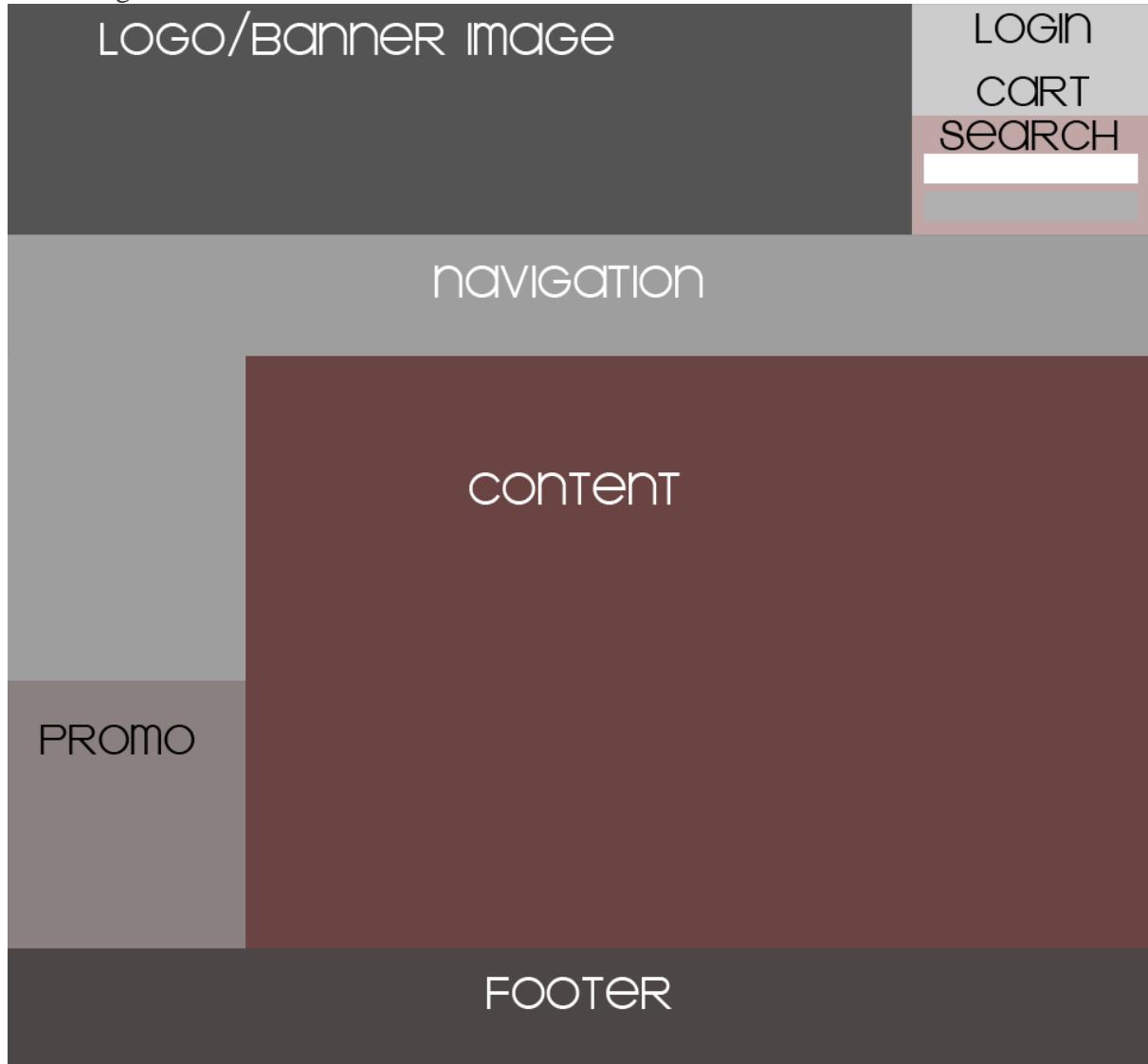
Home Page



*Idea 3:*

The idea sketch below includes a tab navigation layout. The vertical section shows links for any subpages of the tab selected. For example, on the home page the vertical section will contain key site pages, but on the various category tabs this section will show subcategories for the particular category.

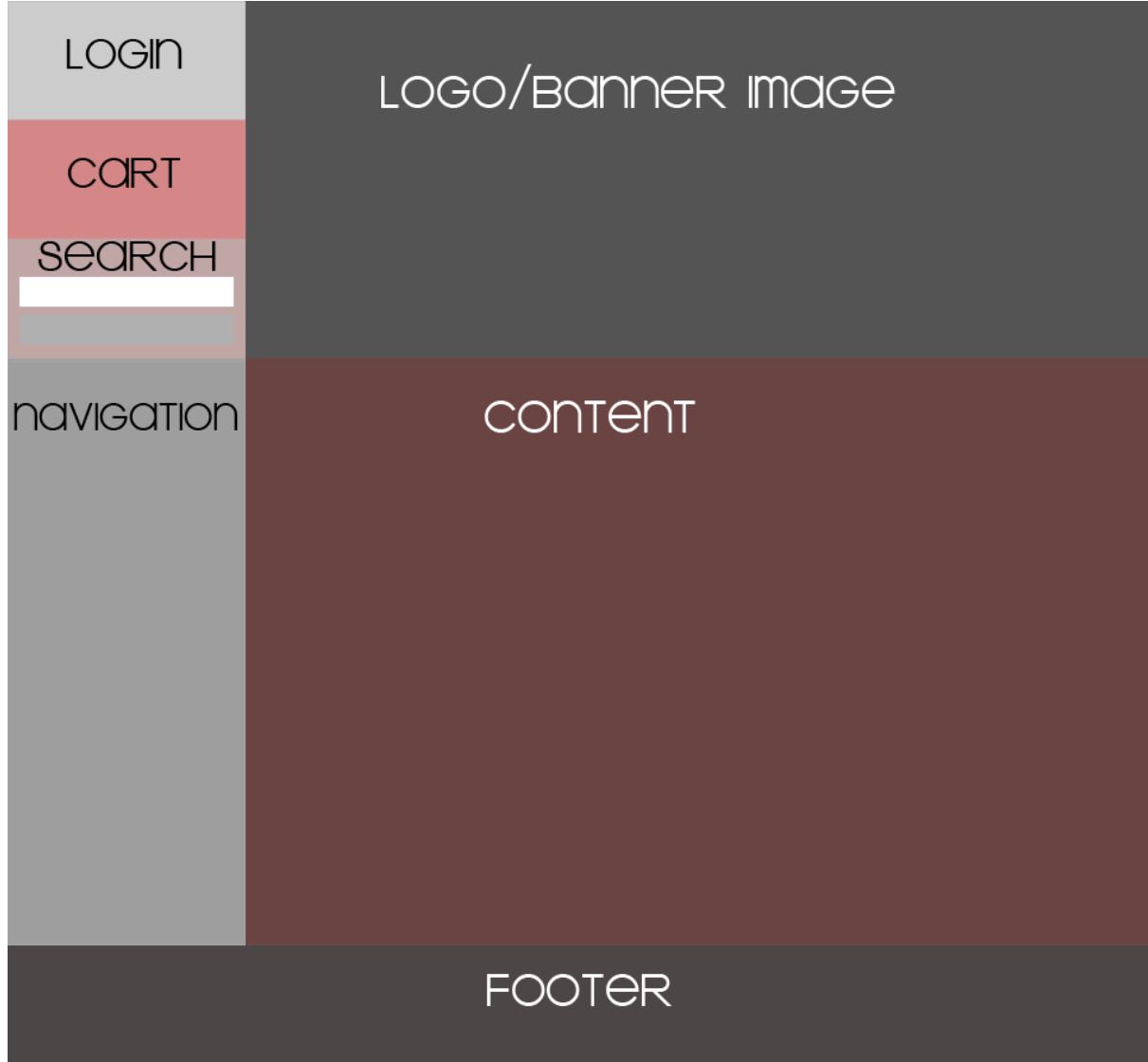
Home Page



#### *Idea 4:*

The sketch below depicts a very clean, simple design with all functional content (i.e. navigation etc) on the left and all content kept to the right. This is a particularly basic design, that could prove ideal for the purpose of eMart.

Home Page

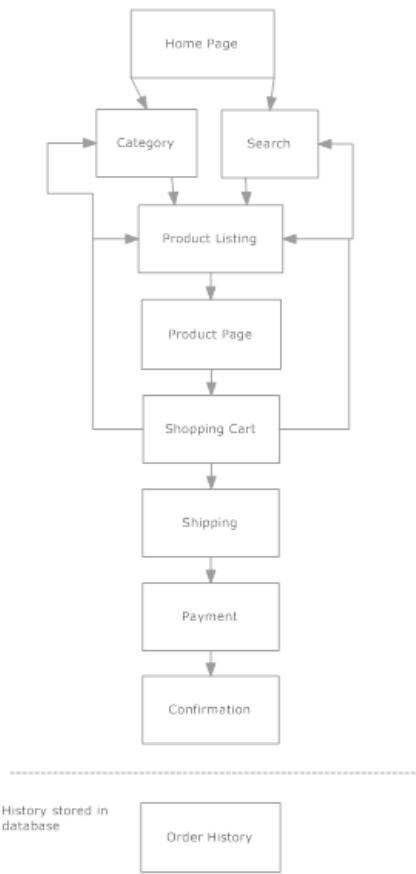


#### 9.1.6 Navigation Path Diagrams

When customers access the storefront, they will typically follow one of several steps for beginning their shopping process. Ideally they will look at category and product data through browsing the site. Or they may search for something specific via the integrated Google search feature. The navigation designed into the system provides a way to jump between any states that are appropriate. However, it should not be possible for someone to go to the checkout page with an empty shopping cart. Finally, it is going to be critical that state be maintained throughout the entire process. It will be necessary to track the current customer ID in order to maintain their basket.

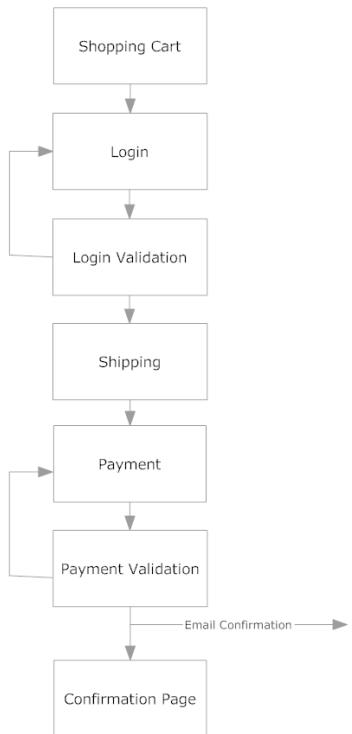
## *Shopping Process*

Flow chart of shopping process:



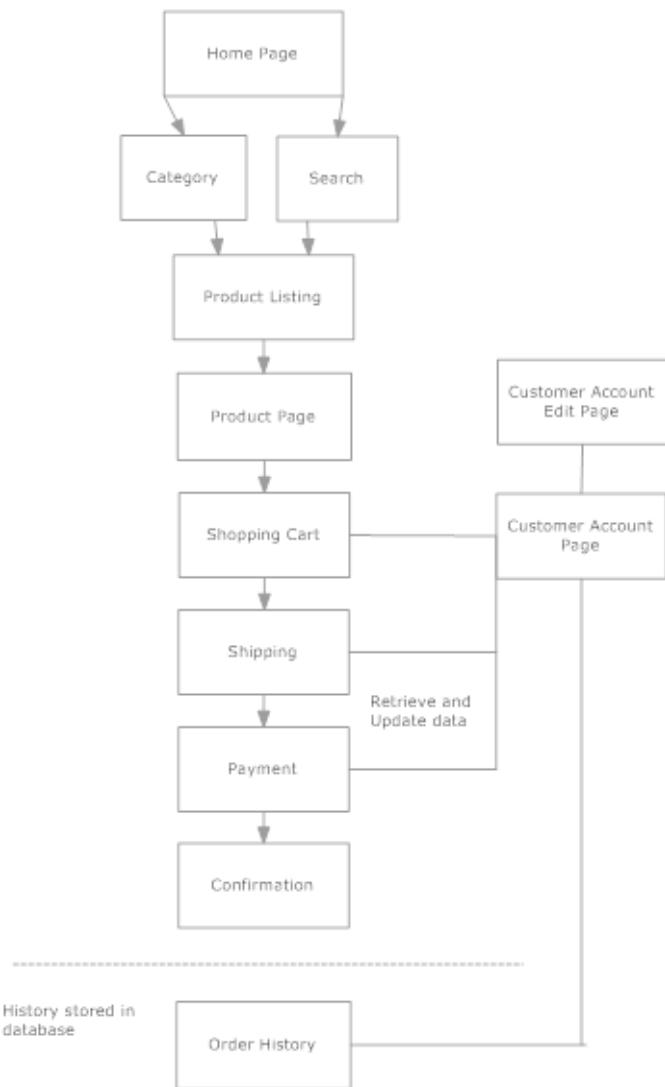
## *Checkout Process*

Flow chart of checkout process



## *Account Data Interaction*

Flow chart of customer account interaction



### 9.1.7 Personas

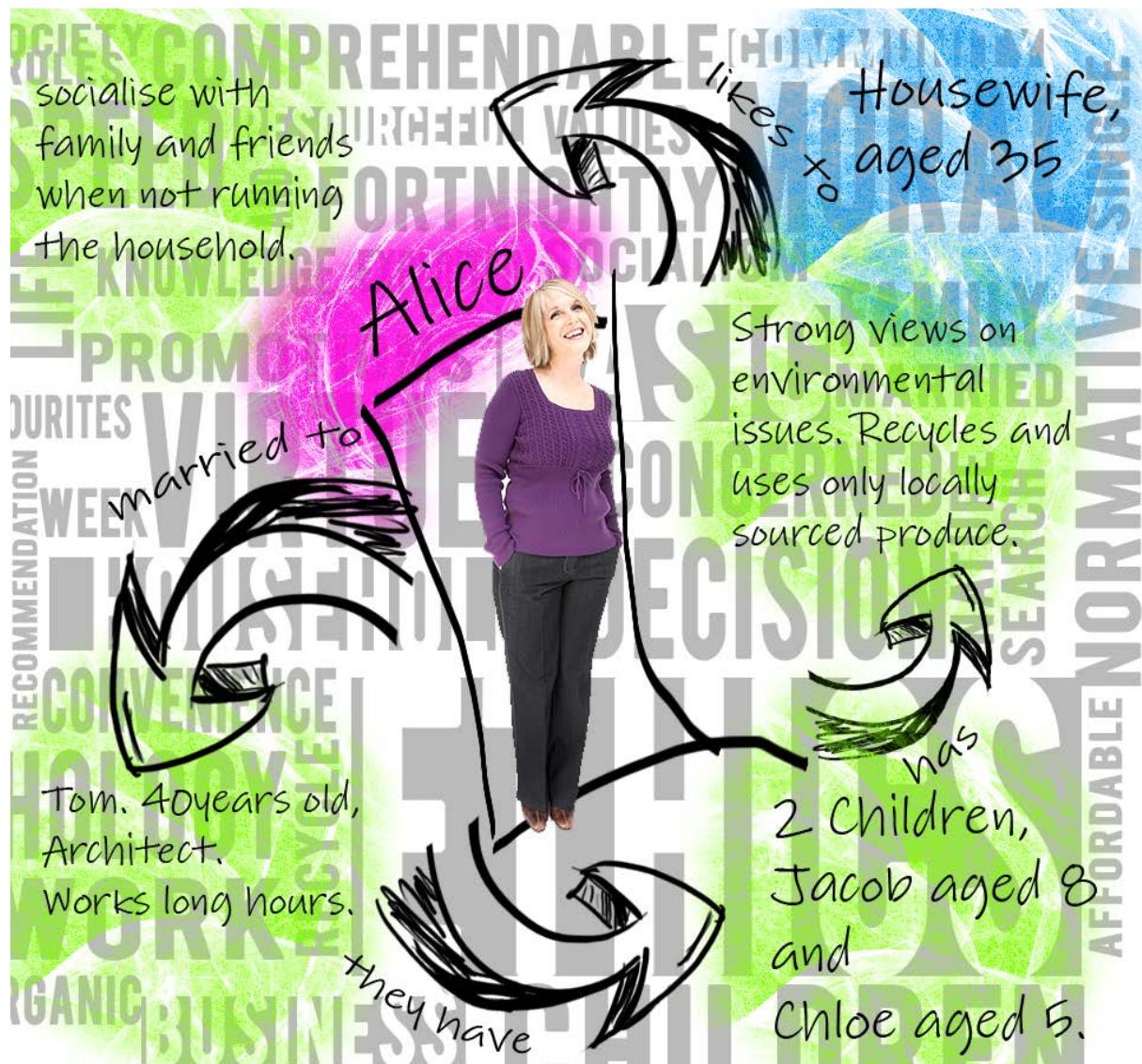
Personas are a key factor in designing an effective website. “Creating personas of website visitors is a powerful technique for helping increase the usability and customer centricity of a web site as part of a user-centred design (UCD) process.” (Chaffey, 2011)

The reason behind creating a persona when developing a design for a web site is to ensure that the website appeals to all types of users. “If the user is satisfied with his or her conversation with your site, you will reap the rewards”. (Steve Mulder, 2007)

Personas are there to help set goals for the design of a website as the web site will then have the objective to meet the goal that the persona has set. Below are a few personas created for the eMart ecommerce site.

### Persona 1:

Alice is a 35 year old housewife. Alice likes to socialise with family and friends when she gets free time away from her daily household errands. Alice has always had aspirations to run her own vegan restaurant. Since settling down with her family she has had to put those aspirations aside. Once Alice's children reach an appropriate age she hopes to return to work. Alice likes a website to be resourceful, giving her detailed information about products she is thinking about purchasing. Alice likes a website to be basic, making navigation straightforward so that she doesn't have to waste time searching for products. Alice has a basic system which has the power to run websites with rich media content, so the website loading speed is not an issue.



### Persona 2:

Mark is a 39 year old, single working parent. Mark has a physically demanding job at a demolition company. Mark has a child aged 15 that resides with him Monday to Thursday. Mark likes to unwind at weekends by going to a local pub with friends to watch the football. Mark also likes to use the computer. The computer system that Mark uses is a laptop which is low in speed, so he likes the websites that he browses to be simple, with fast loading times.



### 9.1.8 Accessibility

Accessibility options will be implemented into the web interface to accommodate those users with visual impairments. These options will ensure the smooth navigation through the site for all. Accessibility is incorporated to assist not only users with visual impairments but also those with cognitive and neurological shortcomings.

#### *Visual impairment*

Issues include:-

- Semi-Blindness
  - Some vision
    - Vision in one eye.
    - Some vision in both.
- Colour Blindness
  - Red-Green
    - Most common.
    - Difficulty to see the difference between Red and Green hues.
  - Blue-Yellow
    - Rarer.
    - Difficult to see the difference between Blue and Yellow.
  - Monochromacy
    - Unable to see any colour
- Fully Blind
  - Cannot see at all out of either eye (Potts, 2007)

A design solution that will be included to meet the potential eyesight deficiencies of users is a feature that allows the user to increase or decrease the size of the font on the screen. This will be done via simple increase and decrease buttons.

#### *Learning Disabilities*

When designing a website the range of developmental and learning disabilities that a potential user might have is vast. They can affect any part of an individual's ability to process the information being displayed. In order to maximise site users site content will be kept concise and easily readable.

#### *Epilepsy*

Epilepsy is a major factor in determining the design of a website. There is no way of knowing whether a site user suffers from this condition. Any flashing content and sudden changes in colour on the webpage can affect the user. To avoid this problem no flashing text/images will be used on the site.

### 9.1.9 Concept Sketches

A concept template has been sketched of the website design. Each page of the site will follow the same structure. Images of some of the key pages of the site are included below.

## Home Page



Category Page



## Product List Page



The image shows a screenshot of the eMART website's product list page. The header features a green circular logo with the word "eMART" in white and green. To the right of the logo are links for "HOME", "CATEGORIES" (which is highlighted in yellow), "ABOUT", and "CONTACT". A shopping basket icon with the number "0" indicates no items in the basket. Below the header is a dark banner with a variety of fresh vegetables like carrots, onions, and tomatoes.

Below the banner, there is a "TWEET" button followed by five "TWEET" links. To the right is a search bar with the placeholder "SEARCH". On the left, there is a sidebar with a list of categories:

- VIEW ALL PRODUCTS
- FRESH FOOD
- BAKERY
- FOOD CUPBOARD
- FROZEN FOOD
- DRINKS
- BABY
- HEALTH & BEAUTY
- HOUSEHOLD PRODUCTS

The main content area displays a grid of products. The first row contains two columns with the text "<category name>". Below this grid are navigation arrows labeled "PREV", "... 2 ...", and "NEXT".

At the bottom of the page, there is a copyright notice: "© COPYRIGHT 2012 eMART." followed by links to "HOME", "CATEGORIES", "ABOUT", "CONTACT", "T's & C's", "PRIVACY POLICY", "FAQ", and "SITE MAP". It also mentions "WEBSITE CREATED BY KISACASA". A "PayPal" logo is present on the right side.

## Product Detail Page



## Shopping Cart Page

The screenshot shows the eMART shopping cart page. At the top, there's a green circular logo with the word "eMART" in white. To the right of the logo are links for "HOME", "CATEGORIES", "ABOUT", and "CONTACT". Above these links is a small green shopping bag icon with the number "0" next to it, followed by "VIEW YOUR BASKET", "MY ACCOUNT", and "HELP". Below the main navigation is a banner featuring various fresh vegetables like lettuce, carrots, and onions.

Below the banner, there's a social media sharing section with "TWEET" buttons and a "SEARCH" bar. The main content area is titled "Your Shopping Cart" and contains a table for "Items". The table has columns for "name", "brief description", and "price". There are four rows in the table, each with a "remove" link at the bottom. A "total:" label is followed by a "update" button and a "Checkout" button. On the left side, there's a sidebar with a list of categories: "VIEW ALL PRODUCTS", "FRESH FOOD", "BAKERY", "FOOD CUPBOARD", "FROZEN FOOD", "DRINKS", "BABY", "HEALTH & BEAUTY", and "HOUSEHOLD PRODUCTS".

At the bottom of the page, there's a copyright notice: "© COPYRIGHT 2012 eMART. HOME CATEGORIES ABOUT CONTACT T's & C's PRIVACY POLICY FAQ SITE MAP WEBSITE CREATED BY KISACASA". There's also a "PayPal" logo.

## Customer Account Page

The screenshot shows the Customer Account Page of the eMART website. The header features the eMART logo with a green circle and a cartoon character. The navigation menu includes links to HOME, CATEGORIES, ABOUT, and CONTACT. A shopping basket icon indicates 0 items. The main content area is titled "Customer Account" and contains sections for "Details" (with a "edit" link) and "Order History" (with two "View details" links). On the left, a sidebar lists product categories: VIEW ALL PRODUCTS, FRESH FOOD, BAKERY, FOOD CUPBOARD, FROZEN FOOD, DRINKS, BABY, HEALTH & BEAUTY, and HOUSEHOLD PRODUCTS. Social media sharing buttons for Twitter are located above the sidebar. The footer includes copyright information (© COPYRIGHT 2012 eMART), links to HOME, CATEGORIES, ABOUT, CONTACT, T's & C's, PRIVACY POLICY, FAQ, and SITE MAP, and a note that the website was created by KISACASA. It also features a PayPal logo.

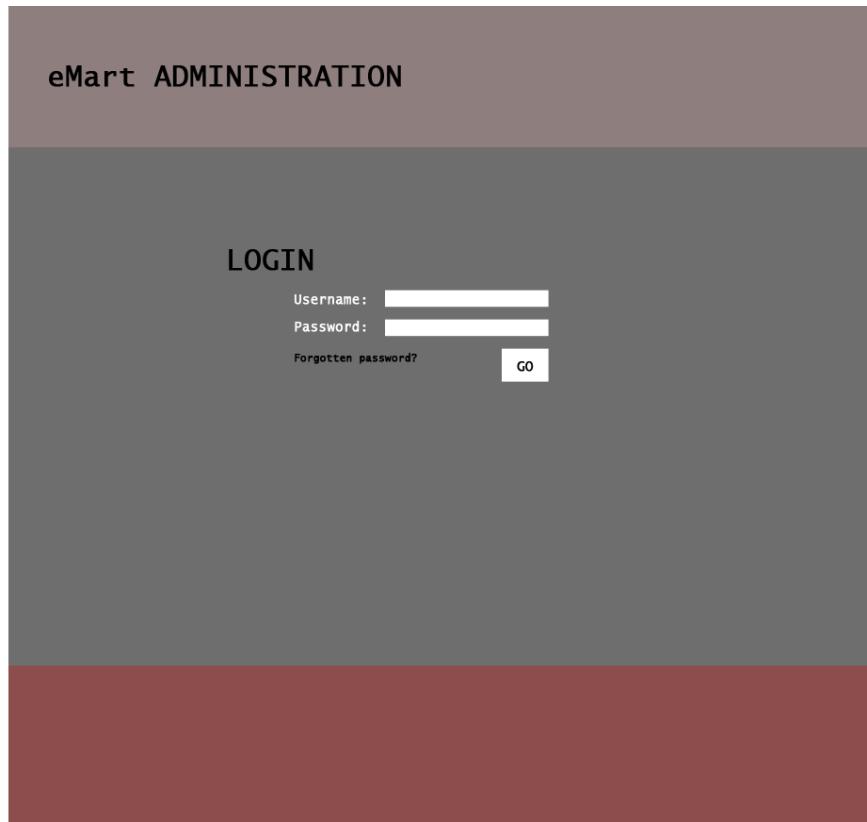
## 9.2 Administrative Interface

### 9.2.1 Layout Sketches

Layout sketches for the key pages of the admin area of the eMart eCommerce solution can be seen below.

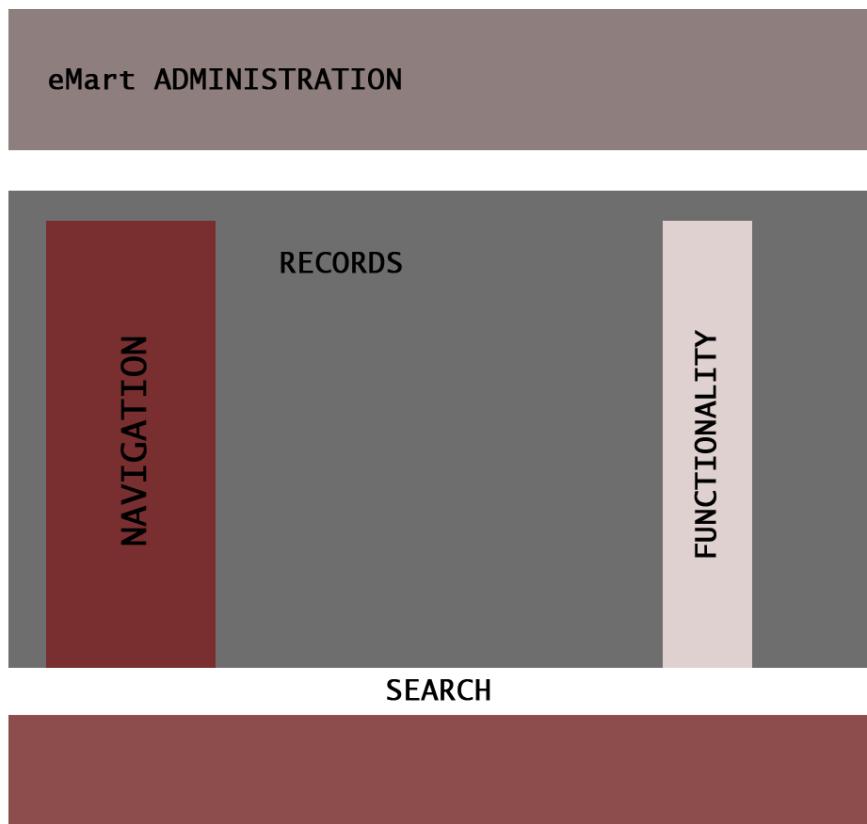
#### *Login Page:*

Here is a layout sketch illustrating the layout for the login page of the administrative back-end.



*Internal Page:*

Here the layout for internal pages of the database management backend is sketched.



### 9.2.2 Concept Sketches

A concept template has been sketched of the administrative back-end design. Each internal page of the site will follow the same structure. Images of some of the key pages of the site are included below. It is worth noting that the navigation buttons viewable on all internal pages become inactive if a logged in user doesn't have the credentials to access that particular part of the system. Also, <\$userID> in the diagrams signifies the current logged in user.

*Admin Login Page*



### Site Administrator Login

Username:   
Password:

[Forgotten password?](#) [GO](#)

© COPYRIGHT 2012 eMART.  
CREATED BY KISACASA

*Main Page*



<\$userid> [LOGOUT](#)

### Main Page

Choose a menu from the left navigation to get started

- [Home](#)
- [Departments](#)
- [Suppliers](#)
- [Stock](#)
- [Orders](#)
- [Customers](#)
- [Reports](#)
- [Configuration](#)

© COPYRIGHT 2012 eMART.  
CREATED BY KISACASA

Departments Page



KISACASA eCommerce Solution      Support

## Support

## STORE ADMINISTRATION

<\$userid>

**LOGOUT**

Departments Page

Home	eMart Departments				
Departments	ID	DEPARTMENT NAME	MANAGER NAME	PHONE NUMBER	SELECT
Suppliers					
Stock					
Orders					
Customers					
Reports					
Configuration					

© COPYRIGHT 2012 eMART.  
CREATED BY KISACASA





## Appendix C

# Realisation

---

## Realisation

Following the design phase of the project, the implementation phase immediately commenced. During this phase the source code for the system was produced and integrated into the overall system. Additionally, user documentation was produced. The source code and documentation was packaged for deployment along with detailed installation instructions. This section depicts the transition from design documentation to a live system and provides an overview of the system produced.

### System Overview

This section provides an overview, along with various screen shots, of the final working system deployed.

#### Source Code

System source code available upon request.

#### MySQL Implementation Script

```
--  
-- Database: `emart`  
--  
  
CREATE DATABASE IF NOT EXISTS emart;  
  
--  

```

```

CREATE TABLE IF NOT EXISTS `audit` (
  `audit_id` int(11) NOT NULL auto_increment,
  `order_id` int(11) NOT NULL,
  `created_on` datetime NOT NULL,
  `message` text NOT NULL,
  `code` int(11) NOT NULL,
  PRIMARY KEY (`audit_id`),
  KEY `idx_audit_order_id` (`order_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 

-- Table structure for table `categories`
-- 

CREATE TABLE IF NOT EXISTS `categories` (
  `categoryID` int(11) NOT NULL auto_increment,
  `departmentID` int(11) NOT NULL,
  `categoryName` varchar(25) NOT NULL,
  `description` varchar(500) NOT NULL,
  PRIMARY KEY (`categoryID`),
  KEY `departmentID` (`departmentID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 

-- Table structure for table `contactdetails`
-- 

CREATE TABLE IF NOT EXISTS `contactdetails` (
  `contactID` int(11) NOT NULL auto_increment,
  `addressLineOne` varchar(50) NOT NULL,
  `town` varchar(50) NOT NULL,
  `city` varchar(50) NOT NULL,
  `county` varchar(50) NOT NULL,
  `postcode` varchar(8) NOT NULL,
  `telephoneNumber` text NOT NULL,
  PRIMARY KEY (`contactID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 

-- Table structure for table `customer`
-- 

CREATE TABLE IF NOT EXISTS `customer` (
  `customerID` int(11) NOT NULL auto_increment,
  `firstname` varchar(30) NOT NULL,
  `surname` varchar(30) NOT NULL,
  `emailAddress` varchar(80) NOT NULL,
  `password` varchar(50) NOT NULL,
  `unencPassword` varchar(50) NOT NULL,

```

```

`creditCard` text NOT NULL,
`contactID` int(11) NOT NULL,
PRIMARY KEY (`customerID`),
UNIQUE KEY `emailAddress` (`emailAddress`),
KEY `contactID` (`contactID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- Table structure for table `department`
-- 

CREATE TABLE IF NOT EXISTS `department` (
`departmentID` int(11) NOT NULL auto_increment,
`name` varchar(100) NOT NULL,
`manager` int(11) default NULL,
`telephoneNumber` text NOT NULL,
`description` varchar(500) NOT NULL,
PRIMARY KEY (`departmentID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- Table structure for table `orderitems`
-- 

CREATE TABLE IF NOT EXISTS `orderitems` (
`productID` int(11) NOT NULL auto_increment,
`orderID` int(11) NOT NULL,
`itemID` int(11) NOT NULL,
`attributes` varchar(1000) NOT NULL,
`itemName` varchar(30) NOT NULL,
`orderQuantity` int(11) NOT NULL,
`price` decimal(10,2) NOT NULL,
PRIMARY KEY (`productID`),
KEY `orderID` (`orderID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- Table structure for table `orders`
-- 

CREATE TABLE IF NOT EXISTS `orders` (
`orderID` int(11) NOT NULL auto_increment,
`customerID` int(11) NOT NULL,
`totalAmount` decimal(10,2) NOT NULL default '0.00',
`createdOn` datetime NOT NULL,
`shippedOn` datetime default NULL,
`orderStatus` enum('Received','Checking Funds','Notifying Stock
Check','Awaiting Stock Confirmation','Awaiting Payment','Notifying
Warehouse Despatch','Awaiting Despatch Confirmation','Notifying
Customer','Complete','Cancelled') default 'Received',

```

```

`comments` varchar(255) default NULL,
`authCode` varchar(50) default NULL,
`reference` varchar(50) default NULL,
`shippingID` int(11) default NULL,
PRIMARY KEY (`orderID`),
KEY `customerID`(`customerID`),
KEY `shippingID`(`shippingID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 

-- 
-- Table structure for table `productattribute`
-- 

CREATE TABLE IF NOT EXISTS `productattribute` (
`itemID` int(11) NOT NULL,
`attributeValueID` int(11) NOT NULL,
PRIMARY KEY (`itemID`, `attributeValueID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 

-- 
-- Table structure for table `productcategories`
-- 

CREATE TABLE IF NOT EXISTS `productcategories` (
`itemID` int(11) NOT NULL,
`categoryID` int(11) NOT NULL,
PRIMARY KEY (`itemID`, `categoryID`),
KEY `categoryID`(`categoryID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 

-- 
-- Table structure for table `shipping`
-- 

CREATE TABLE IF NOT EXISTS `shipping` (
`shippingID` int(11) NOT NULL auto_increment,
`shippingType` varchar(100) NOT NULL,
`shippingCost` decimal(10,2) NOT NULL,
PRIMARY KEY (`shippingID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 

-- 
-- Table structure for table `shoppingcart`
-- 

CREATE TABLE IF NOT EXISTS `shoppingcart` (

```

```

`productID` int(11) NOT NULL auto_increment,
`cartID` char(32) NOT NULL,
`itemID` int(11) NOT NULL,
`attributes` varchar(1000) NOT NULL,
`quantity` int(11) NOT NULL,
`inventoryQty` int(11) NOT NULL,
`buyNow` tinyint(1) NOT NULL default '1',
`addedOn` datetime NOT NULL,
PRIMARY KEY (`productID`),
KEY `cartID` (`cartID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- 
-- Table structure for table `stock`
-- 

CREATE TABLE IF NOT EXISTS `stock` (
`itemID` int(11) NOT NULL auto_increment,
`itemName` varchar(30) NOT NULL,
`briefDescription` varchar(500) NOT NULL,
`longDescription` varchar(1000) default NULL,
`supplierID` int(11) default NULL,
`price` decimal(10,2) NOT NULL default '0.00',
`discountedPrice` decimal(10,2) NOT NULL default '0.00',
`thumbImage` varchar(150) default NULL,
`bigImage` varchar(150) default NULL,
`display` smallint(6) NOT NULL default '0',
PRIMARY KEY (`itemID`),
KEY `supplierID` (`supplierID`),
FULLTEXT KEY `itemName` (`itemName`, `longDescription`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- 
-- Table structure for table `stockcontrol`
-- 

CREATE TABLE IF NOT EXISTS `stockcontrol` (
`itemID` int(11) NOT NULL,
`idealQuantity` int(11) NOT NULL default '50',
`currentQuantity` int(11) NOT NULL,
`status` enum('Stocked','Medium','Reorder','OutOfStock') NOT NULL
default 'OutOfStock',
PRIMARY KEY (`itemID`),
KEY `currentQuantity` (`currentQuantity`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- -----
-- 
-- 
-- Table structure for table `supplierdeliveries`
-- 

```

```

CREATE TABLE IF NOT EXISTS `supplierdeliveries` (
  `deliveryID` int(11) NOT NULL auto_increment,
  `supplierID` int(11) NOT NULL,
  `dateOfDelivery` date NOT NULL,
  `timeOfDelivery` time default NULL,
  PRIMARY KEY (`deliveryID`),
  KEY `supplierID` (`supplierID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

-- -----
-- 
-- Table structure for table `suppliers`
-- 

CREATE TABLE IF NOT EXISTS `suppliers` (
  `supplierID` int(11) NOT NULL auto_increment,
  `supplierName` varchar(30) NOT NULL,
  `emailAddress` varchar(80) NOT NULL,
  `contactID` int(11) NOT NULL,
  PRIMARY KEY (`supplierID`),
  KEY `contactID` (`contactID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

DELIMITER $$

-- 
-- Procedures
-- 

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_add_attribute`(IN
inName VARCHAR(100))
BEGIN
INSERT INTO attribute (name) VALUES (inName);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_add_attribute_value`(
IN inAttributeId INT, IN inValue VARCHAR(100))
BEGIN
INSERT INTO attributevalue (attributeID, value)
VALUES (inAttributeId, inValue);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_add_category`(
IN inDepartmentId INT,
IN inName VARCHAR(25), IN inDescription VARCHAR(500))
BEGIN
INSERT INTO categories (departmentID, categoryName, description)
VALUES (inDepartmentId, inName, inDescription);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_add_department`(
IN inName VARCHAR(100), IN inManager INT, IN inNumber TEXT, IN
inDescription VARCHAR(500))
BEGIN
INSERT INTO department (name, manager, telephoneNumber, description)
VALUES (inName, inManager, inNumber, inDescription);
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_add_product_to_category`(IN inCategoryId INT,
IN inName VARCHAR(30), IN inBriefDescription VARCHAR(500), IN
inDescription VARCHAR(1000),
IN inPrice DECIMAL(10, 2), IN inCurrentQuantity INT, IN
inIdealQuantity INT)
BEGIN
DECLARE productLastInsertId INT;
INSERT INTO stock (itemName, briefDescription, longDescription,
price)
VALUES (inName, inBriefDescription, inLongDescription, inPrice);
SELECT LAST_INSERT_ID() INTO productLastInsertId;
INSERT INTO productcategories (itemID, categoryID)
VALUES (productLastInsertId, inCategoryId);
INSERT INTO stockcontrol(itemID, currentQuantity, idealQuantity)
VALUES (productLastInsertId, inCurrentQuantity, inIdealQuantity);
IF inCurrentQty <= 0 THEN
UPDATE stockcontrol
SET status = 'OutOfStock'
WHERE itemID = productLastInsertId;
ELSEIF inCurrentQty >= (inIdealQty/2) AND inCurrentQty < inIdealQty
THEN
UPDATE stockcontrol
SET status = 'Medium'
WHERE itemID = productLastInsertId;
ELSEIF inCurrentQty > inIdealQty THEN
UPDATE stockcontrol
SET status = 'Stocked'
WHERE itemID = productLastInsertId;
ELSEIF inCurrentQty <= (inIdealQty/2) THEN
UPDATE stockcontrol
SET status = 'Reorder'
WHERE itemID = productLastInsertId;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_assign_attribute_value_to_product`(
IN inProductId INT, IN inAttributeValueId INT)
BEGIN
INSERT INTO productattribute (itemID, attributeValueID)
VALUES (inProductId, inAttributeValueId);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_assign_product_to_category`(
IN inProductId INT, IN inCategoryId INT)
BEGIN
INSERT INTO productcategories (itemID, categoryID)
VALUES (inProductId, inCategoryId);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_count_products_in_category` (IN inCategoryID INT)
BEGIN

```

```

SELECT COUNT(*) AS categories_count
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
WHERE pc.categoryID = inCategoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_count_products_on_catalog`() BEGIN
SELECT COUNT(*) AS products_on_catalog_count
FROM stock
WHERE display = 1 OR display = 3;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_count_products_on_department`(IN inDepartmentID INT)
BEGIN
SELECT DISTINCT COUNT(*) AS products_on_department_count
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
INNER JOIN categories c
ON pc.categoryID = c.categoryID
WHERE (p.display = 2 OR p.display = 3)
AND c.departmentID = inDepartmentID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_delete_attribute`(IN inAttributeId INT)
BEGIN
DECLARE attributeRowsCount INT;
SELECT count(*)
FROM attributevalue
WHERE attributeID = inAttributeId
INTO attributeRowsCount;
IF attributeRowsCount = 0 THEN
DELETE FROM attribute WHERE attributeID = inAttributeId;
SELECT 1;
ELSE
SELECT -1;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_delete_attribute_value`(IN inAttributeValueId INT)
BEGIN
DECLARE productAttributeRowsCount INT;
SELECT count(*)
FROM stock p
INNER JOIN productattribute pa
ON p.itemID = pa.itemID
WHERE pa.attributeValueID = inAttributeValueId
INTO productAttributeRowsCount;
IF productAttributeRowsCount = 0 THEN

```

```

DELETE FROM attributevalue WHERE attributeValueID =
inAttributeValueId;
SELECT 1;
ELSE
SELECT -1;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_delete_category`(IN
inCategoryId INT)
BEGIN
DECLARE productCategoryRowsCount INT;
SELECT count(*)
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
WHERE pc.categoryID = inCategoryId
INTO productCategoryRowsCount;
IF productCategoryRowsCount = 0 THEN
DELETE FROM categories WHERE categoryID = inCategoryId;
SELECT 1;
ELSE
SELECT -1;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_delete_department`(IN
inDepartmentId INT)
BEGIN
DECLARE categoryRowsCount INT;
SELECT count(*)
FROM categories
WHERE departmentID = inDepartmentId
INTO categoryRowsCount;
IF categoryRowsCount = 0 THEN
DELETE FROM department WHERE departmentID = inDepartmentId;
SELECT 1;
ELSE
SELECT -1;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_delete_product`(IN
inProductId INT)
BEGIN
DELETE FROM productattribute WHERE itemID = inProductId;
DELETE FROM productcategories WHERE itemID = inProductId;
DELETE FROM shoppingcart WHERE itemID = inProductId;
DELETE FROM stock WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_attributes`() BEGIN
SELECT attributeID, name FROM attribute ORDER BY attributeID;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_attributes_not_assigned_to_product`(
IN inProductId INT)
BEGIN
SELECT a.name AS attribute_name,
av.attributeValueID, av.value AS attribute_value
FROM attributevalue av
INNER JOIN attribute a
ON av.attributeID = a.attributeID
WHERE av.attributeValueID NOT IN
(SELECT attributeValueID
FROM productattribute
WHERE itemID = inProductId)
ORDER BY attribute_name, av.attributeValueID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_attribute_details`(IN inAttributeId INT)
BEGIN
SELECT attributeID, name
FROM attribute
WHERE attributeID = inAttributeId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_attribute_values`(IN inAttributeId INT)
BEGIN
SELECT attributeValueID, value
FROM attributevalue
WHERE attributeID = inAttributeId
ORDER BY attributeID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_categories`() 
BEGIN
SELECT categoryID, categoryName, description
FROM categories
ORDER BY categoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_categories_for_product`(IN inProductId INT)
BEGIN
SELECT c.categoryID, c.departmentID, c.categoryName
FROM categories c
JOIN productcategories pc
ON c.categoryID = pc.categoryID
WHERE pc.itemID = inProductId
ORDER BY categoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_categories_list`(IN inDepartmentID INT)
BEGIN
SELECT categoryID, categoryName
FROM categories

```

```

WHERE departmentID = inDepartmentID
ORDER BY categoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_category_details`(IN inCategoryID INT)
BEGIN
SELECT categoryName, description
FROM categories
WHERE categoryID = inCategoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_category_name`(IN inCategoryID INT)
BEGIN
SELECT categoryName FROM categories WHERE categoryID = inCategoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_category_products`(IN inCategoryId INT)
BEGIN
SELECT p.itemID, p.itemName, p.briefDescription, p.longDescription,
p.price,
p.discountedPrice
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
WHERE pc.categoryID = inCategoryId
ORDER BY p.itemID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_departments`()()
BEGIN
SELECT departmentID, name, manager, telephoneNumber, description
FROM department
ORDER BY departmentID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_departments_list`()()
BEGIN
SELECT departmentID, name FROM department ORDER BY departmentID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_department_categories`(IN inDepartmentId INT)
BEGIN
SELECT categoryID, categoryName, description
FROM categories
WHERE departmentID = inDepartmentId
ORDER BY categoryID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_department_details`(IN inDepartmentID INT)
BEGIN

```

```

SELECT name, description
FROM department
WHERE departmentID = inDepartmentID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_department_name`(IN inDepartmentID INT)
BEGIN
SELECT name FROM department WHERE departmentID = inDepartmentID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_products_in_category`(
IN inCategoryID INT, IN inShortProductDescriptionLength INT,
IN inProductsPerPage INT, IN inStartItem INT)
BEGIN
PREPARE statement FROM
"SELECT p.itemID, p.itemName,
IF(LENGTH(p.briefDescription) <= ?, ,
p.briefDescription,
CONCAT(LEFT(p.briefDescription, ?),
'...')) AS briefDescription,
p.price, p.discountedPrice, p.thumbImage
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
WHERE pc.categoryID = ?
ORDER BY p.display DESC
LIMIT ?, ?";
SET @p1 = inShortProductDescriptionLength;
SET @p2 = inShortProductDescriptionLength;
SET @p3 = inCategoryID;
SET @p4 = inStartItem;
SET @p5 = inProductsPerPage;
EXECUTE statement USING @p1, @p2, @p3, @p4, @p5;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_products_on_catalog`(
IN inShortProductDescriptionLength INT,
IN inProductsPerPage INT, IN inStartItem INT)
BEGIN
PREPARE statement FROM
"SELECT itemID, itemName,
IF(LENGTH(briefDescription) <= ?, ,
briefDescription,
CONCAT(LEFT(briefDescription, ?),
'...')) AS briefDescription,
price, discountedPrice, thumbImage
FROM stock
WHERE display = 1 OR display = 3
ORDER BY display DESC
LIMIT ?, ?";
SET @p1 = inShortProductDescriptionLength;
SET @p2 = inShortProductDescriptionLength;
SET @p3 = inStartItem;

```

```

SET @p4 = inProductsPerPage;
EXECUTE statement USING @p1, @p2, @p3, @p4;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_products_on_department`(
IN inDepartmentID INT, IN inShortProductDescriptionLength INT,
IN inProductsPerPage INT, IN inStartItem INT)
BEGIN
PREPARE statement FROM
"SELECT DISTINCT p.itemID, p.itemName,
IF(LENGTH(p.briefDescription) <= ?, 
p.briefDescription,
CONCAT(LEFT(p.briefDescription, ?),
'...')) AS briefDescription,
p.price, p.discountedPrice, p.thumbImage
FROM stock p
INNER JOIN productcategories pc
ON p.itemID = pc.itemID
INNER JOIN categories c
ON pc.categoryID = c.categoryID
WHERE (p.display = 2 OR p.display = 3)
AND c.departmentID = ?
ORDER BY p.display DESC
LIMIT ?, ?";
SET @p1 = inShortProductDescriptionLength;
SET @p2 = inShortProductDescriptionLength;
SET @p3 = inDepartmentID;
SET @p4 = inStartItem;
SET @p5 = inProductsPerPage;
EXECUTE statement USING @p1, @p2, @p3, @p4, @p5;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_product_attributes` (IN inProductID INT)
BEGIN
SELECT a.name AS attribute_name,
av.attributeValueID, av.value AS attribute_value
FROM attributevalue av
INNER JOIN attribute a
ON av.attributeID = a.attributeID
WHERE av.attributeValueID IN
(SELECT attributeValueID
FROM productattribute
WHERE itemID = inProductID)
ORDER BY a.name;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_product_details` (IN inProductID INT)
BEGIN
SELECT itemID, itemName, longDescription,
price, discountedPrice, bigImage
FROM stock
WHERE itemID = inProductID;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_product_info`(IN
inProductId INT)
BEGIN
SELECT itemID, itemName, briefDescription, longDescription,
supplierID, price, discountedPrice,
thumbImage, bigImage, display
FROM stock
WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_get_product_locations`(IN inProductID INT)
BEGIN
SELECT c.categoryID, c.categoryName AS category_name,
c.departmentID,
(SELECT name
FROM department
WHERE departmentID = c.departmentID) AS department_name
FROM categories c
WHERE c.categoryID IN
(SELECT categoryID
FROM productcategories
WHERE itemID = inProductID);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_get_product_name`(IN
inProductId INT)
BEGIN
SELECT itemName FROM stock WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_move_product_to_category`(IN inProductId INT,
IN inSourceCategoryId INT, IN inTargetCategoryId INT)
BEGIN
UPDATE productcategories
SET categoryID = inTargetCategoryId
WHERE itemID = inProductId
AND categoryID = inSourceCategoryId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_remove_product_attribute_value`(
IN inProductId INT, IN inAttributeValueId INT)
BEGIN
DELETE FROM productattribute
WHERE itemID = inProductId AND
attributeValueID = inAttributeValueId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_remove_product_from_category`(
IN inProductId INT, IN inCategoryId INT)
BEGIN
DECLARE productCategoryRowsCount INT;

```

```

SELECT count(*)
FROM productcategories
WHERE itemID = inProductId
INTO productCategoryRowsCount;
IF productCategoryRowsCount = 1 THEN
CALL catalog_delete_product(inProductId);
SELECT 0;
ELSE
DELETE FROM productcategories
WHERE categoryID = inCategoryId AND itemID = inProductId;
SELECT 1;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_set_image`(
IN inProductId INT, IN inImage VARCHAR(150))
BEGIN
UPDATE stock SET thumbImage = inImage WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_set_image_2`(
IN inProductId INT, IN inImage VARCHAR(150))
BEGIN
UPDATE stock SET bigImage = inImage WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_set_product_display_option`(
IN inProductId INT, IN inDisplay SMALLINT)
BEGIN
UPDATE stock SET display = inDisplay WHERE itemID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_update_attribute`(
IN inAttributeId INT, IN inName VARCHAR(100))
BEGIN
UPDATE attribute SET name = inName WHERE attributeID =
inAttributeId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`catalog_update_attribute_value`(
IN inAttributeValueId INT, IN inValue VARCHAR(100))
BEGIN
UPDATE attributevalue
SET value = inValue
WHERE attributeValueID = inAttributeValueId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_update_category`(IN
inCategoryId INT,
IN inName VARCHAR(25), IN inDescription VARCHAR(500))
BEGIN
UPDATE categories
SET categoryName = inName, description = inDescription
WHERE categoryID = inCategoryId;

```

```
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_update_department`(IN  
inDepartmentId INT,  
IN inName VARCHAR(100), IN inManager INT, IN inNumber TEXT, IN  
inDescription VARCHAR(500))  
BEGIN  
UPDATE department  
SET name = inName, manager = inManager, telephoneNumber = inNumber,  
description = inDescription  
WHERE departmentID = inDepartmentId;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `catalog_update_product`(IN  
inProductId INT,  
IN inName VARCHAR(100), IN inBriefDescription VARCHAR(500), IN  
inLongDescription VARCHAR(1000),  
IN inSupplier INT, IN inPrice DECIMAL(10, 2), IN inDiscountedPrice  
DECIMAL(10, 2))  
BEGIN  
UPDATE stock  
SET itemName = inName, briefDescription = inBriefDescription,  
longDescription = inLongDescription,  
supplierID = inSupplier, price = inPrice, discountedPrice =  
inDiscountedPrice  
WHERE itemID = inProductId;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_add`(IN inFirstName  
VARCHAR(30),  
IN inSurname VARCHAR(30),  
IN inEmail VARCHAR(80), IN inPassword VARCHAR(50), IN  
inOriginalPassword VARCHAR(50),  
IN inAddressLineOne VARCHAR(50), IN inTown VARCHAR(50), IN inCity  
VARCHAR(50),  
IN inCounty VARCHAR(50), IN inPostCode VARCHAR(8), IN inTelephone  
TEXT)  
BEGIN  
INSERT INTO contactdetails (addressLineOne, town, city, county,  
postcode, telephoneNumber)  
VALUES (inAddressLineOne, inTown, inCity, inCounty, inPostCode,  
inTelephone);  
INSERT INTO customer (firstname, surname, emailAddress, password,  
unencPassword, contactID)  
VALUES (inFirstName, inSurname, inEmail, inPassword,  
inOriginalPassword, (SELECT contactID  
FROM contactdetails  
WHERE addressLineOne = inAddressLineOne  
AND postcode = inPostCode));  
SELECT LAST_INSERT_ID();  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_delete_customer`(IN  
inCustomerId INT)  
BEGIN  
DELETE FROM customer WHERE customerID = inCustomerId;
```

```
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_get_customer`(IN  
inCustomerId INT)  
BEGIN  
SELECT c.customerID, c.firstname, c.surname, c.emailAddress,  
c.password, c.unencPassword, c.creditCard,  
cd.addressLineOne, cd.town, cd.city, cd.county, cd.postcode,  
cd.telephoneNumber  
FROM customer c, contactdetails cd  
WHERE c.customerID = inCustomerId  
AND c.contactID = cd.contactID;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_get_customers_list`()  
BEGIN  
SELECT customerID, firstname, surname, emailAddress FROM customer  
ORDER BY surname ASC;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_get_login_info`(IN  
inEmail VARCHAR(100))  
BEGIN  
SELECT customerID, password FROM customer WHERE emailAddress =  
inEmail;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_get_unencPassword`(IN  
inEmail VARCHAR(80))  
BEGIN  
SELECT unencPassword  
FROM customer  
WHERE emailAddress = inEmail;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `customer_update_account`(IN  
inCustomerId INT,  
IN inFirstName VARCHAR(30), IN inSurname VARCHAR(30), IN inEmail  
VARCHAR(80),  
IN inPassword VARCHAR(50), IN inOriginalPassword VARCHAR(50), IN  
inTelephone TEXT, IN inAddressLineOne VARCHAR(50), IN inTown  
VARCHAR(50),  
IN inCity VARCHAR(50), IN inCounty VARCHAR(50), IN inPostCode  
VARCHAR(8))  
BEGIN  
IF (inPassword = null OR inOriginalPassword = null) THEN  
UPDATE customer c, contactdetails cd  
SET c.firstname = inFirstName, c.surname = inSurname, c.emailAddress  
= inEmail, cd.telephoneNumber = inTelephone, cd.addressLineOne =  
inAddressLineOne, cd.city = inCity,  
cd.town = inTown, cd.county = inCounty, cd.postcode = inPostCode  
WHERE c.customerID = inCustomerId  
AND c.contactID = cd.contactID;  
ELSE  
UPDATE customer c, contactdetails cd
```

```

SET c.firstname = inFirstName, c.surname = inSurname, c.emailAddress
= inEmail, c.password = inPassword, c.unencPassword =
inOriginalPassword, cd.telephoneNumber = inTelephone,
cd.addressLineOne = inAddressLineOne, cd.city = inCity,
cd.town = inTown, cd.county = inCounty, cd.postcode = inPostCode
WHERE c.customerID = inCustomerId
AND c.contactID = cd.contactID;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `customer_update_address`(IN
inCustomerId INT,
IN inAddressLineOne VARCHAR(50), IN inTown VARCHAR(50),
IN inCity VARCHAR(50), IN inCounty VARCHAR(50),
IN inPostCode VARCHAR(8))
BEGIN
UPDATE customer c, contactdetails cd
SET cd.addressLineOne = inAddressLineOne, cd.town = inTown, cd.city
= inCity,
cd.county = inCounty, cd.postcode = inPostCode
WHERE c.customerID = inCustomerId
AND c.contactID = cd.contactID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `customer_update_credit_card`(IN
inCustomerId INT, IN inCreditCard TEXT)
BEGIN
UPDATE customer
SET creditcard = inCreditCard
WHERE customerID = inCustomerId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_create_audit`(IN
inOrderId INT,
IN inMessage TEXT, IN inCode INT)
BEGIN
INSERT INTO audit (order_id, created_on, message, code)
VALUES (inOrderId, NOW(), inMessage, inCode);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_get_audit_trail`(IN
inOrderId INT)
BEGIN
SELECT audit_id, order_id, created_on, message, code
FROM audit
WHERE order_id = inOrderId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_get_by_customer_id`(IN
inCustomerId INT)
BEGIN
SELECT o.orderID, o.totalAmount, o.createdOn,
o.shippedOn, o.orderStatus, c.firstname, c.surname
FROM orders o
INNER JOIN customer c
ON o.customerID = c.customerID

```

```

WHERE o.customerID = inCustomerID
ORDER BY o.createdOn DESC;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`orders_get_most_recent_orders`(IN inHowMany INT)
BEGIN
PREPARE statement FROM
"SELECT o.orderID, o.totalAmount, o.createdOn,
o.shippedOn, o.orderStatus, c.firstname, c.surname
FROM orders o
INNER JOIN customer c
ON o.customerID = c.customerID
ORDER BY o.createdOn DESC
LIMIT ?";
SET @p1 = inHowMany;
EXECUTE statement USING @p1;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`orders_get_orders_between_dates`(
IN inStartDate DATETIME, IN inEndDate DATETIME)
BEGIN
SELECT o.orderID, o.totalAmount, o.createdOn,
o.shippedOn, o.orderStatus, c.firstname, c.surname
FROM orders o
INNER JOIN customer c
ON o.customerID = c.customerID
WHERE o.createdOn >= inStartDate AND o.createdOn <= inEndDate
ORDER BY o.createdOn DESC;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`orders_get_orders_by_status`(IN inStatus enum('Received','Checking
Funds','Notifying Stock Check','Awaiting Stock
Confirmation','Awaiting Payment','Notifying Warehouse
Despatch','Awaiting Despatch Confirmation','Notifying
Customer','Complete','Cancelled'))
BEGIN
SELECT o.orderID, o.totalAmount, o.createdOn,
o.shippedOn, o.orderStatus, c.firstname, c.surname
FROM orders o
INNER JOIN customer c
ON o.customerID = c.customerID
WHERE o.orderstatus = inStatus
ORDER BY o.createdOn DESC;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_get_order_details`(`IN
inOrderId INT)
BEGIN
SELECT orderID, itemID, attributes, itemName,
orderQuantity, price, (orderQuantity * price) AS subtotal
FROM orderitems
WHERE orderID = inOrderId;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_get_order_info`(IN
inOrderId INT)
BEGIN
SELECT o.orderID, o.totalAmount, o.createdOn, o.shippedOn,
o.orderStatus,
o.comments, o.authCode, o.reference, o.shippingID, s.shippingType,
s.shippingCost, o.customerID, c.firstname, c.surname,
cd.addressLineOne, cd.town, cd.city,
cd.county, cd.postcode, cd.telephoneNumber, c.emailAddress
FROM orders o, customer c, contactdetails cd, shipping s
WHERE orderID = inOrderId
AND o.customerID = c.customerID
AND c.contactID = cd.contactID
AND s.shippingID = o.shippingID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`orders_get_order_short_details`(IN inOrderId INT)
BEGIN
SELECT o.orderID, o.totalAmount, o.createdOn,
o.shippedOn, o.orderStatus, c.firstname, c.surname
FROM orders o
INNER JOIN customer c
ON o.customerID = c.customerID
WHERE o.orderID = inOrderId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_get_shipping_info`() 
BEGIN
SELECT shippingID, shippingType, shippingCost
FROM shipping;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_set_auth_code`(IN
inOrderId INT,
IN inAuthCode VARCHAR(50), IN inReference VARCHAR(50))
BEGIN
UPDATE orders
SET authCode = inAuthCode, reference = inReference
WHERE orderID = inOrderId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_set_date_shipped`(IN
inOrderId INT)
BEGIN
UPDATE orders SET shippedOn = NOW() WHERE orderID = inOrderId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_update_order`(IN
inOrderId INT, IN inStatus ENUM('Received', 'Checking
Funds', 'Notifying Stock Check', 'Awaiting Stock
Confirmation', 'Awaiting Payment', 'Notifying Warehouse
Despatch', 'Awaiting Despatch Confirmation', 'Notifying
Customer', 'Complete', 'Cancelled'),
IN inComments VARCHAR(255), IN inAuthCode VARCHAR(50),

```

```

IN inReference VARCHAR(50), IN inFirstName VARCHAR(30), IN inSurname
VARCHAR(30),
IN inAddressLineOne VARCHAR(50), IN inTown VARCHAR(50), IN inCity
VARCHAR(50),
IN inCounty VARCHAR(50), IN inPostCode VARCHAR(8), IN
inCustomerEmail VARCHAR(50), IN inTelephone TEXT)
BEGIN
DECLARE currentDateShipped DATETIME;
SELECT shippedOn
FROM orders
WHERE orderID = inOrderId
INTO currentDateShipped;
UPDATE orders o, customer c, contactdetails cd
SET o.orderStatus = inStatus, o.comments = inComments,
o.authCode = inAuthCode, o.reference = inReference,
c.firstname = inFirstName,
c.surname = inSurname,
cd.addressLineOne = inAddressLineOne,
cd.town = inTown,
cd.city = inCity,
cd.county = inCounty,
cd.postcode = inPostCode,
c.emailAddress = inCustomerEmail,
cd.telephoneNumber = inTelephone
WHERE o.orderID = inOrderId
AND o.customerID = c.customerID
AND c.contactID = cd.contactID;
IF inStatus = 'Received' OR inStatus = 'Checking Funds' OR inStatus
= 'Notifying Stock Check' OR inStatus = 'Awaiting Stock
Confirmation' OR inStatus = 'Awaiting Payment' OR inStatus =
'Notifying Warehouse Despatch' OR inStatus = 'Notifying Customer' OR
inStatus = 'Complete' OR inStatus = 'Cancelled' THEN
UPDATE orders SET shippedOn = NULL WHERE orderID = inOrderId;
ELSEIF inStatus != currentStatus AND inStatus = 'Awaiting Despatch
Confirmation' THEN
UPDATE orders SET shippedOn = NOW() WHERE orderID = inOrderId;
END IF;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `orders_update_status`(IN
inOrderId INT, IN inStatus ENUM('Received','Checking
Funds','Notifying Stock Check','Awaiting Stock
Confirmation','Awaiting Payment','Notifying Warehouse
Despatch','Awaiting Despatch Confirmation','Notifying
Customer','Complete','Cancelled'))
BEGIN
UPDATE orders SET orderStatus = inStatus WHERE orderID = inOrderId;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `shipping_add_shipping`(IN
inType VARCHAR(100), IN inCost DECIMAL(10, 2))
BEGIN
INSERT INTO shipping (shippingType, shippingCost) VALUES (inType,
inName);
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `shipping_delete_shipping`(IN
inShippingId INT)
BEGIN
DELETE FROM shipping WHERE shippingID = inShippingId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shipping_get_shipping`() 
BEGIN
SELECT shippingID, shippingType, shippingCost FROM shipping ORDER BY
shippingID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shipping_update_shipping`(IN
inShippingId INT, IN inType VARCHAR(100), IN inCost
DECIMAL(10,2))
BEGIN
UPDATE shipping SET shippingType = inType, shippingCost = inCost
WHERE shippingID = inShippingId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shopping_cart_add_product`(IN
inCartId CHAR(32),
IN inProductId INT, IN inAttributes VARCHAR(1000))
BEGIN
DECLARE productQuantity INT;
SELECT quantity
FROM shoppingcart
WHERE cartID = inCartId
AND itemID = inProductId
AND attributes = inAttributes
INTO productQuantity;
IF productQuantity IS NULL THEN
    INSERT INTO shoppingcart(cartID, itemID, attributes,
                           quantity, addedOn)
    VALUES (inCartId, inProductId, inAttributes, 1, NOW());
ELSE
    UPDATE shoppingcart
    SET quantity = quantity + 1, buyNow = true
    WHERE cartID = inCartId
    AND itemID = inProductId
    AND attributes = inAttributes;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_count_old_carts`(IN inDays INT)
BEGIN
SELECT COUNT(cartID) AS old_shopping_carts_count
FROM (SELECT cartID
      FROM shoppingcart
      GROUP BY cartID
      HAVING DATE_SUB(NOW(), INTERVAL inDays DAY) >= MAX(addedOn))
AS old_carts;
END$$

```

```

CREATE DEFINER=`emart`@`%` PROCEDURE `shopping_cart_create_order`(IN
inCartId CHAR(32),
IN inCustomerId INT, IN inShippingId INT)
BEGIN
DECLARE new_order INT;
INSERT INTO orders (createdOn, customerID, shippingID, orderStatus)
VALUES (NOW(), inCustomerId, inShippingId, 'Received');

SELECT LAST_INSERT_ID() INTO new_order;

INSERT INTO orderitems (orderID, itemID, attributes,
itemName, orderQuantity, price)
SELECT new_order, p.itemID, sc.attributes, p.itemName, sc.quantity,
COALESCE(NULLIF(p.discountedPrice, 0), p.price) AS price
FROM shoppingcart sc
INNER JOIN stock p
ON sc.itemID = p.itemID
WHERE sc.cartID = inCartId AND sc.buyNow;
UPDATE orders o
SET o.totalAmount = (SELECT SUM(price * orderQuantity)
FROM orderitems
WHERE orderID = new_order)
WHERE o.orderID = new_order;
CALL shopping_cart_empty(inCartId);
SELECT new_order;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_delete_old_carts`(IN inDays INT)
BEGIN
DELETE FROM shoppingcart
WHERE cartID IN
(SELECT cartID
FROM (SELECT cartID
FROM shoppingcart
GROUP BY cartID
HAVING DATE_SUB(NOW(), INTERVAL inDays DAY) >=
MAX(addedOn))
AS sc);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shopping_cart_empty`(IN
inCartId CHAR(32))
BEGIN
DELETE FROM shoppingcart WHERE cartID = inCartId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shopping_cart_get_products`(IN
inCartId CHAR(32))
BEGIN
SELECT sc.productID, p.itemName, sc.attributes,
COALESCE(NULLIF(p.discountedPrice, 0), p.price) AS price,
sc.quantity,
COALESCE(NULLIF(p.discountedPrice, 0),
p.price) * sc.quantity AS subtotal
FROM shoppingcart sc

```

```

INNER JOIN stock p
ON sc.itemID = p.itemID
WHERE sc.cartID = inCartId AND sc.buyNow;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_get_saved_products`(IN inCartId CHAR(32))
BEGIN
SELECT sc.productID, p.itemName, sc.attributes,
COALESCE(NULLIF(p.discountedPrice, 0), p.price) AS price
FROM shoppingcart sc
INNER JOIN stock p
ON sc.itemID = p.itemID
WHERE sc.cartID = inCartId AND NOT sc.buyNow;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_get_total_amount`(IN inCartId CHAR(32))
BEGIN
SELECT SUM(COALESCE(NULLIF(p.discountedPrice, 0), p.price)
* sc.quantity) AS total_amount
FROM shoppingcart sc
INNER JOIN stock p
ON sc.itemID = p.itemID
WHERE sc.cartID = inCartId AND sc.buyNow;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_move_product_to_cart`(IN inProductId INT)
BEGIN
UPDATE shoppingcart
SET buyNow = true, addedOn = NOW()
WHERE productID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_remove_product`(IN inProductId INT)
BEGIN
DELETE FROM shoppingcart WHERE productID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE
`shopping_cart_save_product_for_later`(IN inProductId INT)
BEGIN
UPDATE shoppingcart
SET buyNow = false, quantity = 1
WHERE productID = inProductId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `shopping_cart_update`(IN
inProductId INT, IN inQuantity INT)
BEGIN
IF inQuantity > 0 THEN
UPDATE shoppingcart
SET quantity = inQuantity, addedOn = NOW()
WHERE productID = inProductId;

```

```

ELSE
CALL shopping_cart_remove_product(inProductId);
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `stock_get_item`(IN inItemId
INT)
BEGIN
SELECT s.itemName, sc.idealQuantity, sc.currentQuantity, sc.status,
s.supplierID
FROM stock s, stockcontrol sc
WHERE sc.itemID = inItemId
AND s.itemID = inItemId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `stock_get_stock`() 
BEGIN
SELECT sc.itemID, s.itemName, sc.status FROM stock s, stockcontrol
sc
WHERE s.itemID = sc.itemID
ORDER BY itemName ASC;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `stock_update_stock`(IN
inItemId INT, IN inIdealQty INT, IN inCurrentQty INT)
BEGIN
UPDATE stockcontrol
SET idealQuantity = inIdealQty, currentQuantity = inCurrentQty
WHERE itemID = inItemId;
IF inCurrentQty <= 0 THEN
UPDATE stockcontrol
SET status = 'OutOfStock'
WHERE itemID = inItemId;
ELSEIF inCurrentQty >= (inIdealQty/2) AND inCurrentQty < inIdealQty
THEN
UPDATE stockcontrol
SET status = 'Medium'
WHERE itemID = inItemId;
ELSEIF inCurrentQty > inIdealQty THEN
UPDATE stockcontrol
SET status = 'Stocked'
WHERE itemID = inItemId;
ELSEIF inCurrentQty <= (inIdealQty/2) THEN
UPDATE stockcontrol
SET status = 'Reorder'
WHERE itemID = inItemId;
END IF;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_add`(IN inName
VARCHAR(30),
IN inEmail VARCHAR(80),
IN inAddressLineOne VARCHAR(50), IN inTown VARCHAR(50), IN inCity
VARCHAR(50),
IN inCounty VARCHAR(50), IN inPostCode VARCHAR(8), IN inTelephone
TEXT)

```

```

BEGIN
INSERT INTO contactdetails (addressLineOne, town, city, county,
postcode, telephoneNumber)
VALUES (inAddressLineOne, inTown, inCity, inCounty, inPostCode,
inTelephone);
INSERT INTO suppliers (supplierName, emailAddress, contactID)
VALUES (inName, inEmail, (SELECT contactID
FROM contactdetails
WHERE addressLineOne = inAddressLineOne
AND postcode = inPostCode));
SELECT LAST_INSERT_ID();
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_add_delivery`(IN
inSupplierId INT, IN inDate DATE, IN inTime TIME)
BEGIN
INSERT INTO supplierdeliveries(supplierID, dateOfDelivery,
timeOfDelivery)
VALUES (inSupplierId, inDate, inTime);
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_delete_supplier`(IN
inSupplierId INT)
BEGIN
DELETE FROM supplierDeliveries WHERE supplierID = inSupplierId;
DELETE FROM suppliers WHERE supplierID = inSupplierId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_get_deliveries`(IN
inSupplierId INT)
BEGIN
SELECT s.supplierName, d.deliveryID, d.dateOfDelivery,
d.timeOfDelivery
FROM suppliers s, supplierdeliveries d
WHERE s.supplierID = inSupplierId
AND d.supplierID = inSupplierId;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_get_delivery_info`(IN
inDeliveryId INT)
BEGIN
SELECT s.supplierName, d.dateOfDelivery, d.timeOfDelivery
FROM suppliers s, supplierdeliveries d
WHERE d.deliveryID = inDeliveryId
AND s.supplierID = d.supplierID;
END$$

CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_get_supplier`(IN
inSupplierId INT)
BEGIN
SELECT s.supplierID, s.supplierName, s.emailAddress,
cd.addressLineOne, cd.town, cd.city, cd.county, cd.postcode,
cd.telephoneNumber
FROM suppliers s, contactdetails cd
WHERE s.supplierID = inSupplierId
AND s.contactID = cd.contactID;

```

```
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_get_supplier_list`()  
BEGIN  
SELECT supplierID, supplierName, emailAddress FROM suppliers ORDER  
BY supplierName ASC;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_remove_delivery`(IN  
inDeliveryId INT)  
BEGIN  
DELETE FROM supplierdeliveries WHERE deliveryID = inDeliveryId;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_update_account`(IN  
inSupplierId INT,  
IN inName VARCHAR(30), IN inEmail VARCHAR(80),  
IN inTelephone TEXT, IN inAddressLineOne VARCHAR(50), IN inTown  
VARCHAR(50),  
IN inCity VARCHAR(50), IN inCounty VARCHAR(50), IN inPostCode  
VARCHAR(8))  
BEGIN  
UPDATE suppliers s, contactdetails cd  
SET s.supplierName = inName, s.emailAddress = inEmail,  
cd.telephoneNumber = inTelephone, cd.addressLineOne =  
inAddressLineOne, cd.city = inCity,  
cd.town = inTown, cd.county = inCounty, cd.postcode = inPostCode  
WHERE s.supplierID = inSupplierId  
AND s.contactID = cd.contactID;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `supplier_update_delivery`(IN  
inDeliveryId INT, IN inDate DATE, IN inTime TIME)  
BEGIN  
UPDATE supplierdeliveries  
SET dateOfDelivery = inDate, timeOfDelivery = inTime  
WHERE deliveryID = inDeliveryId;  
END$$
```

```
CREATE DEFINER=`emart`@`%` PROCEDURE `update_contact_id`(IN  
inAddressLineOne VARCHAR(50), IN inPostCode VARCHAR(8), IN  
inCustomerId INT)  
BEGIN  
UPDATE customer c, contactdetails cd SET c.contactID = cd.contactID  
WHERE cd.addressLineOne = inAddressLineOne AND cd.postcode =  
inPostCode AND c.customerID = inCustomerId;  
END$$
```

## Screen Shots

### Home Page

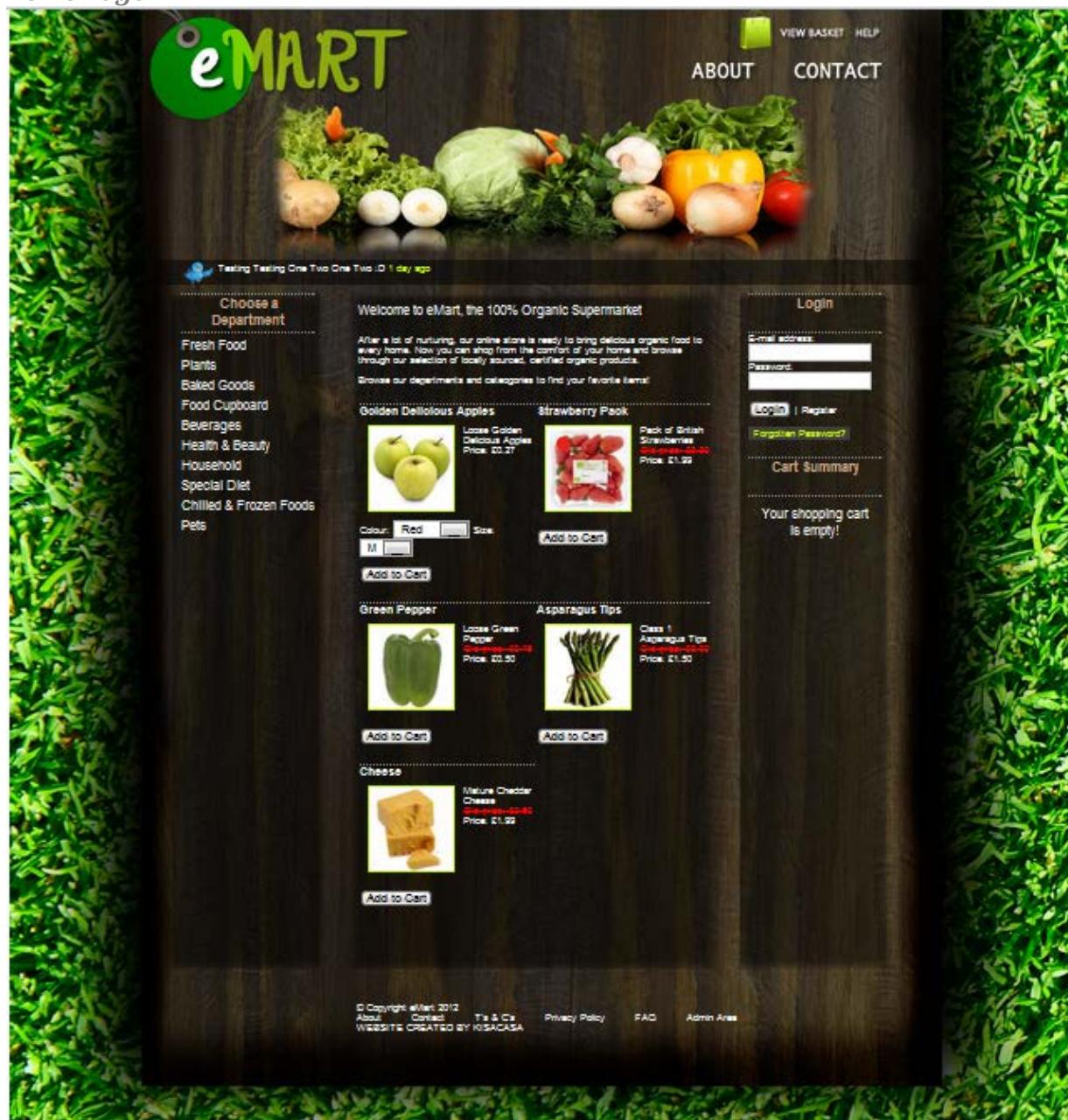


Figure 1

This is the home page for the front end of the eMart system.

## Contact Form



The screenshot shows the eMART contact form. At the top left is the eMART logo with a green leaf. Top right buttons include 'VIEW BASKET', 'HELP', 'ABOUT', and 'CONTACT'. A decorative banner at the top features various fresh vegetables like carrots, onions, and lettuce. On the left, there's a sidebar with a green grass background containing a list of departments: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. A message from 'Testing Testing One Two One Two' is visible. The main contact area has sections for 'Choose a Department' (with Fresh Food selected), 'Contact us' (Name and Email fields), 'Message' (Type Your Message field), and a 'Submit' button. To the right is a 'Login' section with fields for 'E-mail address' and 'Password', and buttons for 'Login | Register' and 'Forgotten Password?'. A note says 'Your shopping cart is empty!'. A timestamp '1 day ago' is also present.

Figure 2

## Customer Login Component



The screenshot shows the customer login component. It has a dark background with gold-colored text. The title 'Login' is at the top. Below it are fields for 'E-mail address:' and 'Password:', each with an input box. At the bottom are buttons for 'Login | Register' and 'Forgotten Password?'.

Figure 3

Customers can login to place orders

## Customer Forgotten Password



The screenshot shows the customer forgotten password component. It has a dark background with gold-colored text. The title 'Login' is at the top. Below it is a red message 'Reminder email sent.' followed by an 'E-mail address:' field containing 'valley1girl88@aol.com'. There is also a 'Password:' field with an input box. At the bottom are buttons for 'Login | Register' and 'Forgotten Password?'.

Figure 4

Customer enters password and a confirmation email is sent to the address.

### *Customer Registration*

Please enter your details:

Email Address:

First name:

Surname:

Password:

Re-enter Password:

Street Address:

Town:

City:

County:

Post Code:

Telephone:

|

Figure 5

This is where customers register on the eMart system. Validation note: Passwords must match, postcode must be less than 8 characters, telephone number must be numbers and 11 characters long, emails must be in the format name@mail.co.uk or .com

### *Department/Category Selection Component*

Choose a Department

Fresh Food

Plants

Baked Goods

Food Cupboard

Beverages

Health & Beauty

Household

Special Diet

Chilled & Frozen Foods

Pets

Choose a Category

Fresh Fruit

Vegetables

Salad

Dairy Produce

Fresh Meat

Figure 6

Customers can browse through products by department or category

### *Product Details*



The screenshot shows a product detail page for 'Asparagus Tips'. At the top, it says 'Asparagus Tips' in bold. Below that is a large image of a bunch of green asparagus tied with twine. To the right of the image, the text reads 'Pack of Asparagus Tips Class 1 125G' and 'Old Price: £2.00'. Underneath that is 'Price: £1.50'. A blue 'Add to Cart' button is visible. At the bottom left is a 'Continue Shopping' link, and at the bottom center is a 'Find similar products in our catalog' link, followed by 'Fresh Food » Vegetables'.

Figure 7

When a product if clicked on, more information can be viewed on it

### *Cart Summary Component*



The screenshot shows a 'Cart Summary' component. It has a header 'Cart Summary' in bold. Below it, there are two items listed: '1 x Green Pepper' and '1 x Asparagus Tips'. At the bottom, it shows a total price of '£2.00' with a '[View details]' link next to it.

Figure 8

A running summary is kept of the cart on the right hand pane

### *Cart Details showing 'Save for later' feature*

These are the products in your shopping cart:

Product Name	Price	Quantity	Subtotal	Tools
Asparagus Tips	£1.50	1	£1.50	<a href="#">Save for later</a> <a href="#">Remove</a>

Total amount: £1.50 [Update](#)

Saved products to buy later:

Product Name	Price	Tools
Green Pepper	£0.50	<a href="#">Move to cart</a> <a href="#">Remove</a>

[Continue Shopping](#)

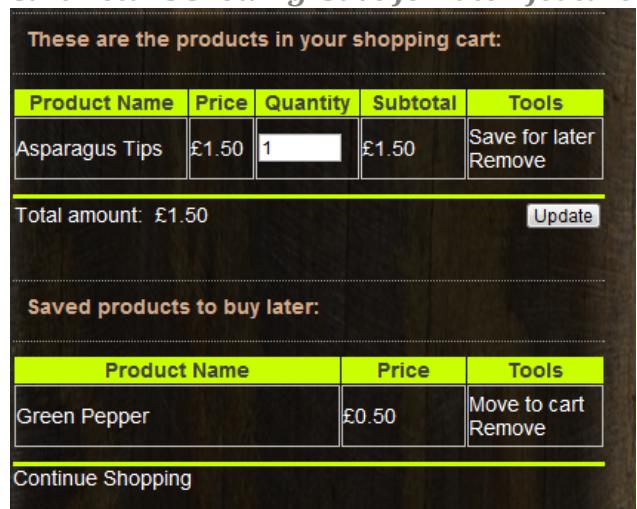


Figure 9

Customers can save their items to use at a later date

### *Checkout*

Your order consists of the following items:

Product Name	Price	Quantity	Subtotal
Green Pepper	0.50	1	0.50
Asparagus Tips	1.50	1	1.50

Total amount: £2.00

Credit card to use: Visa  
Card number: XXXX-XXXX-XXXX-1111

Shipping address:  
999 High Street  
Random  
Liverpool  
Merseyside  
L7 9YS

Shipping type: [Next Day Delivery \(£6.75\)](#)

[Place Order](#) | [Edit Shopping Cart](#) | [Continue Shopping](#)

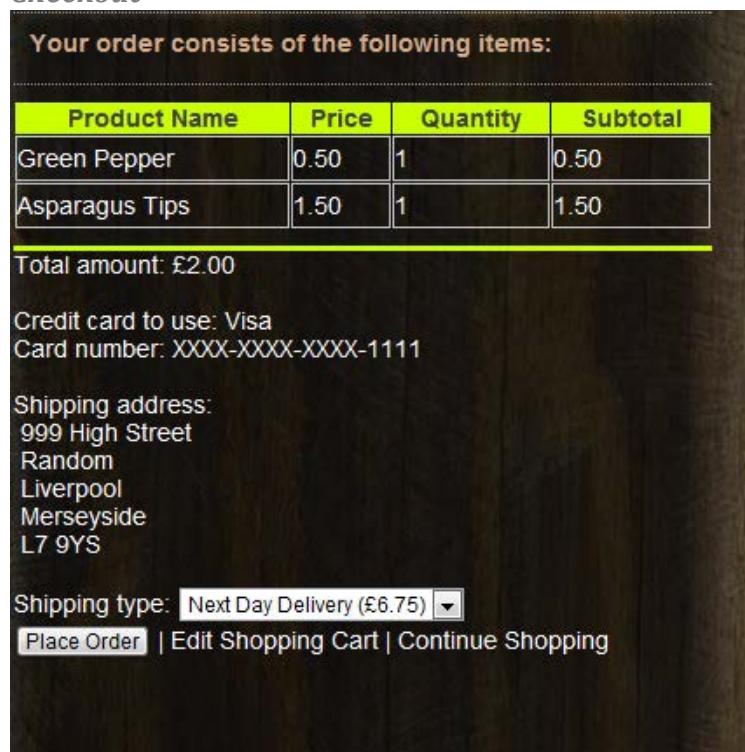


Figure 10

This is the checkout where the customer can change payment details and shipping type

### *Order Process*



**Thank you for your order!**

*A confirmation of your order will be sent to your registered email address.*

Figure 11

This confirms that the order has been processed

## Order Table State with Audit Trail

Editing details for order ID: 6 [ back to admin orders... ]

Total Amount:	£2.00
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-17 16:16:06
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	OPNW02
Reference Number:	2171975078
Comments:	
Customer Name:	kirsty williams
Shipping Address:	Street: 999 High Street Town: Random City: Liverpool County: Merseyside Postcode: L7 9YS
Customer Email:	kisacasa@hotmail.co.uk
Telephone Number:	01111111111

---

[Edit](#) [Update](#) [Cancel](#) [Confirm Stock for Order](#)

Order contains these products:

Product ID	Product Name	Quantity	Unit Cost	Subtotal
6	Green Pepper	1	£0.50	£0.50
8	Asparagus Tips	1	£1.50	£1.50

Order audit trail:

Audit ID	Created On	Code	Message
184	2012-05-17 16:16:06	10000	Order Processor started.
185	2012-05-17 16:16:06	20000	PsInitialNotification started.
186	2012-05-17 16:16:06	20002	Notification e-mail sent to customer.
187	2012-05-17 16:16:06	20001	PsInitialNotification finished.
188	2012-05-17 16:16:06	20100	PsCheckFunds started.
189	2012-05-17 16:16:07	20102	Funds available for purchase.
140	2012-05-17 16:16:07	20101	PsCheckFunds finished.
141	2012-05-17 16:16:07	20200	PsCheckStock started.
142	2012-05-17 16:16:07	20202	Notification e-mail sent to warehouse.
143	2012-05-17 16:16:07	20201	PsCheckStock finished.
144	2012-05-17 16:16:07	10001	Order Processor finished.

Figure 12

This displays the details of the order, including the audit trail of the order

### *Order Stock Check Email – Warehouse*

eMart stock check.

kisacasa@hotmail.co.uk  
To kisacasa@hotmail.co.uk

From: **kisacasa@hotmail.co.uk** ⓘ  
Sent: 18 May 2012 00:16:09  
To: kisacasa@hotmail.co.uk

 Microsoft SmartScreen has blocked this message for your safety and we'll delete it after ten days.  
[Show content](#)

The following products have been ordered:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Asparagus Tips() £1.50 each, total cost £1.50  
Shipping: Next Day Delivery (£6.75)

Total order cost: £8.75

Please check availability and confirm via <http://www.kisacasa.com/emart/admin.php>

Order reference number: 6

Figure 13

The warehouse must confirm that they have the stock to fulfill the order

### *Order Ship Goods Email – warehouse*

Editing details for order ID: 6 [ back to admin orders... ]

<b>Total Amount:</b>	£2.00
<b>Shipping:</b>	Next Day Delivery (£6.75)
<b>Date Created:</b>	2012-05-17 16:16:06
<b>Date Shipped:</b>	
<b>Status:</b>	Awaiting Despatch Confirmation
<b>Authorization Code:</b>	OPNWD2
<b>Reference Number:</b>	2171975078
<b>Comments:</b>	
<b>Customer Name:</b>	kirsty williams
<b>Shipping Address:</b>	<b>Street</b> 999 High Street <b>Town</b> Random <b>City</b> Liverpool <b>County</b> Merseyside <b>Postcode</b> L7 9YS
<b>Customer Email:</b>	kisacasa@hotmail.co.uk
<b>Telephone Number:</b>	01111111111

**Edit** **Update** **Cancel** **Confirm Shipment for Order**

Figure 14

Warehouse has confirmed stock. Waiting for dispatch confirmation

## *Confirm order dispatch- warehouse*

eMart ship goods.

kisacasa@hotmail.co.uk  
To kisacasa@hotmail.co.uk

From: **kisacasa@hotmail.co.uk** ⓘ  
Sent: 18 May 2012 02:21:48  
To: kisacasa@hotmail.co.uk

Microsoft SmartScreen has blocked this message for your safety and we'll delete it after ten days.  
[Show content](#)

Payment has been received for the following goods:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Asparagus Tips() £1.50 each, total cost £1.50  
Shipping: Next Day Delivery (£6.75)

Total order cost: £8.75

Please ship to:

Kirsty Williams  
999 High Street  
Random  
Liverpool  
Merseyside  
L7 9YS

Figure 15

The warehouse must confirm that the order has been shipped

## *Payment successful- Authorise.NET*

Subject: Merchant Email Receipt  
From: "Authorize.Net Auto-Receipt" <developer@authorize.net>  
Date: Fri, May 18, 2012 12:16 am  
To: "TestFirstName TestLastName" <kisacasa@kirstywilliams.co.uk>

Figure 16

Email for Authorise.NET that payment has been successfully processed.

### Order Shipment Email – Customer

From: kisacasa@hotmail.co.uk   
Sent: 18 May 2012 02:26:44  
To: kisacasa@hotmail.co.uk

 This message looks suspicious to our SmartScreen filters and we'll delete it after ten days.

Your order has now been dispatched! The following products have been shipped:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Asparagus Tips() £1.50 each, total cost £1.50  
Shipping: Next Day Delivery (£<sup>6.75</sup>)

Total order cost: £8.75

Your order has been shipped to:

kirsty williams  
999 High Street  
Random  
Liverpool  
Merseyside  
L7 9YS

Order reference number: 6

Thank you for shopping at eMart!

Figure 17

Confirmation for the customer that the goods have been dispatched

### Warehouse Login



Kisacasa eCommerce Solution

Support

STORE ADMINISTRATION

### Site Administrator Login

Enter login information or go back to eMart Storefront.

Username:

Password:

[Forgotten Password?](#)

Figure 18

Staff can log in to the system

## *Warehouse Forgotten Password Email*

Password reminder requested.

kisacasa@hotmail.co.uk  
To kisacasa@hotmail.co.uk

From: **kisacasa@hotmail.co.uk** ⓘ  
Sent: 18 May 2012 00:22:22  
To: kisacasa@hotmail.co.uk

 This message looks suspicious to our SmartScreen filters and we'll delete it after ten days.

Message: You have requested a password reminder:

Your current password is "Password2012"

For security purposes we recommend this is changed.  
This is done by accessing the configuration file,  
located in the include directory, and changing the  
administrator login information password

Figure 19

When an employee requests a password, they are sent a password reminder to  
[kisacasa@hotmail.co.uk](mailto:kisacasa@hotmail.co.uk)

### *Warehouse Departments View*



Kisacasa eCommerce Solution      Support

STORE ADMINISTRATION

STOREFRONT      LOGOUT

| CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |

eMart Departments				
Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Chilled & Frozen Foods	0	01510000009	Here you will find our chilled and frozen food selection.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Pets	0	01510000010	Spoil your adored pets with our large selection of luxurious treats and toys.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

Add new department:

© Copyright 2012  
CREATED BY KISACASA

Figure 20

This is the main screen of the admin area. It's the highest level of the system hierarchy

## Edit Product Details View



Kisacasa eCommerce Solution Support  
STORE ADMINISTRATION

STOREFRONT LOGOUT  
CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING

Editing product: ID #2 — Golden Delicious Apples [back to products ... ]

Product name: Golden Delicious Apples	Product belongs to these categories: Fresh Fruit Remove this product from: Fresh Fruit <input type="button" value="Remove"/>
Product brief description: Locally Golden delicious Apples	Assign product to this category: Vegetables <input type="button" value="Assign"/> Move product to this category: Vegetables <input type="button" value="Move"/>
Product long description: Locally Golden delicious Apples Class 1	
Product supplier: 1	Product attributes: Color: Red <input type="button" value="Remove"/> Assign attribute to product: Size: XS <input type="button" value="Assign"/>
Product price: 0.27	Set display option for this product: On Sale <input type="button" value="Set"/>
Product discounted price: 0.00	<input type="button" value="Remove product from catalog"/>

Update Info

Thumbnail name: goldendelicious.jpg  No file chosen



Large Image name: goldendeliciouslrg.jpg  No file chosen



© Copyright eMart 2012

Figure 21

This is where the information on products can be edited

## Warehouse Stock Control Details



Kisacasa eCommerce Solution Support  
STORE ADMINISTRATION

STOREFRONT LOGOUT  
CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING

eMart Stock

Item ID	Item Name	Status	Tools
8	Asparagus Tips	Stocked	<input type="button" value="View Details"/>
4	Banana	Reorder	<input type="button" value="View Details"/>
9	Broccoli	Medium	<input type="button" value="View Details"/>
10	Brusel Sprouts	OutOfStock	<input type="button" value="View Details"/>
11	Butter	Stocked	<input type="button" value="View Details"/>
15	Cheese	Medium	<input type="button" value="View Details"/>
3	Conference Pears	Medium	<input type="button" value="View Details"/>
14	Eggs	Medium	<input type="button" value="View Details"/>
7	Garlic	Medium	<input type="button" value="View Details"/>
2	Golden Delicious Apples	Stocked	<input type="button" value="View Details"/>
6	Green Pepper	Stocked	<input type="button" value="View Details"/>
12	Pouring Cream	Stocked	<input type="button" value="View Details"/>
18	Semi Skimmed Milk	Reorder	<input type="button" value="View Details"/>
5	Strawberry Pack	OutOfStock	<input type="button" value="View Details"/>

Figure 22

This is the stock control page

Carts View

The screenshot shows the 'Carts View' section of the eMart application. At the top, there's a logo for 'eMART' with a green shopping bag icon. To the right are links for 'Kisacasa eCommerce Solution' and 'Support'. Below that is a 'STORE ADMINISTRATION' link. A navigation bar at the bottom has 'STOREFRONT' and 'LOGOUT' buttons, along with links for 'CATALOG', 'STOCK CONTROL', 'PRODUCTS ATTRIBUTES', 'CARTS', 'ORDERS', 'CUSTOMERS', 'SUPPLIERS', and 'SHIPPING'. The main content area displays a message: 'eMart Shopping Carts: There are 6 old shopping carts (selected option: All shopping carts.)'. It includes buttons for 'Select carts: All shopping carts', 'Count Old Shopping Carts', and 'Delete Old Shopping Carts'. At the bottom, it says '© Copyright eMart 2012' and 'CREATED BY KISACASA'.

Figure 23

This feature count the carts which are stored on the system

Orders Main View

The screenshot shows the 'Orders Main View' section of the eMart application. It features the same 'eMART' logo and navigation links as the previous page. The main content area is titled 'eMart Orders' and contains several search and filter options: 'Show orders by customer' dropdown set to 'kirsty williams' with a 'Go!' button; 'Get by order ID' input field with a 'Go!' button; 'Show the most recent 20 orders' dropdown with a 'Go!' button; 'Show all records created between' date range input fields with a 'Go!' button; and 'Show orders by status' dropdown set to 'Received' with a 'Go!' button. The footer includes the copyright notice '© Copyright eMart 2012' and 'CREATED BY KISACASA'.

Figure 24

This is the page where employees can search for orders

*Orders View with a filtered search*

The screenshot shows the 'eMart Orders' page. At the top, there are several search filters: 'Show orders by customer' set to 'kirsty williams' with a 'Go!' button; 'Get by order ID' with a dropdown and 'Go!' button; 'Show the most recent 20 orders' with a 'Go!' button; 'Show all records created between' with two date inputs and a 'Go!' button; and 'Show orders by status' set to 'Received' with a 'Go!' button. Below these filters is a table titled 'eMart Orders' containing four rows of order data:

Order ID	Date Created	Date Shipped	Status	Customer	Tools
6	2012-05-17 16:16:06		Awaiting Stock Confirmation	kirsty williams	<a href="#">View Details</a>
5	2012-05-16 21:16:38	2012-05-16 21:26:38	Complete	kirsty williams	<a href="#">View Details</a>
3	2012-05-15 22:19:19	2012-05-16 01:39:20	Complete	kirsty williams	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	kirsty williams	<a href="#">View Details</a>

Figure 25

This is displaying the most recent orders

*Order Details View*

The screenshot shows the 'Editing details for order ID: 6 [ back to admin orders... ]'. The form fields include:

- Total Amount: £2.00
- Shipping: Next Day Delivery (£6.75)
- Date Created: 2012-05-17 16:16:06
- Date Shipped: Awaiting Stock Confirmation
- Status: Awaiting Stock Confirmation
- Authorization Code: 0PNWD2
- Reference Number: 2171975078
- Comments: (empty)
- Customer Name: kirsty williams
- Shipping Address: Street: 999 High Street, Town: Random, City: Liverpool, County: Merseyside, Postcode: L7 9YS
- Customer Email: kisacasa@hotmail.co.uk
- Telephone Number: 01111111111

At the bottom of the form are buttons for Edit, Update, Cancel, and Confirm Stock for Order.

Figure 26

This displays the details of an order

### *Storefront View (logged in as employee)*

The screenshot shows a storefront interface for employees. On the left, a sidebar titled "Choose a Department" lists categories like Fresh Food, Plants, Baked Goods, etc. "Fresh Food" is selected and highlighted in yellow. The main content area is titled "Fresh Food" and contains a descriptive text about locally sourced organic fruit, vegetables, and dairy products. Below this is a navigation bar with links "1 2 Next page". Two product cards are displayed: "Strawberry Pack" (image of strawberries, price £1.99) and "Golden Delicious Apples" (image of apples, price £0.27). Each card has an "Add to Cart" button. At the bottom, there are dropdown menus for "Colour: Red" and "Size: M", and a "Edit Product Details" button.

Figure 27

When logged into the admin area, an employee can edit the products on the storefront

### *Live Twitter feed*

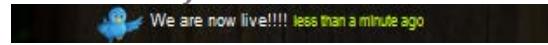


Figure 28

Shows the live Twitter feed of the website

## Appendix D

# Testing

---

## **Overview**

This document will identify the tests required to test whether the implementation has met the specification and specify the results of these tests.

## **Introduction to testing**

This document will test Kisacasa's ecommerce solution. It will test the full functions of the website ([www.kisacasa.com](http://www.kisacasa.com)). The functions of the website to be tested will be:

- The shopping cart
- Payment processing
- Data entry (ensuring it is saving to the database correctly)
- Data editing/deletion
- Navigation through the store front
- Calculations (e.g. running total of the shopping cart)
- Error logging
- Searches
- Email correspondence
- Password testing

## **Objectives and tasks**

The main objective for testing is to find any errors in the software and to correct them before the website goes live to customers. It is crucial that the system is meticulously experimented with so that when customers do visit the site, it will not have unfound bugs.

The tasks for the test document are-

- Testing
- Error reporting
- Error debugging

## **Scope of the testing**

All features of the website are to be tested by Kisacasa. We will try the website in several different web browsers to ensure the layout and features work consistently in all of them.

The tests will be executed under the following headings:

- Browser compatibility
- Testing browsing (navigating through the website in the tested browsers)
- Functional testing
- Non-functional testing
- Component testing

## **Testing strategy**

The overall approach to testing is to step by step utilise the system how a customer would. We aim for the testing to be very fluid in that each step will be somewhat related to the last. This way, the testing team can see through the users eyes how the system would be used.

## Participants

- Samantha Kviecinskas
- Sara Chattun

Both participants will carry out the tests and collect evidence throughout the testing. Once all the testing has finished, the team members will compare results and discuss any problems that the system has. Testing will be carried out locally prior to deployment.

## Reference Materials

### Requirements Document

The requirements document (Appendix A) specifies the requirements of the project. These will be referenced to distinguish expected results of the test runs.

### Design Document

The design document (Appendix B) specifies the design anticipated. These will be referenced to distinguish expected results of the test runs.

## Specific Testing Areas

### Static Testing

“Static testing is the testing of objects in a web browser that do not change, or are not transaction based.”

### Content Checking

This test will test for accuracy, consistency, spelling and accessibility.

### *How test will be carried out*

Two members of Kisacasa will check the website thoroughly for the above. A survey has been created (see appendix )which will be distributed to help collate results.

### Expected Result

No errors are anticipated.

### Actual Result

User 1:

- Accuracy – No issues noted.
- Consistency – No issues noted.
- Spelling – No issues noted.
- Accessibility – No issues noted.

User 2:

- Accuracy – No issues noted.
- Consistency – No issues noted.
- Spelling – No issues noted.
- Accessibility – No issues noted.

## Browser Compatibility

This test will look at whether the website will work in various browsers and whether the content looks the same, regardless of the browser used.

### Comodo Dragon

#### Expected Result

Website will display as intended.

#### Actual Result



Figure 29

### Google Chrome

#### Expected Result

Website will display as intended.

#### Actual Result



Figure 30

## Internet Explorer

### Expected Result

Website will display as intended.

### Actual Result



Figure 31

## Maxthon

### Expected Result

Website will display as intended.

### Actual Result



Figure 32

## Mozilla Firefox

### Expected Result

Website will display as intended.

### Actual Result

Figure



Figure 33

## Opera

### Expected Result

Website will display as intended.

### Actual Result

Figure



Figure 34

## Safari

### Expected Result

Website will display as intended.

### Actual Result

Figure

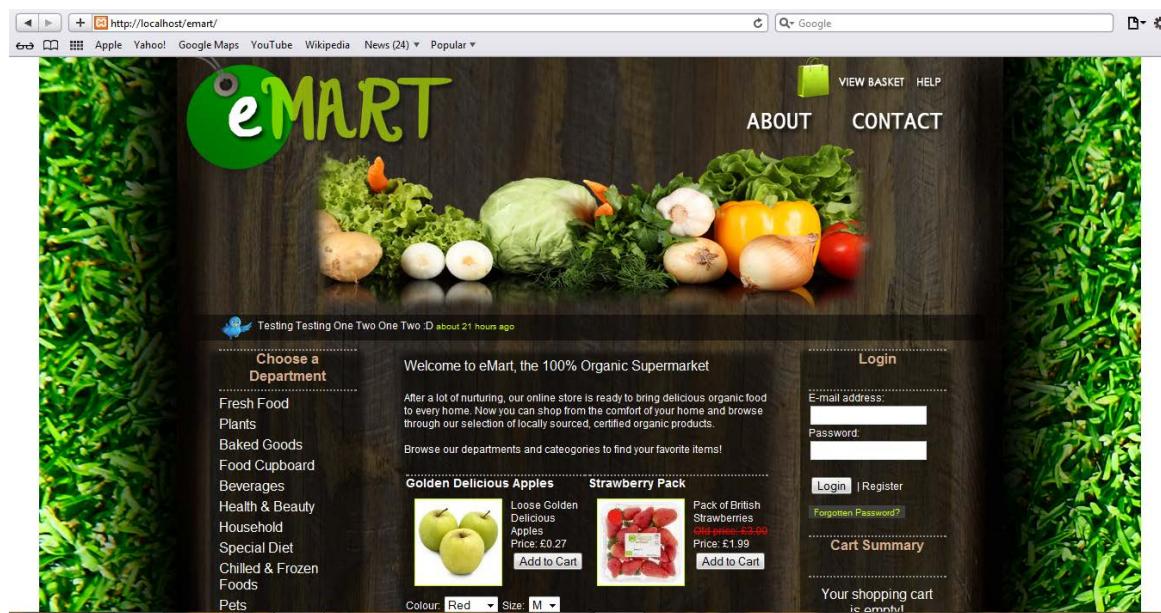


Figure 35

## Test Browsing

*“Test browsing tests aim to find the defects regarding navigation through web pages, including the availability of linked pages and other objects as well as the download speed of the individual page under test.”(Gerrard, 2000)*

### Browsing the Site

#### Expected Result

- All objects load in an acceptable time.
- All of the text and graphical links work.

#### Actual Result

- All objects load within an acceptable time which meets eMarts requirements
- All text and graphical links work as expected.

#### Action to be taken

- None

## Functional Testing

### Browser-Page Tests

*“This type of test covers the objects and code that executes within the browser ”(Gerrard, 2006).*

#### Expected Result

No errors expected.

#### Actual Result

No errors occurred.

## Non-Functional Testing

### Performance

*“Performance testing is the validation that the system meets performance requirements.”(Stout 2001)*

#### How it will be tested

The performance will be tested by using the website generally and making sure all load times are sufficient.

#### Expected Result

Everything will load within 2 seconds.

#### Actual Result

Every page loaded in sufficient time during testing.

### Security

Ensuring data is secure is one of the most important aspects of the symbol and needs to be tested thoroughly.

### **How it will be tested**

Testing will be done using test scripts (provided by Kirsty Williams) for the authorize net, card, encryption and password hasher. See *Appendices 1-4 of this documentation*.

### **Expected Result**

Using the password hasher the password should be encrypted without being able to unencrypt; because of the use of SHA1 hashing.

With the card encryption it shall be encrypted initially however the original data should still be retrievable later, after decryption.

### **Actual Result**

They worked as expected.

## **Component Testing**

This section will test individual components of the eMart system to see if they work as anticipated.

### **Test Strategy**

The testing will be done on the individual components based on the Use Cases specified in the Requirements Document. Print screens have been used to support the testing.

### **Test Case 1 – View error log**

System files>>

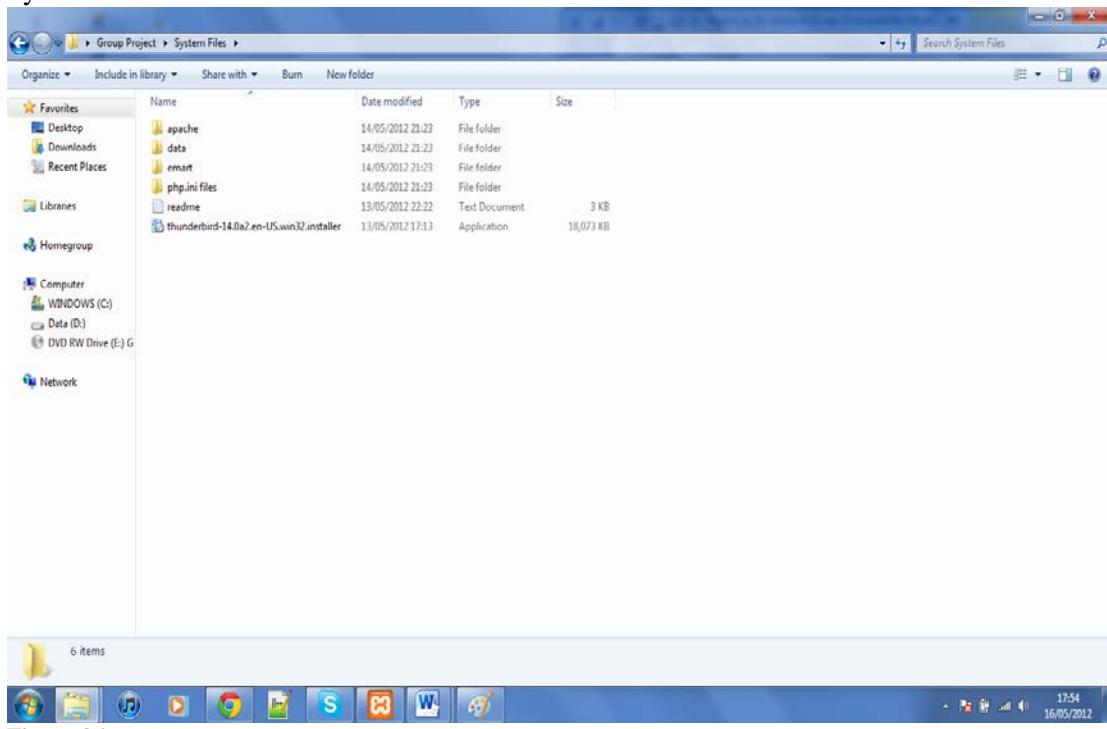


Figure 36

eMart>>

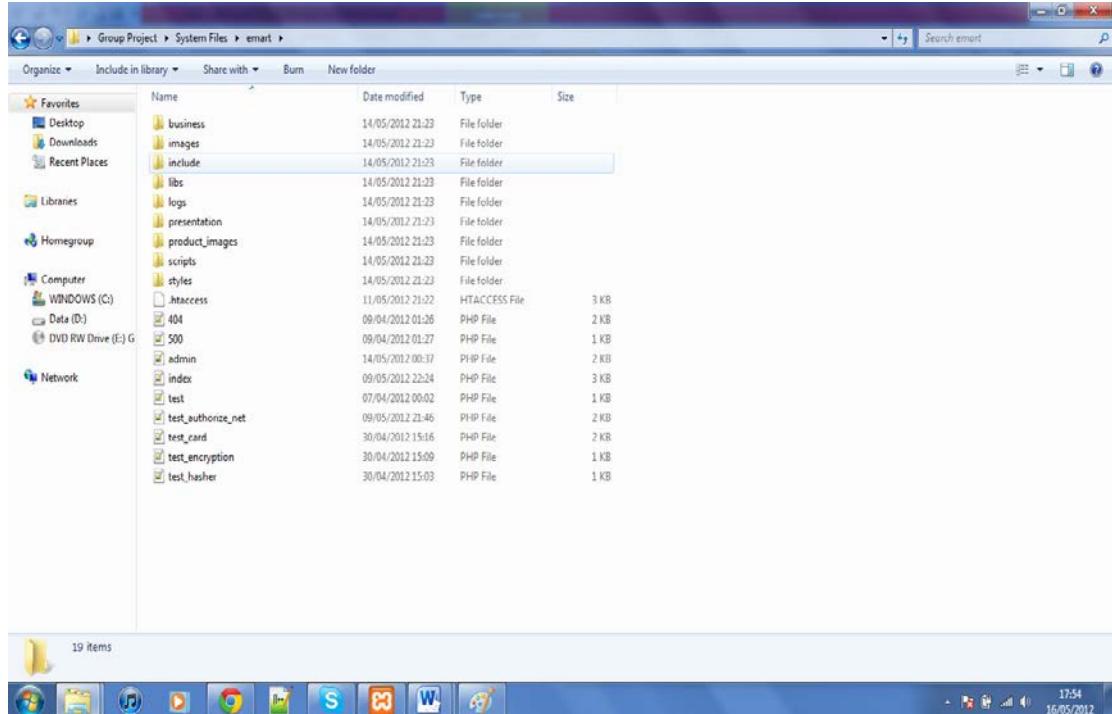


Figure 37

Logs&gt;&gt;

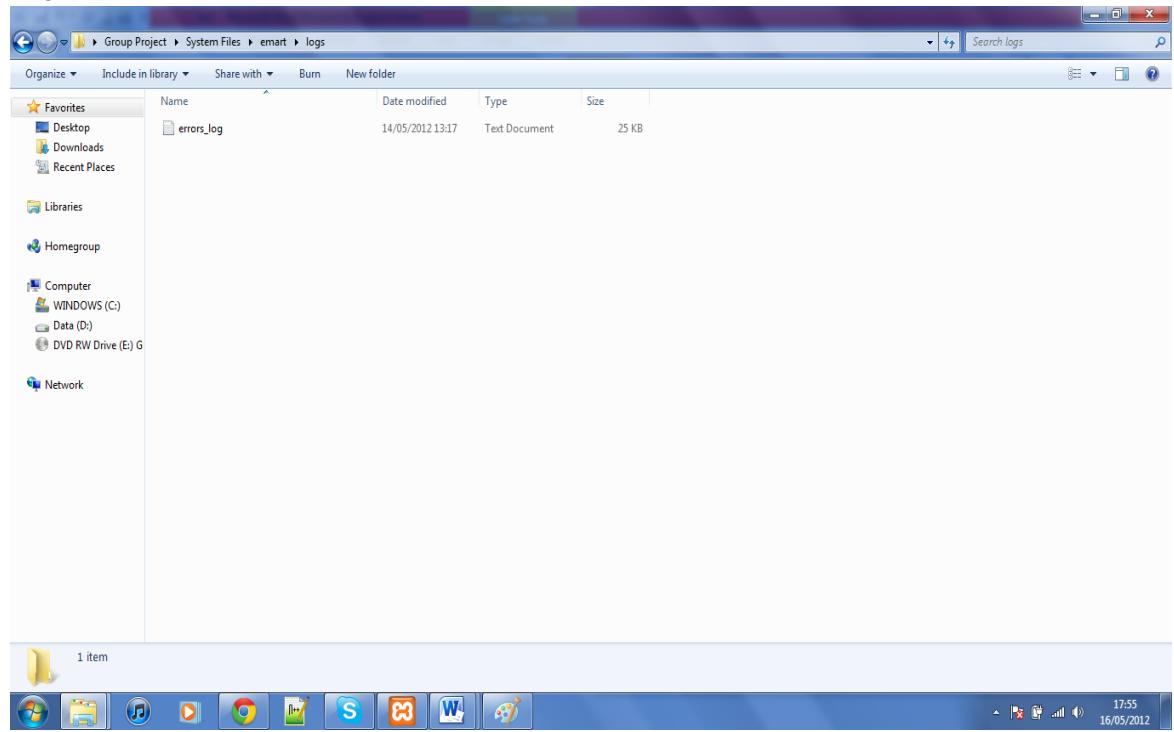


Figure 38

errors\_log

A screenshot of Notepad displaying the 'errors\_log.txt' file. The file contains two entries of error logs, each starting with 'ERRNO: 8' and 'TEXT: Undefined property: AdminMenu:\$mLinkToStockControl'. Both entries point to 'C:\emart\presentation\templates\_c\...'. The log details the stack trace for these errors, involving Smarty classes like Smarty\_Internal\_Template and Smarty\_Internal\_TemplateBase, and includes file paths such as 'C:\emart\libs\smarty\sysplugins\smarty\_internal\_templatebase.php'. The Notepad window shows standard Windows-style icons in the toolbar and status bar at the bottom.

Figure 39

## Test Case 2- Successfully register customer

Navigate to right hand pane>>Register>>Enter details>>Confirm

The screenshot shows the registration process on the eMart website. On the left, a sidebar lists departments: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. The main area has a form titled "Please enter your details:" with fields for Email Address, First name, Surname, Password, Re-enter Password, Street Address, Town, City, County, Post Code, and Telephone. The email, first name, surname, address, town, city, county, post code, and telephone fields are filled with sample data. To the right, a "Login" panel shows fields for E-mail address and Password, and links for Login, Register, and Forgotten Password. Below it is a "Cart Summary" section stating "Your shopping cart is empty!".

Figure 40

Customer is now registered and logged into the system

The screenshot shows the eMart homepage after login. The left sidebar remains the same. The main content area features a welcome message: "Welcome to eMart, the 100% Organic Supermarket". It says, "After a lot of nurturing, our online store is ready to bring delicious organic food to every home. Now you can shop from the comfort of your home and browse through our selection of locally sourced, certified organic products." Below this is a "Browse our departments and categories to find your favorite items!" section. Two product cards are displayed: "Golden Delicious Apples" (three green apples) and "Strawberry Pack" (a box of strawberries). Each card includes a price (e.g., £0.27 for apples), a "Add to Cart" button, and a color selection dropdown set to "Red". To the right, a sidebar for the logged-in user "Ken" shows options for "Update Account Details", "Add Payment Details", and "Logout". A "Cart Summary" section states "Your shopping cart is empty!".

Figure 41

Test successful

Customer successfully added to database

The screenshot shows the "STORE ADMINISTRATION" section of the eMart website. At the top, there's a logo for "eMART", a shopping bag icon, and links for "Kisacasa eCommerce Solution", "Support", "LOGOUT", and "STORE ADMINISTRATION". Below this is a navigation bar with links for "STOREFRONT", "CATALOG", "STOCK CONTROL", "PRODUCTS ATTRIBUTES", "CARTS", "ORDERS", "CUSTOMERS", "SUPPLIERS", and "SHIPPING". The main content area is titled "eMart Customers" and displays a table of registered users:

First Name	Surname	Email	Tools
customer	Four	emartcustomer4@localhost	<a href="#">View Details</a>
Ken	Williams	emartcustomer5@localhost	<a href="#">View Details</a>

Below the table is a form for "Add new customer:" with fields for First Name, Surname, Email, Password, Telephone number, Street address, Town, City, County, Postcode, and a "Add" button.

Figure 42

### Test Case 3- Customer not registered

Navigate to right hand pane>>Register>>Enter details>>Confirm

The screenshot shows the registration process for a new customer. On the left, a sidebar lists departments: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. The main area is titled "Please enter your details:" and contains the following form fields:

Email Address:	emartcustomer6@localhost
First name:	Kirsty
Surname:	Davies
Password:	.....
Re-enter Password:	.....
Street Address:	15 Leopold Road
Town:	Liverpool
City:	Liverpool
County:	Merseyside
Post Code:	L1 9HX
Telephone:	01517685094

At the bottom of the registration form are "Confirm" and "Cancel" buttons.

To the right, a "Login" section is visible with fields for "E-mail address" and "Password", and buttons for "Login | Register" and "Forgotten Password?". Below it is a "Cart Summary" section stating "Your shopping cart is empty!".

Figure 43

Passwords do not match>>user prompted to re-enter passwords

The screenshot shows the same registration process as Figure 43, but with a password mismatch. The "Re-enter Password" field is empty, and a red error message "You must re-enter the same password." appears below the "Re-enter Password" input field.

The rest of the registration form and the right-hand "Login" and "Cart Summary" sections are identical to Figure 43.

Figure 44

Test successful

#### Test Case 4- Customer updates own information

Navigate to right hand pane>>Enter login credentials>>Click login

**Login**

E-mail address:  
emartcustomer5@loc

Password:  
\*\*\*\*\*

[Login](#) | [Register](#)

[Forgotten Password?](#)

**Cart Summary**

Your shopping cart

Figure 45

Navigate to right hand pane>>Click ‘Update Account Details’

**Welcome, Ken**

[Update Account Details](#)

[Add Payment Details](#)

[Logout](#)

**Cart Summary**

Your shopping cart  
is empty!

Figure 46

Edit details>>Click ‘Confirm’ OR click ‘Cancel’ to discard changes

**Welcome, Ken**

[Update Account Details](#)

[Add Payment Details](#)

[Logout](#)

**Cart Summary**

Your shopping cart  
is empty!

Figure 47

Customer will then be redirected to the eMart home page

Test successful

*Test Case 5- Employee updates customer details on behalf of customer*

Login>>Customers tabs>>Click ‘View Details’

The screenshot shows the 'eMart Customers' page. It displays a table with columns: First Name, Surname, Email, and Tools. Two rows are present: one for 'customer' (First Name: customer, Surname: Four, Email: emartcustomer4@localhost) and one for 'Ken' (First Name: Ken, Surname: Williams, Email: emartcustomer5@localhost). The 'Tools' column contains a 'View Details' button for each row, with the one for 'Ken' highlighted by a red box.

First Name	Surname	Email	Tools
customer	Four	emartcustomer4@localhost	<a href="#">View Details</a>
Ken	Williams	emartcustomer5@localhost	<a href="#">View Details</a>

**Add new customer:**  
 [first name]    [surname]    [email address]    [password]    [telephone number]  
 [street address]    [town]    [city]    [county]    [postcode]   [Add](#)

Figure 48

Update details>>Click ‘Update info’

The screenshot shows the 'Editing customer: ID #2 [back to customers ... ]' page. It lists the following fields and their values:

- First name:** Ken
- Surname:** Williams
- Email address:** emartcustomer5@localhost
- Password:** e7bb0d8ed793905ca536b9960c
- Telephone number:** 01518473679
- Shipping Address:**
  - Street:** 43 Hall Lane
  - Town:** Liverpool
  - City:** Liverpool
  - County:** Merseyside
  - Postcode:** L7 8TF

At the bottom, there are two buttons: [Update info](#) and [Remove customer from database](#).

Figure 49  
Test successful

**Test Case 6- Delete customer**

Login>>Customers tabs>>Click ‘View Details’

The screenshot shows the 'eMart Customers' page. It features a header with the eMart logo, a shopping bag icon, 'Kisacasa eCommerce Solution', 'Support', and 'STORE ADMINISTRATION'. Below the header is a navigation bar with 'STOREFRONT' and 'LOGOUT' on the left, and a menu with 'CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |'. The main content area displays a table of customers:

First Name	Surname	Email	Tools
customer	Four	emartcustomer4@localhost	<a href="#">View Details</a>
Ken	Williams	emartcustomer5@localhost	<a href="#">View Details</a>

Below the table is a form for adding new customers with fields for first name, surname, email address, password, telephone number, street address, town, city, county, and postcode, along with an 'Add' button.

Figure 50

Click ‘Remove customer from database’>>Customer will be permanently deleted

The screenshot shows the 'Editing customer: ID #2 [back to customers ... ]' page. It features a header with the eMart logo, a shopping bag icon, 'Kisacasa eCommerce Solution', 'Support', and 'STORE ADMINISTRATION'. Below the header is a navigation bar with 'STOREFRONT' and 'LOGOUT' on the left, and a menu with 'CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |'. The main content area displays customer details:

First name:	Ken
Surname:	Williams
Email address:	emartcustomer5@localhost
Password:	e7bb0d8ed793905ca536b9960c
Telephone number:	01518473679
Shipping Address:	Street: 43 Hall Lane Town: Liverpool City: Liverpool County: Merseyside Postcode: L7 8TF

At the bottom of the page are buttons for 'Update info' and 'Remove customer from database'.

Figure 51

Test successful

### Test Case 7 – Browse through products

On the homepage, the customer can chose to view a product displayed, chose a department and chose a category>>Choose a product



Figure 52

Customers can either click to return to category or department



Figure 53  
Test successful

### Test Case 8- Add to cart

Browse products(see UC20)>>Click ‘Add to cart’

The screenshot shows a shopping cart interface. On the left, there's a sidebar with 'Choose a Department' and 'Choose a Category' sections. The main area displays a product titled 'Asparagus Tips' with a price of £1.50. An 'Add to Cart' button is visible. To the right, there's a 'Login' form and a 'Cart Summary' section. The 'Cart Summary' section contains a single item: '1 xAsparagus Tips' at '£1.50 [ View details ]'. A red box highlights this summary section.

Figure 54

Product now shows up in left hand pane (running total of cart)

Test successful

### Test Case 9- View cart

Navigate to right hand pane>>click view details

The screenshot shows the same shopping cart interface as Figure 54. The 'Cart Summary' section is highlighted with a red box, and the '£1.50 [ View details ]' link is also highlighted with a red box.

Figure 55

Customer can now view items in cart

The screenshot shows the shopping cart interface with the cart contents table highlighted. The table has columns for Product Name, Price, Quantity, Subtotal, and Tools. One row shows 'Asparagus Tips' with a quantity of 1 and a subtotal of £1.50. The 'Tools' column contains 'Save for later' and 'Remove' links. The 'Update' button is also visible. The 'Cart Summary' section is partially visible on the right.

Figure 56

Test successful

### Test Case 10- Save Cart

Navigate to 'View Cart' (see Test Case 9)

Click 'Save for later'

The screenshot shows a shopping cart interface. On the left, there's a sidebar with a Twitter icon and the text "Testing Testing One Two One Two :D about 18 hours ago". Below this is a "Choose a Department" dropdown menu with options like Fresh Food, Plants, Baked Goods, etc. The main area displays a table titled "These are the products in your shopping cart:" with one item: Asparagus Tips at £1.50. To the right of the table is a "Tools" column containing "Save for later" and "Remove" buttons, with "Save for later" being highlighted with a red box. On the far right, there's a "Login" section with fields for E-mail address and Password, and links for Login | Register and Forgotten Password?

Figure 57

Items have been saved for later

This screenshot shows the same shopping cart interface after saving an item. The main message is "Your shopping cart is empty!". Below this, under "Saved products to buy later:", there is a table with one item: Asparagus Tips at £1.50. The "Tools" column includes "Move to cart" and "Remove". The rest of the interface is identical to Figure 57.

Figure 58

View number of carts in administrative side. Login>>Carts>>Count Old Shopping Carts

The screenshot shows the "eMart" store administration interface. At the top, there's a logo, a shopping bag icon, and links for "Kisacasa eCommerce Solution" and "Support". Below this is a "STORE ADMINISTRATION" header. A navigation bar includes "STOREFRONT" and "LOGOUT" on the left, and "CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |" on the right. The main content area is titled "eMart Shopping Carts:" and displays the message "There are 1 old shopping carts (selected option: All shopping carts.)". It includes a dropdown menu "Select carts: All shopping carts" and buttons for "Count Old Shopping Carts" and "Delete Old Shopping Carts". At the bottom, there's a copyright notice: "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 59

Shows that 1 cart has been saved

Test successful

### Test Case 11- Update quantity

Navigate to view cart (Test Case9)>>Click on quantity textbox and edit number

The screenshot shows a shopping cart interface. On the left, a sidebar lists categories: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. In the center, a table displays the products in the cart:

Product Name	Price	Quantity	Subtotal	Tools
Asparagus Tips	£1.50	5	£1.50	<a href="#">Save for later</a> <a href="#">Remove</a>

Total amount: £1.50 [Update](#) [Continue Shopping](#)

On the right, there is a login form with fields for E-mail address and Password, and buttons for Login, Register, and Forgotten Password?

Figure 60

Click 'Update'

The screenshot shows the same shopping cart interface after the 'Update' button was clicked. The quantity for 'Asparagus Tips' is now 5, and the subtotal is £7.50. The 'Subtotal' cell is highlighted with a red box.

Product Name	Price	Quantity	Subtotal	Tools
Asparagus Tips	£1.50	5	£7.50	<a href="#">Save for later</a> <a href="#">Remove</a>

Total amount: £7.50 [Update](#) [Continue Shopping](#)

On the right, there is a login form with fields for E-mail address and Password, and buttons for Login, Register, and Forgotten Password?

Figure 61

Quantity, subtotal and total amount have now updates

Test successful

### Test Case 12- Remove for cart

Navigate to view cart (TC 9)>>Click 'Remove'

The screenshot shows a shopping cart interface. On the left, a sidebar lists categories: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. The main area displays a table titled "These are the products in your shopping cart:". The table has columns: Product Name, Price, Quantity, Subtotal, and Tools. A single row shows "Asparagus Tips" with price £1.50, quantity 1, subtotal £1.50, and tools "Save for later" and "Remove". The "Remove" button is highlighted with a red box. Below the table, it says "Total amount: £1.50" and "Continue Shopping". At the bottom right are "Update" and "Login | Register" buttons.

Figure 62

Item has now been successfully removed

The screenshot shows a shopping cart interface. On the left, a sidebar lists categories: Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. The main area displays a message "Your shopping cart is empty!" and a "Continue Shopping" link. At the bottom right are "Login | Register" and "Forgotten Password?" buttons.

Figure 63

Customer can now continue shopping

Test successful

### Test Case 13- Add payment details

Navigate to view cart (TC 9)>> Login (TC 21)>>Click ‘Add Payment Details’

**Welcome, Ken**

Add Payment Details  
Logout

These are the products in your shopping cart:				
Product Name	Price	Quantity	Subtotal	Tools
Green Pepper	£0.50	1	£0.50	Save for later Remove
Cheese	£1.99	1	£1.99	Save for later Remove

Total amount: £2.49

Update Checkout Continue Shopping

Figure 64

Enter card details

**Welcome, Ken**

Update Account Details  
Add Payment Details  
Logout

**Cart Summary**

1 x Green Pepper  
1 x Cheese  
£2.49 [ View details ]

Please enter your credit card details:	
Card Holder:	Mr Ken Williams
Card Number (digits only):	4879805948373646
Expiry Date (MM/YY):	10/13
Issue Date (MM/YY if applicable):	11/10
Issue Number (if applicable):	
Card Type:	Visa

Confirm | Cancel

Figure 65

Click ‘Confirm’

**Welcome, Ken**

Change Payment Details  
Logout

These are the products in your shopping cart:				
Product Name	Price	Quantity	Subtotal	Tools
Green Pepper	£0.50	1	£0.50	Save for later Remove
Cheese	£1.99	1	£1.99	Save for later Remove

Total amount: £2.49

Update Checkout Continue Shopping

Figure 66

Customer is redirected to cart, right hand pane no reads ‘Change Payment Details’

Test successful

**Test Case 14- unsuccessful process of order**

Navigate to cart (TC 9)>> Login(TC 21)>> Add payment details(13)>>Click ‘Checkout’

The screenshot shows a shopping cart summary for two items: Green Pepper and Cheese. The total amount is £2.49. Payment details show a Visa card with number XXXX-XXXX-XXXX-3646. Shipping address is listed as 43 Hall Lane, Liverpool, Merseyside, L7 8TF. Shipping type is set to 'Next Day Delivery (£8.75)'. Navigation links include Place Order, Edit Shopping Cart, and Continue Shopping.

Product Name	Price	Quantity	Subtotal
Green Pepper	0.50	1	0.50
Cheese	1.99	1	1.99

Total amount: £2.49

Credit card to use: Visa  
Card number: XXXX-XXXX-XXXX-3646

Shipping address:  
43 Hall Lane  
Liverpool  
Liverpool  
Merseyside  
L7 8TF

Shipping type: Next Day Delivery (£8.75)

Place Order | Edit Shopping Cart | Continue Shopping

Figure 67

Select shipping type from dropdown box

The screenshot shows a shopping cart summary for two items: Green Pepper and Cheese. The total amount is £2.49. Payment details show a Visa card with number XXXX-XXXX-XXXX-3646. Shipping address is listed as 43 Hall Lane, Liverpool, Merseyside, L7 8TF. A dropdown menu for 'Shipping type' is open, showing three options: 'Next Day Delivery (£8.75)', '3-4 Days (£4.50)', and '7 Days (£3)'. The first option is highlighted with a red box. Navigation links include Place Order, Edit Shopping Cart, and Continue Shopping.

Product Name	Price	Quantity	Subtotal
Green Pepper	0.50	1	0.50
Cheese	1.99	1	1.99

Total amount: £2.49

Credit card to use: Visa  
Card number: XXXX-XXXX-XXXX-3646

Shipping address:  
43 Hall Lane  
Liverpool  
Liverpool  
Merseyside  
L7 8TF

Shipping type:

- Next Day Delivery (£8.75)
- 3-4 Days (£4.50)
- 7 Days (£3)

Place Order | Edit Shopping Cart | Continue Shopping

Figure 68

Click ‘Place Order’



Figure 69

Order was unsuccessful due to incorrect card details-Test successful

Email is sent to emartadmin@localhost when errors occur

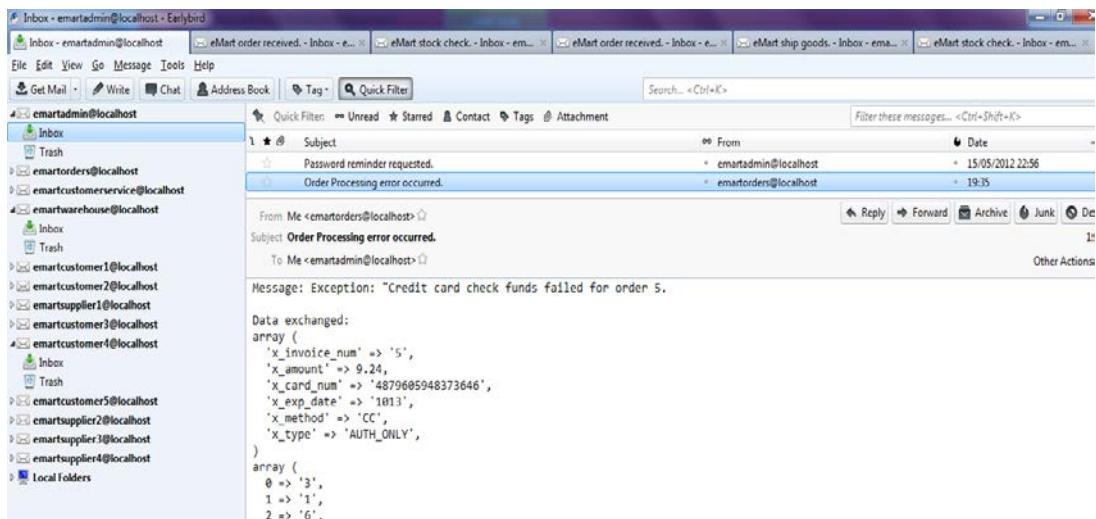


Figure 70

Test successful

To correct payment details click ‘Change payment details’ on right hand pane



Figure 71

Edit details and click ‘Confirm’

### Test Case 15- Successful processing of order



Figure 72

Test successful

Email sent to customer

File Edit View Go Message Tools Help  
Get Mail Write Chat Address Book Tag Quick Filter Search... <Ctrl+K>

From: Me <emartcustomerservice@localhost>☆  
Subject: eMart order received.  
To: Me <emartcustomer5@localhost>☆

Thank you for your order! The products you have ordered are as follows:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Cheese() £1.99 each, total cost £1.99  
Shipping: Next Day Delivery (£6.75)

Total order cost: £9.24

Your order will be shipped to:

Ken Williams  
43 Hall Lane  
Liverpool  
Liverpool  
Merseyside  
L7 8TF

Order reference number: 6

You will receive a confirmation e-mail when this order has been dispatched. Thank you for shopping at eMart!

Figure 73

Email sent to warehouse

File Edit View Go Message Tools Help  
Get Mail Write Chat Address Book Tag Quick Filter

From: Me <emartorders@localhost>☆  
Subject: eMart stock check.  
To: Me <emartwarehouse@localhost>☆

The following products have been ordered:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Cheese() £1.99 each, total cost £1.99  
Shipping: Next Day Delivery (£6.75)

Total order cost: £9.24

Please check availability and confirm via <http://localhost/emart/admin.php>

Order reference number: 6

Figure 74

Warehouse must now confirm that order has relevant stock

- Warehouse has stock

Login (TC21)>>Orders tab>>Using order reference(as stated in email) filter search>>Click ‘Go!’

The screenshot shows the eMart Orders page. At the top, there are several search filters:

- Show orders by customer: Ken Williams, Go!
- Get by order ID: 6, Go!
- Show the most recent 20 orders, Go!
- Show all records created between [start date] and [end date], Go!
- Show orders by status: Received, Go!

The main area displays a table of orders:

Order ID	Date Created	Date Shipped	Status	Customer	Tools
6	2012-05-16 19:43:05		Awaiting Stock Confirmation	Ken Williams	<a href="#">View Details</a>

At the bottom, there is copyright information: © Copyright eMart 2012, CREATED BY KISACASA.

Figure 75

Click ‘View details’

The screenshot shows the eMart Orders page with the 'View Details' link in the 'Tools' column of the first row of the table highlighted with a red box.

The table data is identical to Figure 75:

Order ID	Date Created	Date Shipped	Status	Customer	Tools
6	2012-05-16 19:43:05		Awaiting Stock Confirmation	Ken Williams	<a href="#">View Details</a>

At the bottom, there is copyright information: © Copyright eMart 2012, CREATED BY KISACASA.

Figure 76

To confirm the warehouse has the necessary stock click ‘Confirm Stock for Order’

Editing details for order ID: 6 [ back to admin orders... ]

Total Amount:	£2.49
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-16 19:43:05
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	PYFTXW
Reference Number:	2171932588
Comments:	
Customer Name:	Ken Williams
Shipping Address:	Street: 43 Hall Lane Town: Liverpool City: Liverpool County: Merseyside Postcode: L7 8TF
Customer Email:	emartcustomer5@localhost
Telephone Number:	01518473679

[Edit](#) [Update](#) [Cancel](#) [Confirm Stock for Order](#)

**Order contains these products:**

Product ID	Product Name	Quantity	Unit Cost	Subtotal
6	Green Pepper	1	£0.50	£0.50
15	Cheese	1	£1.99	£1.99

Figure 77

The warehouse will now receive an email to confirm when the order has been dispatched

From Me <emartorders@localhost>★  
Subject eMart ship goods.  
To Me <emartwarehouse@localhost>★

Payment has been received for the following goods:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Cheese() £1.99 each, total cost £1.99  
Shipping: Next Day Delivery (£6.75)

Total order cost: £9.24

Please ship to:

Ken Williams  
43 Hall Lane  
Liverpool  
Liverpool  
Merseyside  
L7 8TF

When goods have been shipped, please confirm via <http://localhost/emart/admin.php>

Order reference number: 6

Figure 78

Login>>Customers>>Filter by order reference number(as above)

Editing details for order ID: 6 [ back to admin orders... ]

Total Amount:	£2.49
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-16 19:43:05
Date Shipped:	2012-05-16 20:35:07
Status:	Awaiting Despatch Confirmation
Authorization Code:	PYFTXW
Reference Number:	2171932588
Comments:	
Customer Name:	Ken Williams
Shipping Address:	Street: 43 Hall Lane Town: Liverpool City: Liverpool County: Merseyside Postcode: L7 8TF
Customer Email:	emartcustomer5@localhost
Telephone Number:	01518473679

[Edit](#) [Update](#) [Cancel](#) [Confirm Shipment for Order](#)

Figure 79

Customer will receive email to confirm dispatch of goods

File Edit View Go Message Tools Help

Get Mail Write Chat Address Book Tag Quick Filter

From Me <emartcustomerservice@localhost> ★  
Subject eMart order dispatched.  
To Me <emartcustomer5@localhost> ★

Your order has now been dispatched! The following products have been shipped:

1 Green Pepper() £0.50 each, total cost £0.50  
1 Cheese() £1.99 each, total cost £1.99  
Shipping: Next Day Delivery (£6.75)

Total order cost: £9.24

Your order has been shipped to:

Ken Williams  
43 Hall Lane  
Liverpool  
Liverpool  
Merseyside  
L7 8TF

Order reference number: 6

Thank you for shopping at eMart!

Figure 80

- b) Unsuccessful process. E.g. Stock not available  
Warehouse will select cancelled

**Editing details for order ID: 6 [ back to admin orders... ]**

<b>Total Amount:</b>	£2.49
<b>Shipping:</b>	Next Day Delivery (£6.75)
<b>Date Created:</b>	2012-05-16 19:43:05
<b>Date Shipped:</b>	
<b>Status:</b>	Cancelled
<b>Authorization Code:</b>	PYFTXW
<b>Reference Number:</b>	2171932588
<b>Comments:</b>	
<b>Customer Name:</b>	Ken Williams
<b>Shipping Address:</b>	Street: 43 Hall Lane Town: Liverpool City: Liverpool County: Merseyside Postcode: L7 8TF
<b>Customer Email:</b>	emartcustomer5@localhost
<b>Telephone Number:</b>	01518473679

**Edit** **Update** **Cancel**

Figure 81  
Test successful

**Test Case 16- Filter orders by different search criteria**

Navigate to Orders. Login (TC24)>> Navigate to Orders tab

The screenshot shows the eMart Orders page. At the top, there is a navigation bar with links for STOREFRONT, LOGOUT, CATALOG, STOCK CONTROL, PRODUCTS ATTRIBUTES, CARTS, ORDERS, CUSTOMERS, SUPPLIERS, and SHIPPING. Below the navigation bar, there is a search section with the following filters:

- Show orders by customer: customer Four, Go!
- Get by order ID: [input field], Go!
- Show the most recent 20 orders, Go!
- Show all records created between [input field] and [input field], Go!
- Show orders by status: Received, Go!

Below the filters, there is a table titled "eMart Orders" displaying two rows of order data:

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

At the bottom of the page, there is a copyright notice: © Copyright eMart 2012 CREATED BY KISACASA.

Figure 82

Filter by customer

The screenshot shows the eMart Orders page with the "Show orders by customer" filter set to "customer Four". The table below displays the same two orders as Figure 82, but with a yellow highlight applied to the entire row for Order ID 3.

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 83

Filter by order ID

The screenshot shows the eMart Orders page with the "Get by order ID" filter set to "1". The table below displays the same two orders as Figure 82, but with a yellow highlight applied to the entire row for Order ID 1.

Order ID	Date Created	Date Shipped	Status	Customer	Tools
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>

Figure 84

Filter by most recent orders

eMart Orders

Show orders by customer customer Four Go!

Get by order ID Go!

Show the most recent 20 orders Go!

Show all records created between and Go!

Show orders by status Received Go!

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 85

Filter by date

- a) Correct date/time order entered YYYY/MM/DD HH:MM:SS

eMart Orders

Show orders by customer customer Four Go!

Get by order ID Go!

Show the most recent 20 orders Go!

Show all records created between 2012/05/15 01:00:00 and 2012/05/17 01:00:00 Go!

Show orders by status Received Go!

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 86

Test successful

Filter by date

- b) Incorrect date/time format

STOREFRONT LOGOUT

CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |

eMart Orders

No orders found matching your searching criteria!

Show orders by customer customer Four Go!

Get by order ID Go!

Show the most recent 20 orders Go!

Show all records created between 1970/01/01 01:00:00 and 1970/01/01 01:00:00 Go!

Show orders by status Received Go!

Figure 87

Filter rejects search and sets date and time to default

Filters all work>>Test successful

### Test Case 17- Edit order details

Navigate to orders. Login(TC24)>> Locate order which needs updating (TC16)>>Click ‘Edit’

Editing details for order ID: 4 [ back to admin orders... ]

Total Amount:	£5.09
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-15 23:51:12
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	SYP1L3
Reference Number:	2171902975
Comments:	
Customer Name:	customer Four
Shipping Address:	Street: 113 hall lane Town: Liverpool City: Liverpool County: merseyside Postcode: L7 8tf
Customer Email:	emartcustomer4@localhost
Telephone Number:	447768033426

**Edit** **Update** **Cancel** **Confirm Stock for Order**

Figure 88

Edit text>>Click ‘Update’  
Click ‘Cancel’ to discard changes

Editing details for order ID: 4 [ back to admin orders... ]

Total Amount:	£5.09
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-15 23:51:12
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	SYP1L3
Reference Number:	2171902975
Comments:	
Customer Name:	customer Four
Shipping Address:	Street: 50 hall lane Town: Liverpool City: Liverpool County: merseyside Postcode: L7 8tf
Customer Email:	emartcustomer4@localhost
Telephone Number:	447768033426

**Edit** **Update** **Cancel** **Confirm Stock for Order**

Figure 89

Details successfully updated

**Editing details for order ID: 4 [ back to admin orders... ]**

Total Amount:	£5.09
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-15 23:51:12
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	SYP1L3
Reference Number:	2171902975
Comments:	
Customer Name:	customer Four
Shipping Address:	Street: 60 hall lane Town: Liverpool City: Liverpool County: merseyside Postcode: L7 8lf
Customer Email:	emartcustomer4@localhost
Telephone Number:	447768033426

**Buttons:** Edit | Update | Cancel | Confirm Stock for Order

Figure 90

Test successful

### Test Case 18

Navigate to suppliers. Login (24)>>Click Supplier tab>>Enter information

The screenshot shows the 'eMart Suppliers' page. At the top, there is a table with one row:

Name	Email	Tools
Green & Co	GreenCo@localhost.com	<a href="#">View Details</a>

Below this is a form titled 'Add new supplier:' with five input fields:

Merseyside Dairy	MerseyDairy@localhost	01519685738	Milk Field	New Brighton
Liverpool	Merseyside	L25 8HG	Add	

Figure 91

Click 'Add'

a) Successful addition

The screenshot shows the 'eMart Suppliers' page after adding a new supplier. The table now has two rows:

Name	Email	Tools
Green & Co	GreenCo@localhost.com	<a href="#">View Details</a>
Merseyside Dairy	MerseyDairy@localhost.com	<a href="#">View Details</a>

The 'Add new supplier:' form is still visible at the bottom.

Figure 92

Test successful

b) Unsuccessful addition

The screenshot shows the 'eMart Suppliers' page. At the top, there is a message: 'Please correct the email address'. Below this is a table with two rows:

Name	Email	Tools
Green & Co	GreenCo@localhost.com	<a href="#">View Details</a>
Merseyside Dairy	MerseyDairy@localhost.com	<a href="#">View Details</a>

Below the table is a form for adding a new supplier:

Add new supplier:

At the bottom of the page, there is a copyright notice: © Copyright eMart 2012 CREATED BY KISACASA.



Figure 93

Edit Supplier. Navigate to Suppliers>>click view details>>Edit text

The screenshot shows the 'Editing supplier: ID #1 [back to suppliers ... ]' page. The supplier details are as follows:

Company Name:	Green & Co
Email address:	GreenCo@localhost.com
Telephone number:	01515543234
Company Address:	Street: 89 hall lane Town: Huyton City: liverpool County: Merseyside Postcode: L30 8HG

At the bottom of the page are three buttons: 'View Deliveries' (green), 'Update info' (red), and 'Remove supplier from database' (red).

At the very bottom, there is a copyright notice: © Copyright eMart 2012 CREATED BY KISACASA.

Figure 94

Click 'Update info'

Delete Supplier. Navigate as above to supplier needed>> Click ‘Remove supplier from database’

The screenshot shows the eMart Store Administration interface. At the top, there's a logo with the word 'eMART' in green, a navigation bar with 'Kisacasa eCommerce Solution' and 'Support', and a main title 'STORE ADMINISTRATION'. Below this is a sub-navigation bar with 'STOREFRONT' on the left and 'LOGOUT' on the right. A horizontal menu bar includes links for CATALOG, STOCK CONTROL, PRODUCTS ATTRIBUTES, CARTS, ORDERS, CUSTOMERS, SUPPLIERS, and SHIPPING. The main content area is titled 'Editing supplier: ID #1 [back to suppliers ...]'. It contains a form with fields for Company Name (Green & Co), Email address (GreenCo@localhost.com), Telephone number (01515543234), Company Address (Street: 89 hall lane, Town: Huyton, City: liverpool, County: Merseyside, Postcode: L30 8HG). Below the form are buttons for 'View Deliveries', 'Update info', and 'Remove supplier from database' (which is highlighted with a red border). At the bottom of the page, there's a copyright notice: '© Copyright eMart 2012' and 'CREATED BY KISACASA'.

Figure 95  
Test successful

PLEASE NOTE: Tests were also carried out to add, edit and delete the following eMart features:

- Supplier deliveries
- Product attributes
- Shipping costs

Results were similar to those of test case 18

**Test Case 19- Add department**

Navigate to home page>>Scroll to bottom of page>>Enter information

<b>Add new department:</b>				
Outdoors	0	01510000011	Here you will find our top quality outdoor products	<input type="button" value="Add"/>

Figure 96

Department added

eMart Departments				
Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Chilled & Frozen Foods	0	01510000009	Here you will find our chilled and frozen food selection.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Pets	0	01510000010	Spoil your adored pets with our large selection of luxurious treats and toys.	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
Outdoors	0	01510000011	Here you will find our top quality outdoor products	<input type="button" value="Edit Categories"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 97

Test successful

**Test Case 20- Delete and edit department**

Edit Department. Navigate to catalog>> Click ‘Edit’

eMart Departments				
Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Update</a> <a href="#">Cancel</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

Figure 98

Click update to save changes

Click cancel to discard changes

eMart Departments				
Department Name	Manager	Tel Number	Description	Tools
Fresh Goods	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

Figure 99

Delete department. Navigate to catalog>>Click delete on department you wish to discard

Outdoors	0	01510000011	Here you will find our top quality outdoor products	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
----------	---	-------------	---	---

Figure 100

Department deleted

eMart Departments				
Department Name	Manager	Tel Number	Description	Tools
Fresh Goods	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Chilled & Frozen Foods	0	01510000009	Here you will find our chilled and frozen food selection.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Pets	0	01510000010	Spoil your adored pets with our large selection of luxurious treats and toys.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

Add new department:  
 [name]  [manager]  [telephoneNumber]  [description]

Figure 101  
Test successful

### Test Case 21- Successful Customer Login

Navigate to right hand pane>>Enter credentials

**Choose a Department**

- Fresh Goods
- Plants
- Baked Goods
- Food Cupboard
- Beverages
- Health & Beauty
- Household
- Special Diet
- Chilled & Frozen Foods
- Pets

**Choose a Category**

- Fresh Fruit
- Vegetables
- Salad
- Dairy Produce

**Fresh Goods**

Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.

[1](#) [2](#) [Next page](#)

<b>Golden Delicious Apples</b>	<b>Asparagus Tips</b>
	
Loose Golden Delicious Apples Price: £0.27	Class 1 Asparagus Tips Old price: £2.00 Price: £1.50
<a href="#">Add to Cart</a>	<a href="#">Add to Cart</a>

Colour: Red

<b>Cheese</b>	<b>Green Pepper</b>
	
Mature Cheddar Cheese Old price: £2.50 Price: £1.99	Loose Green Pepper Old price: £0.78 Price: £0.50
<a href="#">Add to Cart</a>	<a href="#">Add to Cart</a>

**Login**

E-mail address: emartcustomer5@lo...  
Password: .....  
[Login](#) | [Register](#)  
[Forgotten Password?](#)

**Cart Summary**

Your shopping cart is empty!

Figure 102

Click Login

Welcome, Ken

Update Account Details  
Change Payment Details  
Logout

Cart Summary

Your shopping cart is empty!

Figure 103

Login successful>>Test successful

**Test Case 22- Unsuccessful Customer Login**

Navigate to right hand pane>>Enter credentials

E-mail address:  
emartcustomer5@lo  
Password:  
.....

Login | Register  
Forgotten Password?

Cart Summary

Your shopping cart is empty!

Figure 104

Click Login

Unrecognised password.

E-mail address:  
emartcustomer5@lo  
Password:  
.....

Login | Register  
Forgotten Password?

Cart Summary

Your shopping cart is empty!

Figure 105

Login failed>>Test successful

### Test Case 23- Customer password reminder

Navigate to right hand pane>>Enter email address>>Click ‘Forgotten Password’

The screenshot shows the grocery store's website interface. On the left, there's a sidebar for 'Choose a Department' with options like Fresh Goods, Plants, Baked Goods, etc. The main content area is titled 'Fresh Goods' and features sections for 'Golden Delicious Apples' and 'Asparagus Tips'. Each product has a small image, a brief description, the original price, the current price, and an 'Add to Cart' button. A dropdown menu for 'Colour' is set to 'Red'. Below these are sections for 'Cheese' and 'Green Pepper'. On the right side, there's a 'Login' form with fields for 'E-mail address' (containing 'emartcustomer5@ioi'), 'Password' (containing '.....'), and buttons for 'Login', 'Register', and 'Forgotten Password?'. A red box highlights the 'E-mail address' field. Below the login form is a 'Cart Summary' section stating 'Your shopping cart is empty!'. At the bottom of the page, there's a footer with a copyright notice.

Figure 106

Message: “Reminder email sent”

This screenshot is identical to Figure 106, showing the grocery store's website. The left sidebar, main product sections, and right-hand login form are all present. However, the 'Reminder email sent.' message is displayed above the 'E-mail address:' field in the login form, indicating that a password reset email has been sent to the provided address. The rest of the page, including the cart summary, footer, and footer links, remains the same.

Figure 107

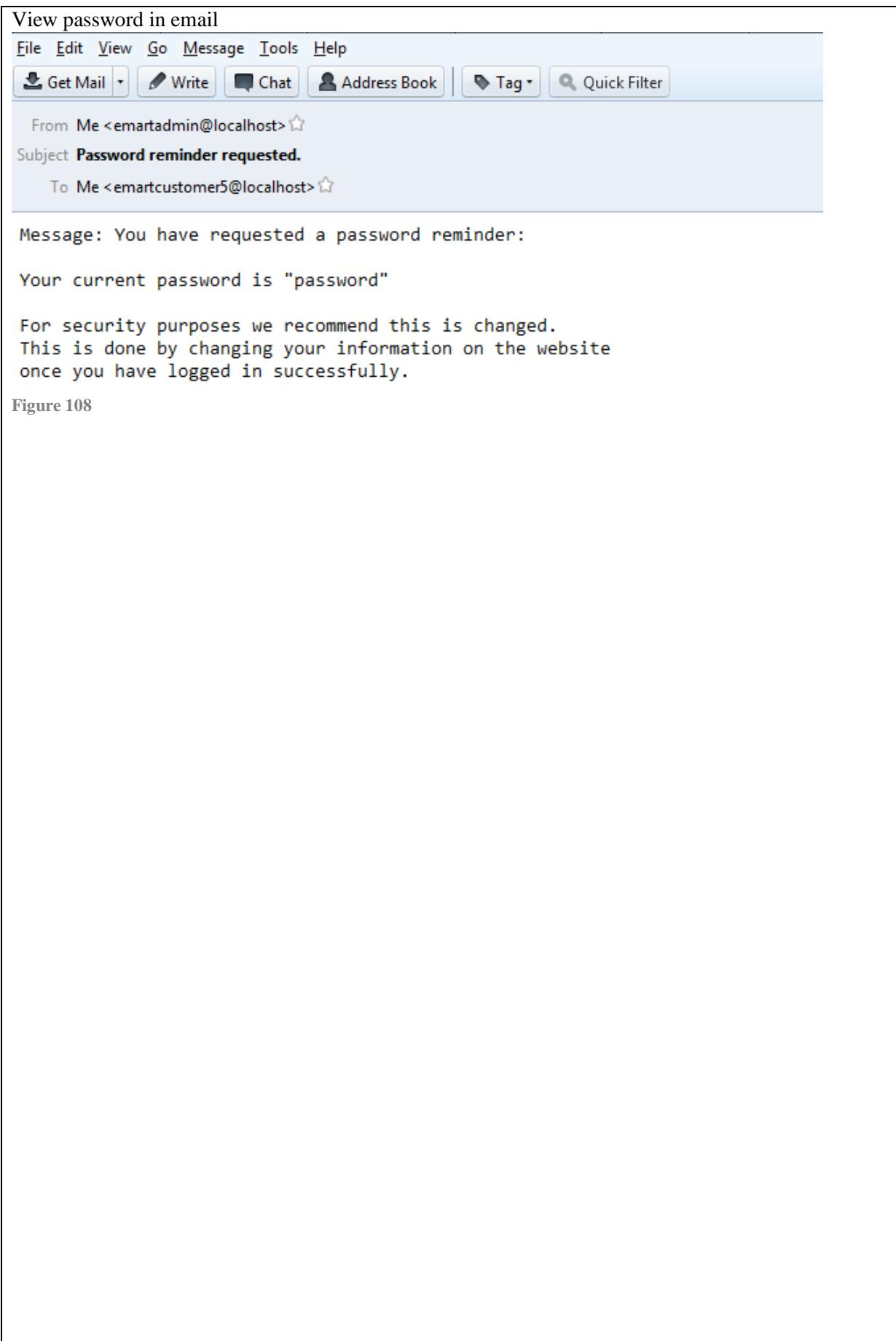


Figure 108

**Test Case 24- Successful staff login**

Enter login credentials

Figure 109

Click 'Login'

Department Name	Manager	Tel Number	Description	Tools
Fresh Goods	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Chilled & Frozen Foods	0	01510000009	Here you will find our chilled and frozen food selection.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Pets	0	01510000010	Spoil your adored pets with our large selection of luxurious treats and toys.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Add new department:**  
 [name]  [manager]  [telephoneNumber]  [description]

Figure 110

User successfully logged in- test successful

**Test Case 25- unsuccessful staff login**

Enter login credentials

The screenshot shows the eMart Site Administrator Login page. At the top right, there is a green shopping bag icon, the text "Kisacasa eCommerce Solution", and a "Support" link. Below that is the "STORE ADMINISTRATION" header. The main content area has a dark grey background and displays the "Site Administrator Login" heading. It includes a message "Enter login information or go back to eMart Storefront." Below this are two input fields: "Username: sysadmin" and "Password: .....". Underneath the password field is a "Forgotten Password?" link. To the right of the password field is a yellow "Login" button. At the bottom of the page, there is a footer bar with the text "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 111

Click 'Login'

The screenshot shows the eMart Site Administrator Login page after a successful login attempt. The layout is identical to Figure 111, with the green shopping bag icon, "Kisacasa eCommerce Solution", "Support", "STORE ADMINISTRATION" header, and "Site Administrator Login" heading. The "Username: sysadmin" and "Password: ....." fields are present. A red error message "Login failed. Please try again:" is displayed above the "Login" button. Below this message, the "Forgotten Password?" link is visible. The "Login" button is yellow. The footer bar at the bottom contains the copyright notice "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 112

User successfully logged in- test successful

Test Case Number	Use Case Reference	Other Comments
1	4	
2	16	
3	16	
4	18	
5	18	
6	19	
7	20	
8	22	
9	23	
10	58	NEW UC: Save Cart
11	24	
12	25	
13	59	NEW UC: Add payment details
14	26	
15	26	
16	27	
17	29	
18	33,35,36	
19	49	
20		
21	55	
22	56	
23	57	NEW UC: Password reminder

24	55	
25	56	
26	57	NEW UC: Password reminder

## **Conclusion**

Overall the testing phase has been a success. Few errors were noted however the errors that were, have now hopefully lead to a fully complete, accurate system.

```

<?php
    session_start();
    if (empty ($_GET['step']))
    {
        require_once 'include/config.php';
        require_once BUSINESS_DIR . 'authorize_net_request.php';
        $request = new AuthorizeNetRequest(AUTHORIZE_NET_URL);

        // Auth
        $transaction =
            array ('x_invoice_num' => '99999', // Invoice number
                   'x_amount' => '45.99', // Amount
                   //'x_currency_code' => 'GBP',
                   //Not yet supported with Authorize.net
                   //will be processed at dollars,
                   //this is ok for test purposes)
                   'x_card_num' => '4007000000027', // Credit card number
                   'x_exp_date' => '1213', // Expiration date
                   'x_method' => 'CC', // Payment method
                   'x_type' => 'AUTH_ONLY');// Transaction type

        $request->SetRequest($transaction);
        $auth_only_response = $request->GetResponse();
        $_SESSION['auth_only_response'] = $auth_only_response;
        $auth_only_response = explode('|', $auth_only_response);

        // Read the transaction ID, which will be necessary for taking the payment
        $ref_trans_id = $auth_only_response[6];

        // Capture
        $transaction =
            array ('x_ref_trans_id' => $ref_trans_id, // Transaction id
                   'x_method' => 'CC', // Payment method
                   'x_type' => 'PRIOR_AUTH_CAPTURE');// Transaction type

        $request->SetRequest($transaction);
        $prior_auth_capture_response = $request->GetResponse();
        $_SESSION['prior_auth_capture_response'] = $prior_auth_capture_response;
    }
    else
    {
        switch ($_GET['step'])
        {

            case 1:
                print $_SESSION['auth_only_response'];
                break;
            case 2:
                print $_SESSION['prior_auth_capture_response'];
                break;
        }

        exit();
    }
?>
<frameset cols="50%, 50%">
    <frame src="test_authorize_net.php?step=1">
    <frame src="test_authorize_net.php?step=2">
</frameset>

```

```

<?php
require_once 'include/config.php';

require_once BUSINESS_DIR . 'symmetric_crypt.php';
require_once BUSINESS_DIR . 'secure_card.php';

$card_holder = 'John Doe';
$card_number = '1234567890123456';
$expiry_date = '01/09';
$issue_date = '01/06';
$issue_number = 100;
$card_type = 'Mastercard';

echo '<br />Credit card data:<br />' .
    $card_holder . ',' . $card_number . ',' .
    $issue_date . ',' . $expiry_date . ',' .
    $issue_number . ',' . $card_type . '<br />';

$credit_card = new SecureCard();
try
{
    $credit_card->LoadPlainDataAndEncrypt($card_holder, $card_number,
    $issue_date, $expiry_date, $issue_number, $card_type);
    $encrypted_data = $credit_card->EncryptedData;
}
catch(Exception $e)
{
    echo '<font color="red">Exception: ' . $e->getMessage() . '</font>';
    exit();
}

echo '<br />Encrypted data:<br />' . $encrypted_data . '<br />';

$our_card = new SecureCard();
try
{
    $our_card->LoadEncryptedDataAndDecrypt($encrypted_data);
    echo '<br/>Decrypted data:<br/>' .
        $our_card->CardHolder . ',' . $our_card->CardNumber . ',' .
        $our_card->IssueDate . ',' . $our_card->ExpiryDate . ',' .
        $our_card->IssueNumber . ',' . $our_card->CardType;
}
catch(Exception $e)
{
    echo '<font color="red">Exception: ' . $e->getMessage() . '</font>';
    exit();
}
?>

```

```

<?php
if (isset ($_GET['my_string']))
{
require_once 'include/config.php';
require_once BUSINESS_DIR . 'symmetric_crypt.php';
$string = $_GET['my_string'];
echo 'The string is:<br />' . $string . '<br /><br />';
$encrypted_string = SymmetricCrypt::Encrypt($string);
echo 'Encrypted string:<br />' . $encrypted_string . '<br /><br />';
$decrypted_string = SymmetricCrypt::Decrypt($encrypted_string);
echo 'Decrypted string:<br />' . $decrypted_string;
}
?>
<br /><br />
<form action="test_encryption.php">
Enter string to encrypt:
<input type="text" name="my_string" /><br />
<input type="submit" value="Encrypt" />
</form>

```

## Appendix 4

### Test hasher

```

<?php
if (isset ($_GET['to_be_hashed']))
{
require_once 'include/config.php';
require_once BUSINESS_DIR . 'password_hasher.php';

$original_string = $_GET['to_be_hashed'];

echo 'The hash of "' . $original_string . '" is ' .
    PasswordHasher::Hash($original_string, false);
echo '<br />';
echo '... and the hash of "' . HASH_PREFIX . $original_string .
    '" (secret prefix concatenated with password) is ' .
    PasswordHasher::Hash($original_string, true);
}

?>
<br /><br />
<form action="test_hasher.php">
Write your password:
<input type="text" name="to_be_hashed" /><br />
<input type="submit" value="Hash it" />
</form>

```

# eMart Content Survey

---

Please answer all questions honestly. All answers are anonymous.

## Accuracy

Enter four items into the shopping cart. Is the total price accurate?

yes

no

Register as a new customer and enter your details. Now view your account. Are your personal details exactly as you entered?

yes

no

If no, please detail

.....  
.....  
.....

## Consistency

In your opinion has the layout been consistent in the user interface?

yes

no

If no please explain why not and where

.....  
.....  
.....  
.....  
.....

Have you found any of the website not aesthetically pleasing?

yes

no

.....  
.....  
.....  
.....

## **Spelling**

Can you see any spelling errors on the homepage?

yes

no

If yes please give details

.....  
.....  
.....  
.....  
.....

Please click through the website for three minutes and note any errors if any. Please specify on which page they occur

.....  
.....  
.....  
.....  
.....

## **Accessibility**

Are all aspects of the website accessible on your browser?

yes

no

What is your browser?

.....  
.....

## **Other**

Please state any other errors not listed above

.....  
.....  
.....  
.....

## Appendix E

# USER MANUAL

---

## Installation

This section describes the steps to be carried out to get the eMart system up and running.

## System Requirements

To effectively use the eMart system it is recommended that your system consists of the minimum specification.

<b>Processor</b>	1.6 GHz
<b>Ram</b>	512MB
<b>Hard drive</b>	500MB
<b>Screen Resolution</b>	1024x768
<b>Database</b>	MySQL
<b>Programming Language</b>	PHP5

eMart was developed in PHP5 and as such any server intending to host the application must have it installed with PDO and output buffer enabled. The system was developed with the ability to cross platforms in mind and as such can be run on mostly all operating systems including Microsoft Windows, Mac OS, Linux and UNIX.

Web hosting is necessary. It is recommended that a service is obtained that provides built in MySQL database optimisation and maintenance.

## Installation

### Database

To install the MySQL database it is recommended that phpMyAdmin be utilised. Any major web hosting company will provide this with their service. Import the database script provided within the package. This is located in the data directory at the root of the package. Once imported the configuration file will need to be amended to match the connection information given to you by your hosting company. The user guide describes how to access the configuration guide and make amendments. The file is well documented and clearly states which sections you should alter.

### Application

Following the installation of the database, setting up the application is straightforward. Simply upload the entire contents of the package to the root of your hosted domain. Every hosting provider will provide different means of uploading site content. Although each one generally supplies a Graphic Interface for this to be done.

### Solutions to possible compatibility issues

With many hosting companies it is common for them to not have all the PHP5 features enabled that this application uses. We strongly recommend that you upload the included php5.ini to the root of your domain to overwrite any local installation on the hosting server. A comprehensive error handler has been incorporated into the design of the system that will clearly state the compatibility problem you are facing. Contact your hosting company with the error and they will more than likely provide you with instructions to alter various settings within your hosting account.

## Products

After logging into the eMart store administration, the page will load:

The screenshot shows the 'eMart' logo at the top left, followed by a navigation bar with icons for file, edit, and search, and the text 'Kisacasa eCommerce Solution' and 'Support'. Below this is the 'STORE ADMINISTRATION' header. A sub-navigation bar includes 'STOREFRONT' and 'LOGOUT' on the left, and 'CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS' on the right. The main content area is titled 'eMart Departments' and contains a table listing ten departments with columns for Name, Manager, Tel Number, Description, and Tools (Edit Categories, Edit, Delete). The departments listed are Fresh Food, Plants, Baked Goods, Food Cupboard, Beverages, Health & Beauty, Household, Special Diet, Chilled & Frozen Foods, and Pets. At the bottom of the table is a form for adding a new department with fields for name, manager, telephone number, and description, and an 'Add' button.

Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Plants	0	01510000002	Here you will find fresh herbs and seeds to grow in your very own vegetable patch.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Baked Goods	0	01510000003	Here you will find mouth watering, freshly baked breads and delicious treats.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Food Cupboard	0	01510000004	Here you will find all those bits and bobs that hide away in the cupboard.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Beverages	0	01510000005	Here you will find the very best in organic drinks to quench your thirst.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Health & Beauty	0	01510000006	Here you will find the very best in cosmetics for you and your body.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Household	0	01510000007	Give your home that luxurious clean feeling with our selection of household products.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Special Diet	0	01510000008	Here you will find all those foods for speciality diets like gluten free, wheat free etc.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Chilled & Frozen Foods	0	01510000009	Here you will find our chilled and frozen food selection.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>
Pets	0	01510000010	Spoil your adored pets with our large selection of luxurious treats and toys.	<a href="#">Edit Categories</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Add new department:**  
[name] [manager] [telephoneNumber] [description]

© Copyright eMart 2012  
CREATED BY KISACASA

Figure 1

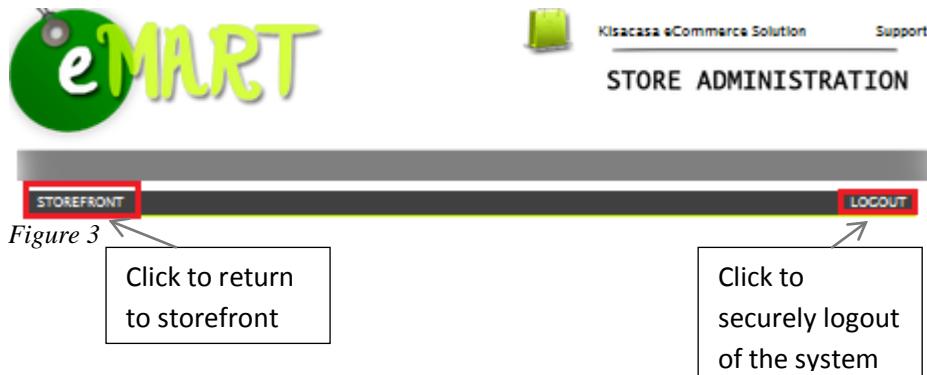
This page displays the features which can be updated and changed on the system

## Add a department:

Add new department:  
 [name]  [manager]  [telephoneNumber]  [description]

Figure 2

Click to in textboxes to add information about a new department and click add to save changes



## Edit a department:

Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and dairy products.	<input type="button" value="Edit Categories"/> <input style="outline: 2px solid red;" type="button" value="Edit"/> <input type="button" value="Delete"/> <input type="button" value="Edit Categories"/> <input type="button" value="Edit"/>

Figure 4

After clicking the following page will load:

Department Name	Manager	Tel Number	Description	Tools
Fresh Food	0	01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and	<input type="button" value="Edit Categories"/> <input style="outline: 2px solid red;" type="button" value="Update"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>

Figure 5

To edit the name, manager, telephone number or description of the department, simply click in the text box and edit. NOTE: Ensure that changes are saved (shown below)

Department Name	Manager	Tel Number	Description	Tools
Fresh Goods		01510000001	Here you will find the very best in locally sourced organic fruit, vegetables and	<input type="button" value="Edit Categories"/> <input style="outline: 2px solid red;" type="button" value="Update"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>

Figure 6

Once edited, the department will be saved/discarded, the department (see fig)

Click to save changes

Click to discard changes

## Delete a department:

Department Name	Manager	Tel Number	Description	Tools
Fresh Goods		0151000001	Here you will find the very best in locally sourced organic fruit, vegetables and	<a href="#">Edit Categories</a> <a href="#">Update</a> <a href="#">Cancel</a> <a href="#">Delete</a>

Figure 7

Click to delete department. NOTE:  
Department must have no products in  
before deleting. Once clicked,  
department is permanently deleted.

## View department categories:

Department Name	Manager	Tel Number	Description	Tools
Fresh Goods		0151000001	Here you will find the very best in locally sourced organic fruit, vegetables and	<a href="#">Edit Categories</a> <a href="#">Update</a> <a href="#">Cancel</a> <a href="#">Delete</a>

Figure 8

Click to edit categories in the department

The following page will then load:

The screenshot shows the eMart STORE ADMINISTRATION interface. At the top, there's a logo, a support link, and a navigation menu with links like STOREFRONT, LOGOUT, CATALOG, STOCK CONTROL, PRODUCTS ATTRIBUTES, CARTS, ORDERS, CUSTOMERS, and SUPPLIERS. Below this is a title bar for 'Editing categories for department: Fresh Goods [back to departments ...]'. The main content area is a table listing categories with their descriptions and tool buttons. At the bottom, there's a form for adding new categories and a copyright notice.

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will always find the highest-quality fresh fruit each and every time you shop with us.	<a href="#">Edit Products</a> <a href="#">Edit</a> <a href="#">Delete</a>
Vegetables		<a href="#">Edit Products</a> <a href="#">Edit</a> <a href="#">Delete</a>
Salad		<a href="#">Edit Products</a> <a href="#">Edit</a> <a href="#">Delete</a>
Dairy Produce		<a href="#">Edit Products</a> <a href="#">Edit</a> <a href="#">Delete</a>
Fresh Meat		<a href="#">Edit Products</a> <a href="#">Edit</a> <a href="#">Delete</a>

Add new category:  
 [name]  [description]

© Copyright eMart 2012  
CREATED BY KISACASA

Figure 9

## Add a category:

Add new category:  
[name] [description] Add

Figure 10

Click to in textboxes to add information about a new category and click add to save changes

Click to return to departments (see Figure 1)

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will always find the highest-quality fresh fruit each and every time you shop with us.	Edit Products Edit Delete

Figure 11

## Delete a category:

Editing categories for department: Fresh Goods [back to departments ... ]

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will always find the highest-quality fresh fruit each and every time you shop with us.	Edit Products Edit Delete

Click to delete the category.  
NOTE: Once clicked, category is permanently deleted. If category is reintroduced then information must be entered manually

Click to edit category

## Edit a category:

Editing categories for department: Fresh Goods [back to departments ... ]

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will always find the highest-quality fresh fruit each and every time you shop with us.	Edit Products Update Cancel Delete

Figure 13

The category can be edited. NOTE: click update to save changes (see below):

Editing categories for department: Fresh Goods [back to departments ... ]

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will find the highest-quality fresh fruit each and every time you shop with us.	<input type="button" value="Edit Products"/> <input type="button" value="Update"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>

Figure 14

The diagram shows the 'Editing categories for department: Fresh Goods' interface. A large callout box labeled 'Edited text' points to the 'Description' field where text is being typed. Another callout box labeled 'Click to save changes' points to the 'Update' button in the 'Tools' column. A third callout box labeled 'Click to discard changes' points to the 'Cancel' button. A fourth callout box labeled 'Click to delete the category. NOTE: Once clicked, category is permanently deleted. If category is reintroduced then information must be entered manually' points to the 'Delete' button.

After saving/discard changes, the category page will load (see figure 10)

### View products:

Editing categories for department: Fresh Goods [back to departments ... ]

Category Name	Description	Tools
Fresh Fruit	From crisp apples to vibrant red strawberries, you will always find the highest-quality fresh fruit each and every time you shop with us.	<input type="button" value="Edit Products"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

Figure 15

The diagram shows the 'Editing categories for department: Fresh Goods' interface after changes have been saved. A callout box labeled 'Click to edit products in the category' points to the 'Edit Products' button in the 'Tools' column.

The following page will then load:

The screenshot shows the eMart STORE ADMINISTRATION interface. At the top, there's a logo with a green circle containing a small icon, followed by the word "eMART" in a stylized font. To the right of the logo are links for "Kisacasa eCommerce Solution" and "Support". Below the logo is a navigation bar with "STOREFRONT" on the left and "LOGOUT" on the right. Underneath the navigation bar is a horizontal menu with links: CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |. The main content area has a title "Editing products for category: Fresh Fruit [back to categories ... ]". Below this is a table with columns: Name, Brief Description, Long Description, Price, Discounted Price, and Tools. The table contains four rows of product data. At the bottom of the table is a form titled "Add new product:" with fields for name, brief description, long description, and price, along with current and ideal quantity fields and an "Add" button. The footer of the page includes copyright information: "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 16

This page contains the different products within the chosen category

### Add a product:

The screenshot shows the same eMart STORE ADMINISTRATION interface as Figure 16, but with a red box highlighting the "Add new product:" form at the bottom of the table. An arrow points from this red box down to a callout box at the bottom of the page.

Figure 17

Click in textboxes to add information about new products. NOTE: current and ideal quantity does not show on products page, they are for stock control purposes

## Edit a product:

Editing products for category: Fresh Fruit [back to categories ... ]

Name	Brief Description	Long Description	Price	Discounted Price	Tools
Golden Delicious Apples	Loose Golden Delicious Apples	Loose Golden Delicious Apples Class 1	0.27	0.00	<a href="#">Edit</a>

Figure 18

Click to edit products

Then the following page will load:

The screenshot shows the eMART Store Administration interface. At the top, there is a logo, navigation links for 'KISACASA e-commerce solution' and 'Support', and a main title 'STORE ADMINISTRATION'. Below this, there are links for 'STOREFRONT' and 'LOGOUT', and a menu bar with 'CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |'.

The main content area is titled 'Editing product: ID #2 — Golden Delicious Apples [back to products ... ]'. It contains the following fields and sections:

- Product name:** Golden Delicious Apples
- Product brief description:** Loose Golden Delicious Apples
- Product long description:** Loose Golden Delicious Apples Class 1
- Product supplier:** 1
- Product price:** 0.27
- Product discounted price:** 0.00
- Product belongs to these categories:** Fresh Fruit
- Remove this product from:** Fresh Fruit (with a Remove button)
- Assign product to this category:** Vegetables (with an Assign button)
- Move product to this category:** Vegetables (with a Move button)
- Product attributes:**
  - Colour: Red (with a Remove button)
  - Assign attribute to product: Size: XS (with an Assign button)
- Set display option for this product:** On Both (with a Set button)
- Remove product from catalog:** (with a red link)

Below these fields, there are two image upload sections:

- Thumbnail name:** goldendelicious.jpg (Choose File) No file chosen (Upload button)
- Large Image name:** goldendeliciousBig.jpg (Choose File) No file chosen (Upload button)

At the bottom of the page, there is a footer with the text: © Copyright eMART 2012 CREATED BY KISACASA

Figure 19

Here, all information for the chosen product can be edited

Edit basic details of product:

The screenshot shows a dark-themed user interface for editing product details. The fields and their current values are:

- Product name:** Golden Delicious Apples
- Product brief description:** Loose Golden Delicious Apples
- Product long description:** Loose Golden Delicious Apples Class 1
- Product supplier:** 1
- Product price:** 0.27
- Product discounted price:** 0.00

Annotations with red arrows and boxes point to specific fields:

- An arrow points to the "Product name" field with the label "Edit product name".
- An arrow points to the "Product brief description" field with the label "Edit brief description".
- An arrow points to the "Product long description" field with the label "Edit long description".
- An arrow points to the "Product supplier" field with the label "Edit supplier (enter supplier ID)".
- An arrow points to the "Product price" field with the label "Edit product price".
- An arrow points to the "Product discounted price" field with the label "Edit product discounted price".
- An arrow points to the "Update info" button with the label "Click to save changes".

Figure 20

### Edit product locations:

The information about the product category can be added. NOTE: please select options from dropdown boxes where relevant:

**Product belongs to these categories:** Fresh Fruit

**Remove this product from:**  
Fresh Fruit  Click to remove product from a category

**Assign product to this category:**  
Vegetables  Select and click to assign product to a category

**Move product to this category:**  
Vegetables  Select and click to move product to a category

**Product attributes:**  
Colour: Red  Select and click to remove an attribute

**Assign attribute to product:**  
Size: XS  Select and click to assign an attribute

**Set display option for this product:**  
On Both  Select and click to set a display option for product

**Remove product from catalog**

Click to remove the product from the catalogue. NOTE: if product is in the catalogue it will be displayed on eMart's home page. Removing from catalogue will not delete product

Figure 21

### Edit product images:

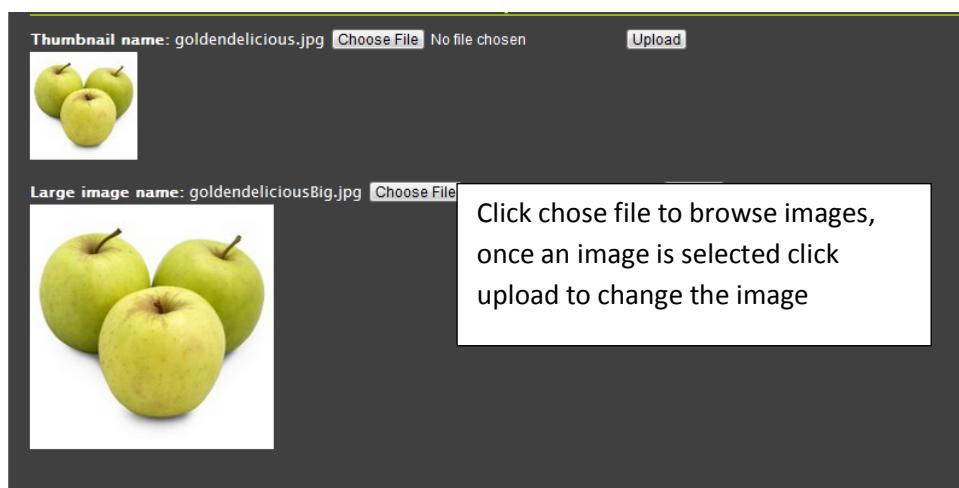


Figure 22

NOTE: Thumbnail will be displayed on home page/list of products in a category. Large image will be displayed when a product is selected to see more about the product

## Stock Control

On the page after log in (Figure 1), there are tabs displayed along the top:

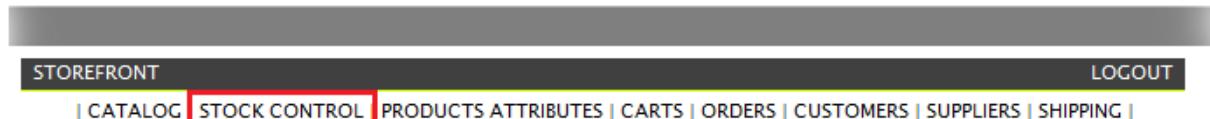


Figure 23

To access the stock control, click the stock control tab. This page will then load:

A screenshot of the eMart Stock Administration page. At the top, there is a logo for "eMART" with a green circle containing a bee-like character. To the right of the logo is a shopping bag icon, the text "Kisacasa eCommerce Solution", and a "Support" link. Below this is a section titled "STORE ADMINISTRATION". The main content area has a dark header bar with "STOREFRONT" on the left and "LOGOUT" on the right. Below this is another navigation bar with tabs: "CATALOG", "STOCK CONTROL" (highlighted with a red border), "PRODUCTS ATTRIBUTES", "CARTS", "ORDERS", "CUSTOMERS", "SUPPLIERS", and "SHIPPING". The main body of the page is titled "eMart Stock" and contains a table with 15 rows of data. The table has four columns: "Item ID", "Item Name", "Status", and "Tools". The "Status" column uses color coding: "Stocked" is grey, "OutOfStock" is yellow, and "OutofStock" is red. The "Tools" column contains "View Details" buttons for each item.

Figure 24

Here the items in the eMart system can be viewed and whether they are stocked or not.

NOTE: Out of stock = 0% of product remaining. Reorder level = 50% of ideal quantity. Medium = 75% - 100% of ideal quantity. Stocked = greater than ideal quantity.

eMart Stock			
Item ID	Item Name	Status	Tools
8	Asparagus Tips	Stocked	<a href="#">View Details</a>

Figure 25

Click to view details of product stock control

This page will then be displayed:

The screenshot shows the eMart Stock Control interface. At the top, there's a navigation bar with the eMart logo, a shopping bag icon, 'Kisacasa eCommerce Solution', 'Support', and 'STORE ADMINISTRATION'. Below the navigation is a menu bar with 'STOREFRONT' and 'LOGOUT' on the left, and links for 'CATALOG', 'STOCK CONTROL', 'PRODUCTS ATTRIBUTES', 'CARTS', 'ORDERS', 'CUSTOMERS', 'SUPPLIERS', and 'SHIPPING' on the right. The main content area is titled 'Editing item: ID #8 [back to stock control ...]'. It contains a form with the following fields:

Item Name:	Asparagus Tips
Current Quantity:	100
Ideal Quantity:	50
Status:	Stocked
Supplier ID:	4

At the bottom right of the form is a 'Update info' button. The footer of the page includes copyright information: '© Copyright eMart 2012' and 'CREATED BY KISACASA'.

Figure 26

Here, the current and ideal quantity of the product can be changed. The status will change accordingly.

NOTE: The item name or supplier ID cannot be changed in this section. The status of the product cannot be changed manually.

The screenshot shows the same 'Editing item' form as Figure 26. The 'Update info' button at the bottom right is highlighted with a red box. A callout box with the text 'NOTE: click to save changes' points to this button.

Figure 27

## Products and their attributes

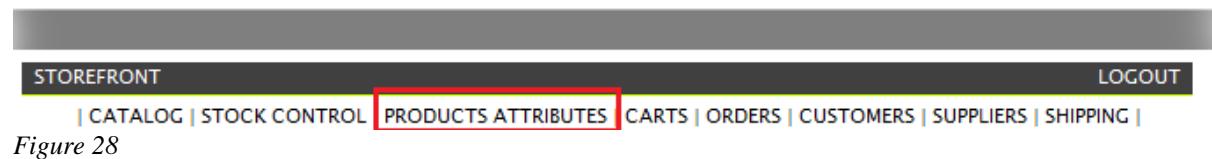


Figure 28

When clicking products attributes tab, the following page will be load:

The screenshot shows the "eMart Attributes" management page. At the top, there is a logo for "eMART" with a green circle and a bee icon. To the right is a shopping bag icon, the text "Kisacasa eCommerce Solution", and links for "Support" and "STORE ADMINISTRATION". Below this is another dark grey header bar with "STOREFRONT" and "LOGOUT" on the sides, and the same menu bar as Figure 28 below it.

The main content area has a dark grey background and features a table titled "eMart Attributes". The table has two columns: "Attribute Name" and "Tools". It lists three attributes: "Size", "Weight", and "Colour". Each row includes a "Tools" column with buttons for "Edit Attribute Values", "Edit", and "Delete".

Below the table, there is a section for adding new attributes with a form labeled "Add new attribute:" containing a text input field "[name]" and a "Add" button. At the bottom of the page, there is a copyright notice: "© Copyright eMart 2012" and "CREATED BY KISACASA".

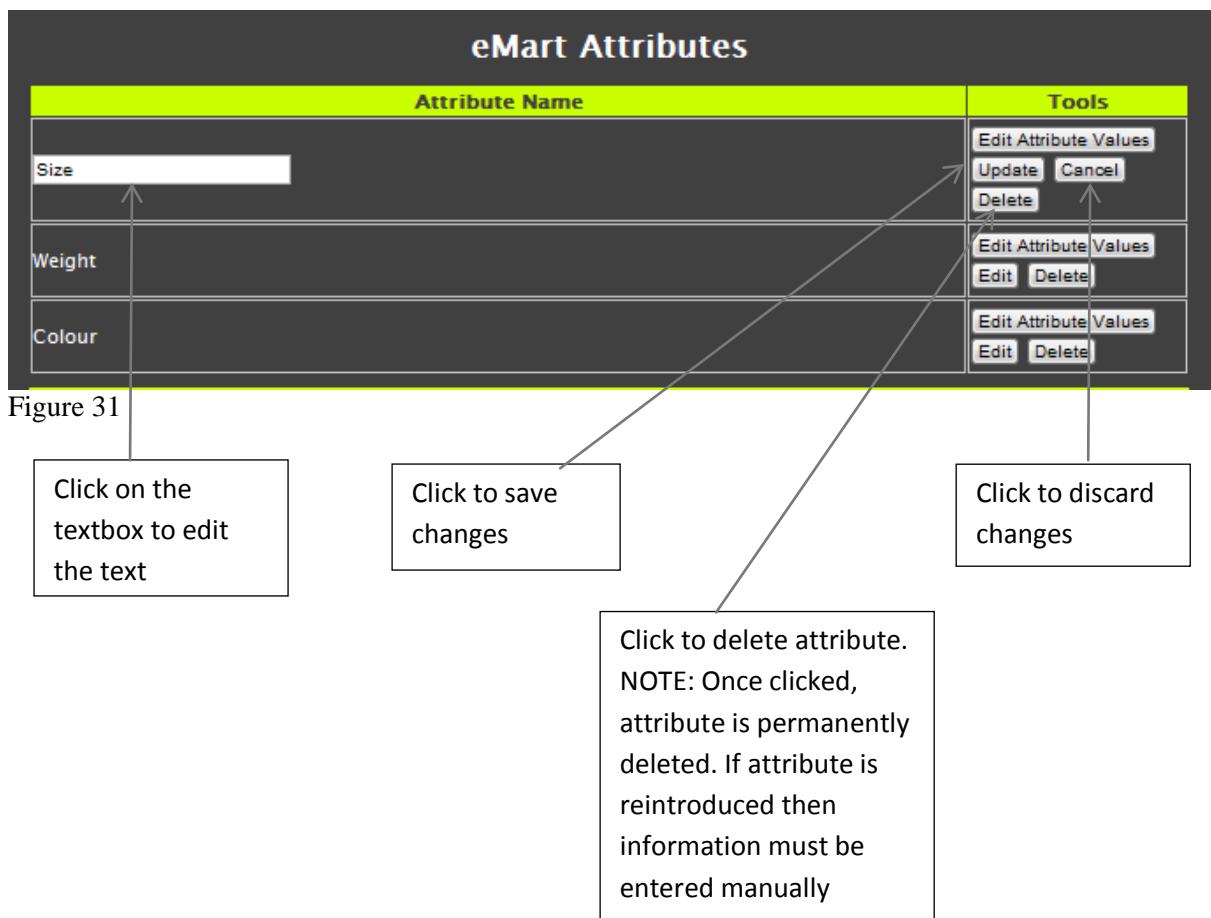
Figure 29

Add an attribute:

The screenshot shows a dark grey form for adding a new attribute. It contains a text input field labeled "Add new attribute:" with the placeholder "[name]", and a "Add" button to its right.

Figure 30

### Edit an attribute:



### View attribute values:



This page will then load:

The screenshot shows the eMart Store Administration interface. At the top, there is a logo with a green circle containing a bee-like character, followed by the word "eMART" in green and white. To the right of the logo are links for "Kisacasa eCommerce Solution" and "Support". Below this is a section titled "STORE ADMINISTRATION". A navigation bar at the top has "STOREFRONT" on the left and "LOGOUT" on the right. Below the navigation bar is a horizontal menu with links: CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |. The main content area is titled "Editing attribute values for attribute: Size [back to attributes ...]". It contains a table with columns for "Attribute Value" and "Tools". The "Attribute Value" column lists size codes: XS, S, M, L, and XL. The "Tools" column for each row contains "Edit" and "Delete" buttons. Below the table is a form for adding new attribute values, with a text input field labeled "[value]" and a "Add" button. At the bottom of the page, there is a copyright notice: "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 33

### Edit attribute value:

The screenshot shows the "Edit attribute value" screen. It features a table with a single row for the attribute value "XS". The "Attribute Value" column contains the text "XS", and the "Tools" column contains buttons for "Update", "Cancel", and "Delete". Callout boxes provide instructions: one points to the "XS" text input field with the text "Click on the textbox to edit the text"; another points to the "Update" button with "Click to save changes"; a third points to the "Delete" button with "Click to delete attribute. NOTE: Once clicked, attribute is permanently deleted. If attribute is reintroduced then information must be entered manually"; and a fourth points to the "Cancel" button with "Click to discard changes".

## Shopping Carts

To view the amount of carts click the carts tab on the top:



The following page will then load:

This screenshot shows the 'eMart Shopping Carts' page. At the top, there is a logo with the word 'eMART' in green and yellow, and a green shopping bag icon. To the right of the icon, it says 'Kisacasa eCommerce Solution' and 'Support'. Below that, it says 'STORE ADMINISTRATION'. At the very top, there is another dark grey bar with 'STOREFRONT' on the left and 'LOGOUT' on the right. The main content area has a dark background. It displays the title 'eMart Shopping Carts:' and a sub-header 'Select carts:'. Below this, there is a dropdown menu with the option 'All shopping carts' selected. Next to it are two buttons: 'Count Old Shopping Carts' and 'Delete Old Shopping Carts'. At the bottom of the page, it says '© Copyright eMart 2012' and 'CREATED BY KISACASA'. Below this, the text 'Figure 36' is visible.

How many carts have been saved in a certain time frame can be seen by choosing from the dropdown box:

This screenshot shows the same 'eMart Shopping Carts' page as Figure 36, but with a different focus. The dropdown menu under 'Select carts:' now has the option 'All shopping carts' highlighted with a blue selection bar. The other options in the dropdown are 'One day old', 'Ten days old', 'Twenty days old', 'Thirty days old', and 'Ninety days old'. The rest of the page, including the buttons and copyright information, remains the same.

Figure 37

Counting old shopping carts:



Figure 38

Delete old shopping carts:



Figure 39

## Customers

By clicking on the customer tab at the top of the page eMart's customers can be viewed:



Figure 40

The following page will load:

The screenshot shows the eMart Store Administration interface. At the top left is the eMart logo. To its right is a green shopping bag icon, followed by the text "Kisacasa eCommerce Solution" and "Support". Below this is the "STORE ADMINISTRATION" header. A navigation bar below the header includes "STOREFRONT" on the left and "LOGOUT" on the right, with several menu items in the middle: CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |. The main content area displays "eMart Customers" with a table showing one customer record: First Name (customer), Surname (one), Email (customer1@localhost), and Tools (View Details). Below this is a form titled "Add new customer:" with fields for first name, surname, email address, password, telephone number, street address, town, city, county, postcode, and an "Add" button. At the bottom of the page, a copyright notice reads "© Copyright eMart 2012 CREATED BY KISACASA".

Figure 41

### Add new customer:

The screenshot shows the "Add new customer" form. It has two rows of input fields. The first row contains [first name], [surname], [email address], [password], and [telephone number]. The second row contains [street address], [town], [city], [county], [postcode], and an "Add" button. The entire form is set against a dark background.

Figure 42

Click on the textboxes to add information on the new customer and click add to save changes.

### View customer details:

The screenshot shows a table with four columns: First Name, Surname, Email, and Tools. The first row contains "customer", "one", "customer1@localhost", and a "View Details" button. A red box highlights the "View Details" button. A callout box with a pointer points to this button, containing the text: "Click to edit, delete and view details of customers".

First Name	Surname	Email	Tools
customer	one	customer1@localhost	<a href="#">View Details</a>

Figure 43

The following page will then load:

The screenshot shows the 'eMart' logo on the left and a navigation bar with a shopping bag icon, 'Kisacasa eCommerce Solution', 'Support', 'STORE ADMINISTRATION', 'STOREFRONT', and 'LOGOUT'. Below the navigation is a menu bar with links: CATALOG, STOCK CONTROL, PRODUCTS ATTRIBUTES, CARTS, ORDERS, CUSTOMERS, SUPPLIERS, and SHIPPING. The main content area is titled 'Editing customer: ID #1 [back to customers ...]'. It contains form fields for customer details and shipping address, with values filled in. At the bottom are 'Update info' and 'Remove customer from database' buttons.

First name:	customer
Surname:	one
Email address:	customer1@localhost
Password:	5fb21e5edf34bfe983bf9f41754
Telephone number:	447768033426
Shipping Address:	
Street	113 hall lane
Town	Liverpool
City	Liverpool
County	merseyside
Postcode	L7 8tf

[Update info](#) [Remove customer from database](#)

© Copyright eMart 2012  
CREATED BY KISACASA

Figure 44

### Edit customer details:

Here, the customer's details can be edited e.g. first name, surname, email address, password (the encrypted password), telephone number and shipping address. NOTE: click update info to save changes.

### Delete a customer:



Figure 45

NOTE: Once clicked, customer is permanently deleted. If customer returns, information will have to be re-entered

## Suppliers

eMart's suppliers can be viewed by clicking on the suppliers tab on the top:



Figure 46

You will then be faced with:

A screenshot of the "eMart Suppliers" page. The page has a dark header with "eMart Suppliers" in white. Below the header is a table with three columns: "Name", "Email", and "Tools". The first row contains the values "Green & Co" and "GreenCo@localhost.com" under the "Tools" column is a "View Details" button. Below the table is a section titled "Add new supplier:" with several input fields: company name, email address, telephone number, street address, town, city, county, postcode, and an "Add" button. At the bottom of the page, there is a copyright notice: "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 47

### Add new supplier:

A screenshot of the "Add new supplier" form. It consists of several input fields arranged in a grid. The first row contains fields for "company name", "email address", "telephone number", "street address", and "town". The second row contains fields for "city", "county", "postcode", and an "Add" button. All fields are currently empty.

Figure 48

Click on textboxes to enter information. Click add to save supplier

## **View supplier details:**

Details on suppliers can be view by clicking view details on the eMart Suppliers page (see Figure 47):

The screenshot shows the eMart Suppliers page. At the top, there is a logo for 'eMART' with a green bee icon. To the right of the logo are links for 'Kisacasa eCommerce Solution' and 'Support'. Below the logo, the text 'STORE ADMINISTRATION' is displayed. A navigation bar at the top includes 'STOREFRONT' on the left and 'LOGOUT' on the right. Below the navigation bar, a horizontal menu lists: CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |. The main content area is titled 'Editing supplier: ID #1 [back to suppliers ...]'. It contains a form with the following data:

Company Name:	Green & Co
Email address:	GreenCo@localhost.com
Telephone number:	01515543234
Company Address:	Street: Green Field Town: Huyton City: Liverpool County: merseyside Postcode: L30 8HG

At the bottom of the form are three buttons: 'View Deliveries', 'Update info', and 'Remove supplier from database'.

© Copyright eMart 2012  
CREATED BY KISACASA

Figure 49

## **Edit supplier details:**

The supplier name , email address, telephone number and company address can be updated. NOTE: Once information is updated click update info to save changes.

## **Delete a supplier:**

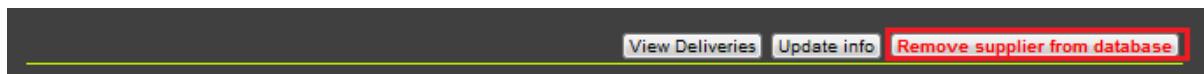


Figure 50

NOTE: Once clicked, customer is permanently deleted. If customer returns, information will have to be re-entered

## **View supplier deliveries:**

Deliveries from the supplier can be viewed by clicking view deliveries:

The screenshot shows the eMart STORE ADMINISTRATION interface. At the top right, there is a green shopping bag icon, the text "Kisacasa eCommerce Solution", and links for "Support", "STORE ADMINISTRATION", "LOGOUT", and "CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |". Below this, a sub-menu bar has "STOREFRONT" on the left and "LOGOUT" on the right. The main content area is titled "Editing deliveries for supplier: Green & Co [back to supplier ... ]". It contains a table with one row: Date (2012-05-12), Time (12:00:00), and Tools (Edit). Below the table is a form for adding new deliveries, with fields for Date (YYYY-MM-DD) and Time (HH:MM:SS), and an "Add" button. At the bottom right of the page, there is a copyright notice: "© Copyright eMart 2012" and "CREATED BY KISACASA".

Figure 51

## **Add a supplier delivery:**

The screenshot shows the "Add new delivery:" form. It has two text input fields: "YYYY-MM-DD" and "HH:MM:SS", and an "Add" button.

Figure 52

Click in the textboxes to add details of a new delivery. NOTE: click add to save changes

## **Edit a supplier delivery:**

To edit a delivery, click edit.

The screenshot shows the "Editing deliveries for supplier: Green & Co [back to supplier ... ]" form. It contains a table with one row: Date (2012-05-12), Time (12:00:00), and Tools (Edit). Below the table is a form for adding new deliveries, with fields for Date (YYYY-MM-DD) and Time (HH:MM:SS), and an "Add" button.

Figure 53

The following page will load:

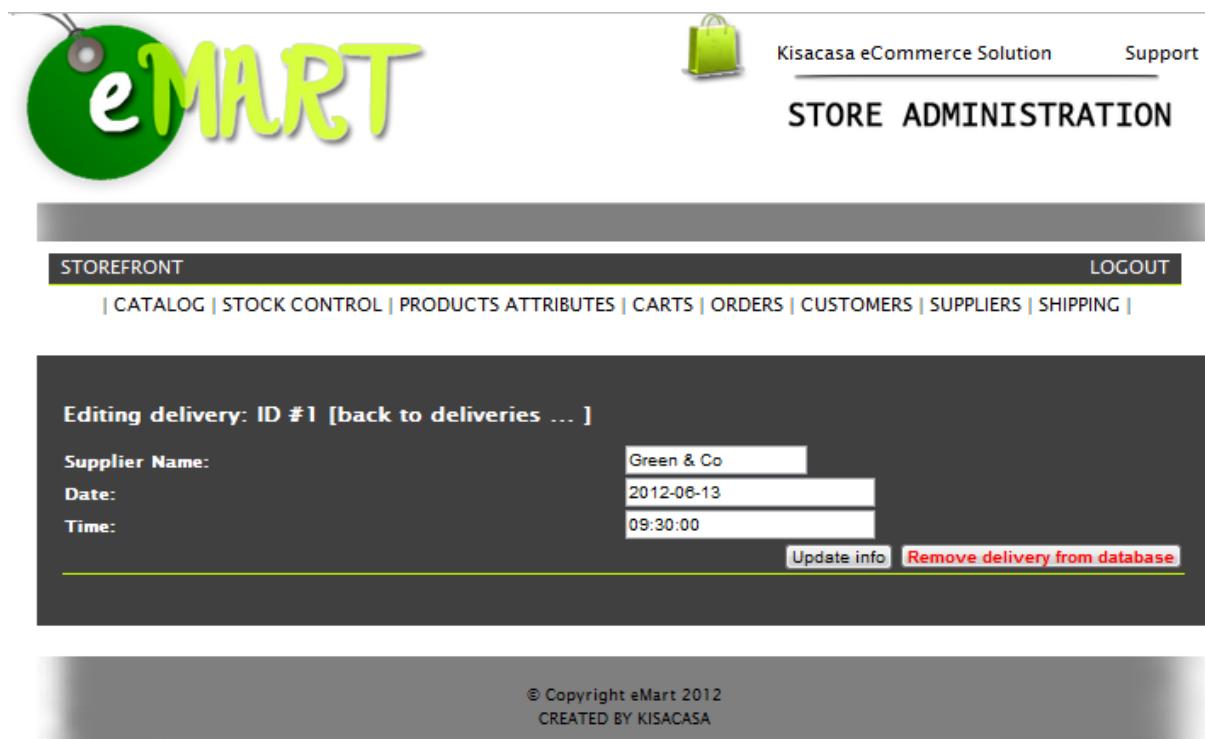


Figure 54

### Edit a supplier delivery:

The date and time of the delivery can be edited and it can also be cancelled:

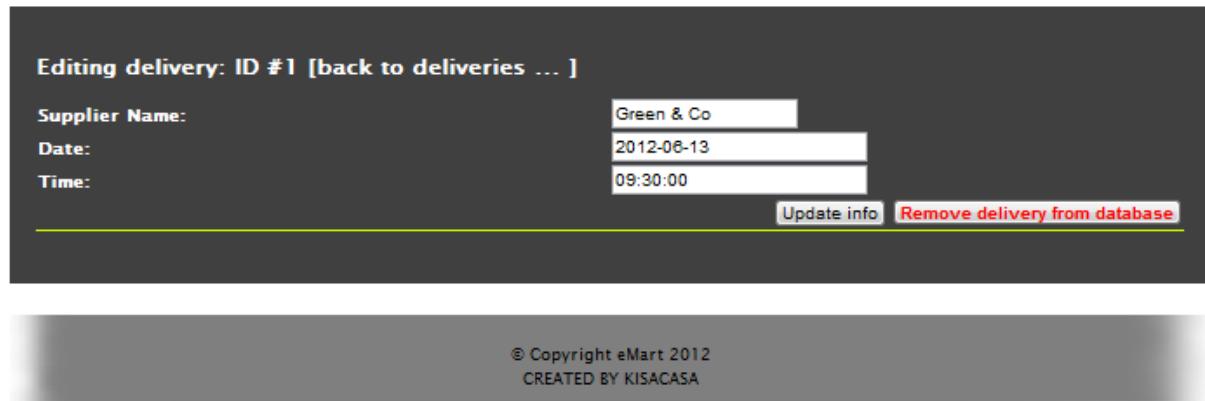


Figure 55

NOTE: The supplier name cannot be changed from deliveries. Click update into to save changes to delivery. If you delete the delivery it will be permanently deleted from the database.

## Orders

By clicking on the orders tab at the top of the page all of eMart's order history can be viewed:



Figure 56

You will then be directed to:

The screenshot shows the 'eMart Orders' page. At the top, there is a navigation bar with 'STOREFRONT' and 'LOGOUT'. Below this is a secondary navigation bar with links: CATALOG, STOCK CONTROL, PRODUCTS ATTRIBUTES, CARTS, ORDERS, CUSTOMERS, SUPPLIERS, and SHIPPING. The main content area is titled 'eMart Orders' and contains five search filters:

- Show orders by customer: dropdown set to 'customer Four' with a 'Go!' button.
- Get by order ID: input field with a 'Go!' button.
- Show the most recent 20 orders: input field with a 'Go!' button.
- Show all records created between [start date] and [end date]: input fields with a 'Go!' button.
- Show orders by status: dropdown set to 'Received' with a 'Go!' button.

At the bottom of the page, there is a copyright notice: © Copyright eMart 2012 CREATED BY KISACASA.

Figure 57

On this page, filters are used to search for orders under a certain criteria.

‘Show orders by customers’ – chose a customer from the dropdown box and click go:

The screenshot shows the 'eMart Orders' page with the 'Customer' filter set to 'customer Four'. The main content area displays a table of order results:

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 58

‘Get by order ID’ - type an ID into the textbox:

**eMart Orders**

Show orders by customer  Go!

Get by order ID  Go!

Show the most recent  orders Go!

Show all records created between  and  Go!

Show orders by status  Go!

Order ID	Date Created	Date Shipped	Status	Customer	Tools
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 59

‘Show the most recent’ - type a number of the most recent orders that need to be viewed:

**eMart Orders**

Show orders by customer  Go!

Get by order ID  Go!

Show the most recent  orders Go!

Show all records created between  and  Go!

Show orders by status  Go!

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 60

‘Show all records created between’ – enter two dates which you wish to view deliveries between:

**eMart Orders**

Show orders by customer  Go!

Get by order ID  Go!

Show the most recent  orders Go!

Show all records created between  and  Go!

Show orders by status  Go!

Order ID	Date Created	Date Shipped	Status	Customer	Tools
3	2012-05-15 22:59:19		Awaiting Stock Confirmation	customer Four	<a href="#">View Details</a>
1	2012-05-15 18:01:32	2012-05-15 18:19:19	Complete	customer Four	<a href="#">View Details</a>

Figure 61

NOTE: both dates must be entered in the format YYYY-MM-DD HH:MM:SS

'Show order by status'- chose from the dropdown box:

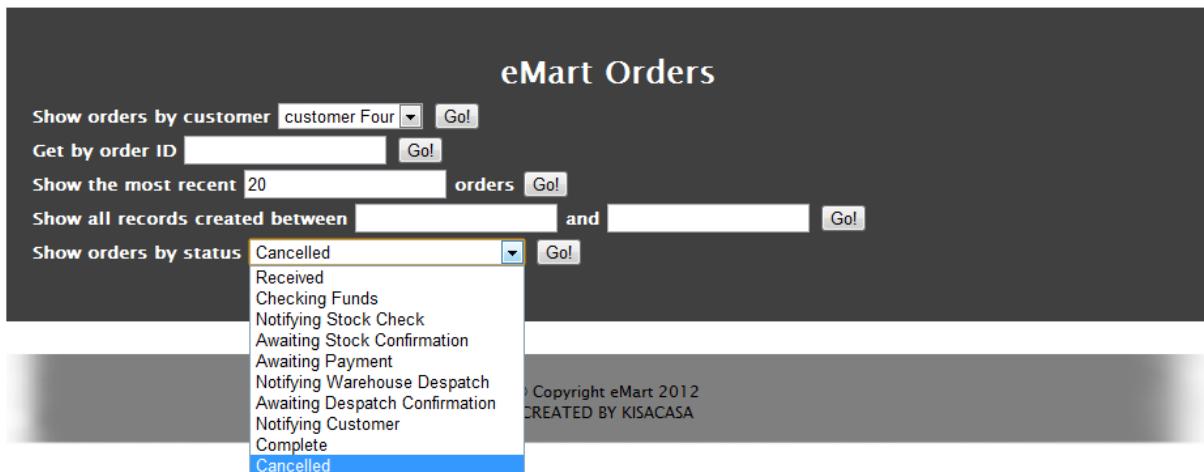


Figure 62

Click go:



Figure 63

NOTE: no orders are currently in the 'checking funds' status

When orders are in the awaiting stock confirmation status, eMart employees must check the stock levels and confirm or cancel the order. When an order is made, an email is sent to emartwarehouse@kisacasa.com. Open your email provider, e.g. Mozilla earlybird mail:

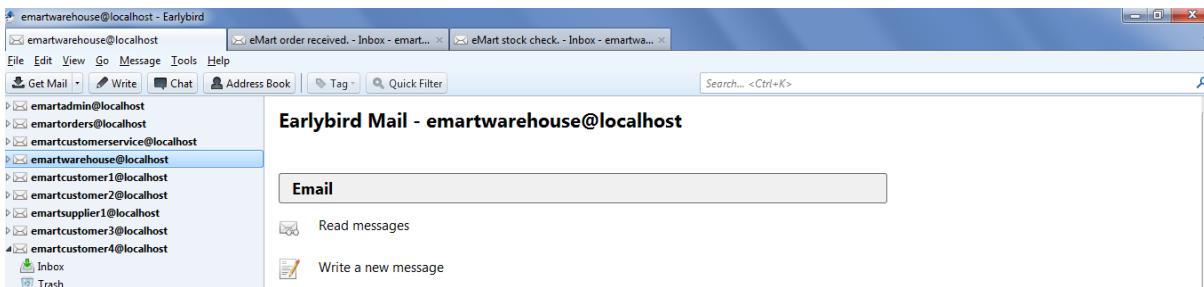


Figure 64

Click read messages:

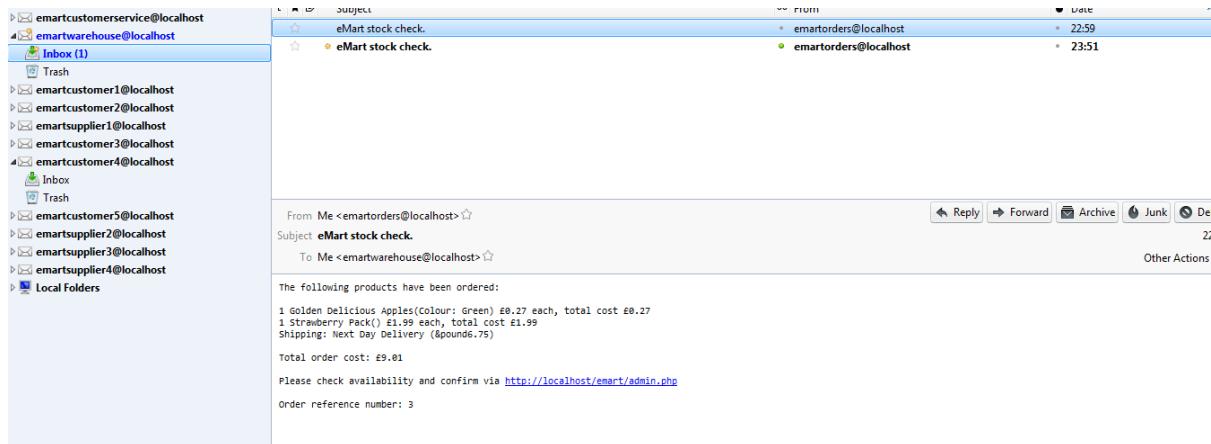


Figure 65

Here employees will be able to view email. Messages entitled eMart stock check will need to be opened and employees will need to confirm that the items are in stock. You will need to navigate to orders (see figure 56 & figure 57) and filter the search by order ID (see figure 58). Using the order reference on the email:



Figure 66

## View and Edit order details:



Kisacasa eCommerce Solution      Support

**STORE ADMINISTRATION**

STOREFRONT      LOGOUT

| CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |

**Editing details for order ID: 3 [ back to admin orders... ]**

Total Amount:	£2.26
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-15 22:59:19
Date Shipped:	
Status:	Awaiting Stock Confirmation
Authorization Code:	NVOOKN
Reference Number:	2171901868
Comments:	
Customer Name:	customer Four
Shipping Address:	Street: 113 hall lane Town: Liverpool City: Liverpool County: merseyside Postcode: L7 8tf
Customer Email:	emartcustomer4@localhost
Telephone Number:	447768033426

**Edit   Update   Cancel   Confirm Stock for Order**

Figure 67

## Order Processing status:

To confirm that the order has the necessary stock items, click ‘confirm stock order’:



Kisacasa eCommerce Solution      Support

**STORE ADMINISTRATION**

STOREFRONT      LOGOUT

| CATALOG | STOCK CONTROL | PRODUCTS ATTRIBUTES | CARTS | ORDERS | CUSTOMERS | SUPPLIERS | SHIPPING |

**Editing details for order ID: 3 [ back to admin orders... ]**

Total Amount:	£2.26
Shipping:	Next Day Delivery (£6.75)
Date Created:	2012-05-15 22:59:19
Date Shipped:	
Status:	Awaiting Despatch Confirmation
Authorization Code:	NVOOKN
Reference Number:	2171901868
Comments:	
Customer Name:	customer Four
Shipping Address:	Street: 113 hall lane Town: Liverpool City: Liverpool County: merseyside Postcode: L7 8tf
Customer Email:	emartcustomer4@localhost
Telephone Number:	447768033426

**Edit   Update   Cancel   Confirm Shipment for Order**

Figure 68

An email will now be sent to emartwarehouse@kisacasa.com:

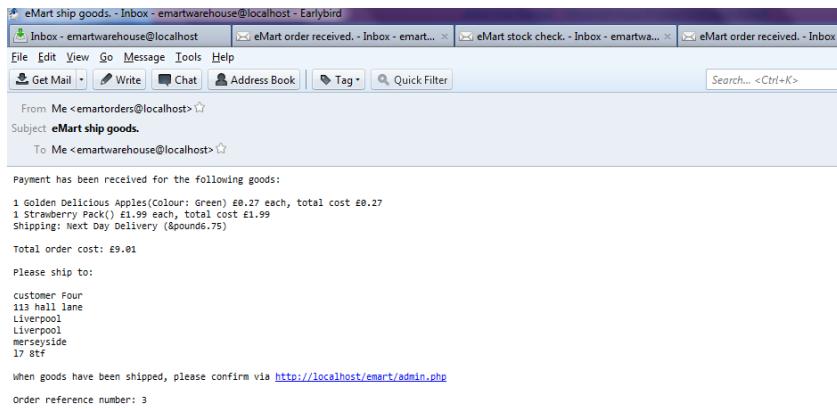


Figure 69

eMart will now have to confirm that the order has been shipped. Similarly, navigate to orders and filter (see figures 56, 57, 58) using the order reference on the email:

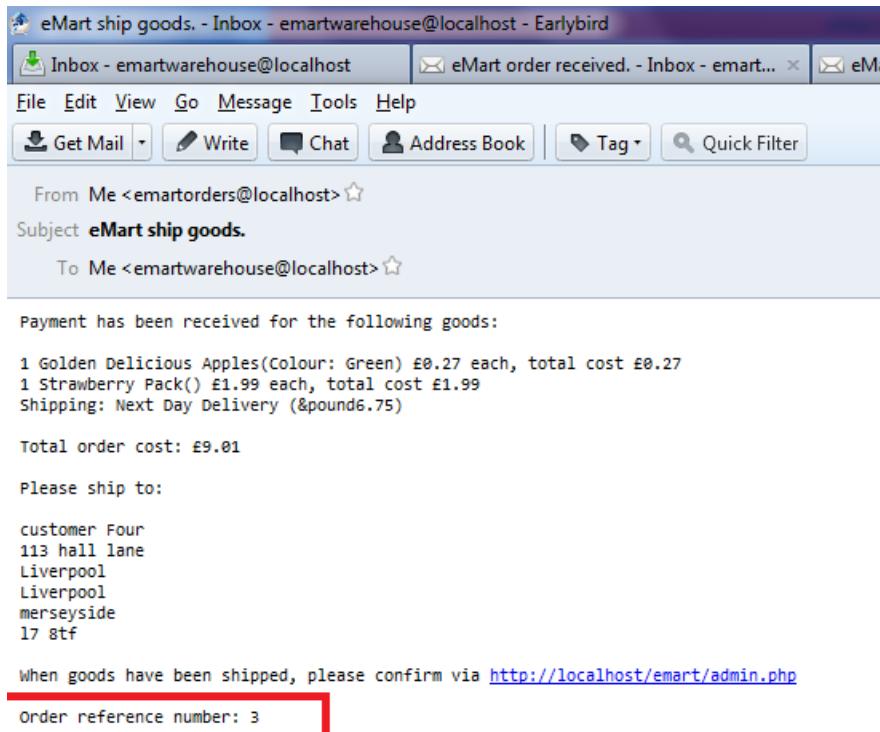


Figure 70

When the order is shipped, click ‘Confirm Shipment for Order’:

Figure 71

The order will now have a ‘complete’ status when the order is navigated to (see figures 56, 57 & 58) :

3	2012-05-15 22:59:19	2012-05-16 00:13:42	Complete	customer Four	<a href="#">View Details</a>
---	---------------------	---------------------	----------	---------------	------------------------------

Figure 72

The customer will receive an email confirming that the goods have been dispatched:

Figure 72

NOTE: On the ‘order details’ page it displays the content of the order:

Order contains these products:				
Product ID	Product Name	Quantity	Unit Cost	Subtotal
2	Golden Delicious Apples Colour: Green	1	£0.27	£0.27
5	Strawberry Pack	1	£1.99	£1.99

Figure 73

NOTE: On the ‘order details’ page it also displays the order audit trail:

Order audit trail:			
Audit ID	Created On	Code	Message
53	2012-05-15 22:59:19	10000	Order Processor started.
54	2012-05-15 22:59:19	20000	PsInitialNotification started.
55	2012-05-15 22:59:19	20002	Notification e-mail sent to customer.
56	2012-05-15 22:59:19	20001	PsInitialNotification finished.
57	2012-05-15 22:59:19	20100	PsCheckFunds started.
58	2012-05-15 22:59:21	20102	Funds available for purchase.
59	2012-05-15 22:59:21	20101	PsCheckFunds finished.
60	2012-05-15 22:59:21	20200	PsCheckStock started.
61	2012-05-15 22:59:21	20202	Notification email sent to warehouse.
62	2012-05-15 22:59:21	20201	PsCheckStock finished.
63	2012-05-15 22:59:21	10001	Order Processor finished.
75	2012-05-16 00:06:21	10000	Order Processor started.
76	2012-05-16 00:06:21	20300	PsStockOk started.
77	2012-05-16 00:06:21	20302	Stock confirmed by warehouse.
78	2012-05-16 00:06:21	20301	PsStockOk finished.
79	2012-05-16 00:06:21	20400	PsTakePayment started.
80	2012-05-16 00:06:23	20402	Funds deducted from customer credit card account.
81	2012-05-16 00:06:23	20401	PsTakePayment finished.
82	2012-05-16 00:06:23	20500	PsShipGoods started.
83	2012-05-16 00:06:23	20502	Ship goods e-mail sent to warehouse.
84	2012-05-16 00:06:23	20501	PsShipGoods finished.
85	2012-05-16 00:06:23	10001	Order Processor finished.
86	2012-05-16 00:18:42	10000	Order Processor started.
87	2012-05-16 00:18:42	20600	PsShipOk started.
88	2012-05-16 00:18:42	20602	Order dispatched by warehouse.
89	2012-05-16 00:18:43	20601	PsShipOk finished.
90	2012-05-16 00:18:43	20700	PsFinalNotification started.
91	2012-05-16 00:18:43	20702	Dispatch email send to customer.
92	2012-05-16 00:18:43	20701	PsFinalNotification finished.
93	2012-05-16 00:18:43	10001	Order Processor finished.

Figure 74

## Shipping

To navigate to the different shipping rates, chooses the shipping tab on the top:



Figure 75

The following page will load:

eMart Shipping Methods		
Shipping Type	Shipping Cost	Tools
Next Day Delivery (£6.75)	6.75	<a href="#">Edit</a> <a href="#">Delete</a>
3-4 Days (£4.50)	4.50	<a href="#">Edit</a> <a href="#">Delete</a>
7 Days (£3)	3.00	<a href="#">Edit</a> <a href="#">Delete</a>

**Add new shipping type:**  
[type] [cost] [Add](#)

Figure 76

Add a new shipping type:

eMart Shipping Methods		
Shipping Type	Shipping Cost	Tools
Next Day Delivery (£6.75)	6.75	<a href="#">Edit</a> <a href="#">Delete</a>
3-4 Days (£4.50)	4.50	<a href="#">Edit</a> <a href="#">Delete</a>
7 Days (£3)	3.00	<a href="#">Edit</a> <a href="#">Delete</a>

**Add new shipping type:**  
[type] [cost] [Add](#)

Figure 77

Click in textboxes to add text. Click add to save changes

## Update or delete shipping types:

**eMart Shipping Methods**

Shipping Type	Shipping Cost	Tools
Next Day Delivery (£6.75)	6.75	<input type="button" value="Update"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>
3-4 Days (£4.50)	4.50	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7 Days (£3)	3.00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

**Add new shipping type:**

[type] [cost]

Figure 78

Edit text

Click to save change

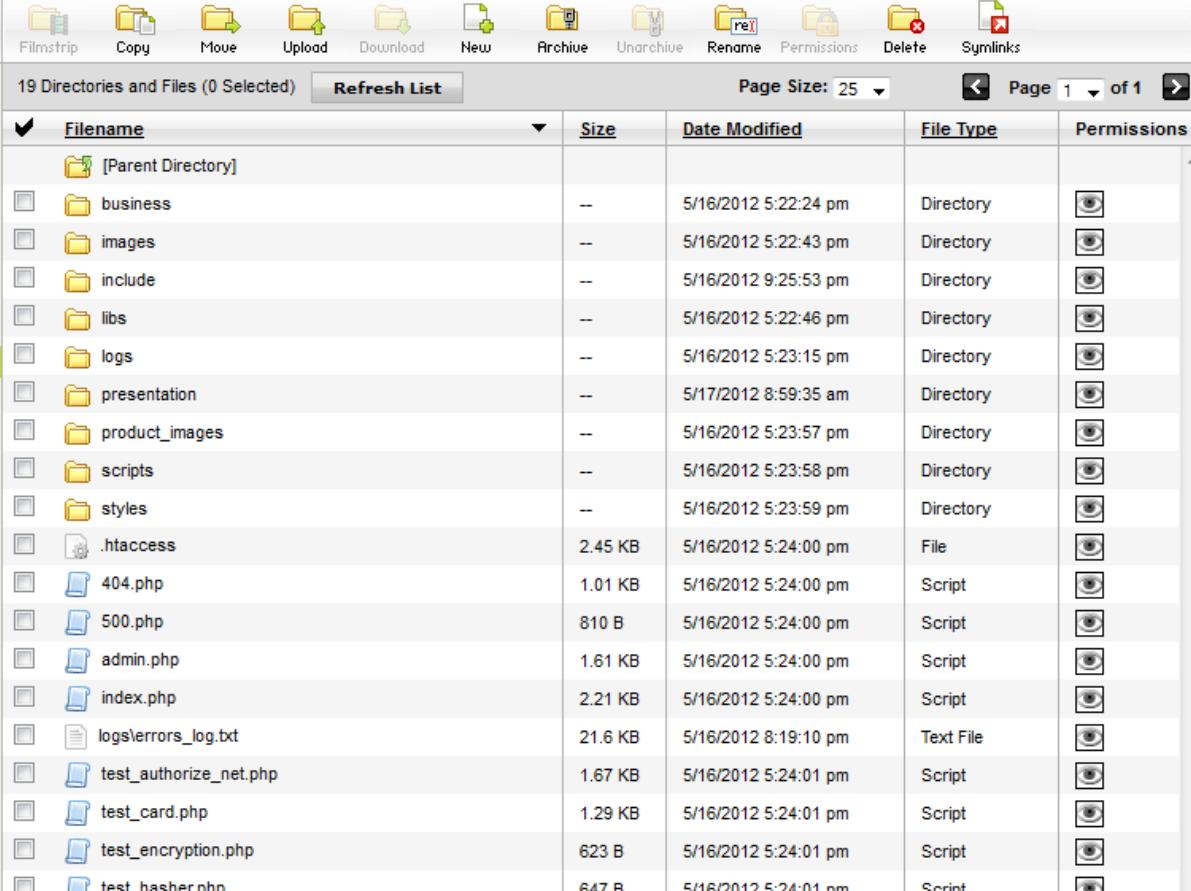
Discard changes

Click to delete shipping type. NOTE: when clicked, shipping type is deleted permanently. If shipping type needs to be reintroduce, data will need to be re-entered

## Configuration

Update sysadmin password:

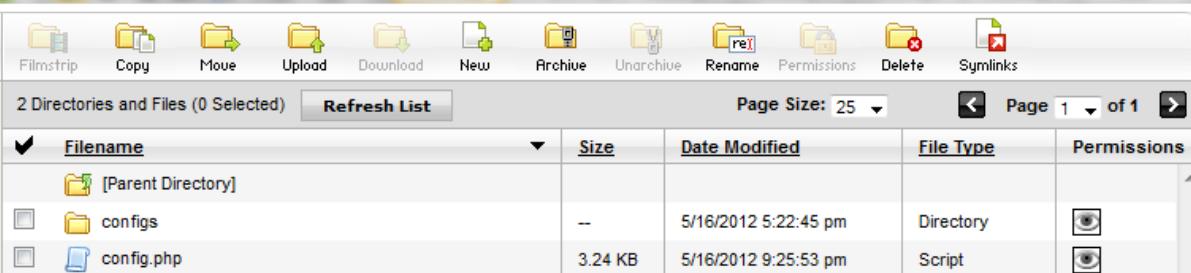
Root directory of your hosting account >>



A screenshot of a file manager interface showing the root directory contents. The toolbar at the top includes: Filmstrip, Copy, Move, Upload, Download, New, Archive, Unarchive, Rename, Permissions, Delete, and Symlinks. Below the toolbar is a header bar with "19 Directories and Files (0 Selected)", a "Refresh List" button, and a "Page Size: 25" dropdown. The main area is a table with columns: Filename, Size, Date Modified, File Type, and Permissions. The table lists 19 entries, including several sub-directories like "business", "images", "include", "libs", "logs", "presentation", "product\_images", "scripts", "styles", and various PHP files like ".htaccess", "404.php", "500.php", "admin.php", "index.php", "logserrors\_log.txt", "test\_authorize\_net.php", "test\_card.php", "test\_encryption.php", and "test\_hasher.php". Most files are script files (File Type: Script) and have a size of 0 KB or less.

Filename	Size	Date Modified	File Type	Permissions
[Parent Directory]				
business	--	5/16/2012 5:22:24 pm	Directory	
images	--	5/16/2012 5:22:43 pm	Directory	
include	--	5/16/2012 9:25:53 pm	Directory	
libs	--	5/16/2012 5:22:46 pm	Directory	
logs	--	5/16/2012 5:23:15 pm	Directory	
presentation	--	5/17/2012 8:59:35 am	Directory	
product_images	--	5/16/2012 5:23:57 pm	Directory	
scripts	--	5/16/2012 5:23:58 pm	Directory	
styles	--	5/16/2012 5:23:59 pm	Directory	
.htaccess	2.45 KB	5/16/2012 5:24:00 pm	File	
404.php	1.01 KB	5/16/2012 5:24:00 pm	Script	
500.php	810 B	5/16/2012 5:24:00 pm	Script	
admin.php	1.61 KB	5/16/2012 5:24:00 pm	Script	
index.php	2.21 KB	5/16/2012 5:24:00 pm	Script	
logserrors_log.txt	21.6 KB	5/16/2012 8:19:10 pm	Text File	
test_authorize_net.php	1.67 KB	5/16/2012 5:24:01 pm	Script	
test_card.php	1.29 KB	5/16/2012 5:24:01 pm	Script	
test_encryption.php	623 B	5/16/2012 5:24:01 pm	Script	
test_hasher.php	647 B	5/16/2012 5:24:01 pm	Script	

include >>



A screenshot of a file manager interface showing the "include" directory contents. The toolbar at the top is identical to the previous screenshot. The main area is a table with columns: Filename, Size, Date Modified, File Type, and Permissions. The table lists 2 entries: a sub-directory "configs" and a file "config.php". Both entries have a size of 0 KB.

Filename	Size	Date Modified	File Type	Permissions
[Parent Directory]				
configs	--	5/16/2012 5:22:45 pm	Directory	
config.php	3.24 KB	5/16/2012 9:25:53 pm	Script	

Open config.php >> edit word in red box.

NOTE: ensure you keep quotation marks

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
config.php

25 define('SEND_ERROR_MAIL', false);
26 define('ADMIN_ERROR_MAIL', 'emartadmin@localhost');
27 define('SENDMAIL_FROM', 'errors@localhost');
28 ini_set('sendmail_from', SENDMAIL_FROM);

29
30 // By default we don't log errors to a file
31 define('LOG_ERRORS', true);
32 define('LOG_ERRORS_FILE', 'logs\errors_log.txt'); // Windows
33
34 /* Generic error message to be displayed instead of debug info
35 (when DEBUGGING is false) */
36 define('SITE_GENERIC_ERROR_MESSAGE', '<h1>eMart Error!</h1>');
37
38 // Database connectivity setup
39 define('DB_PERSISTENCY', 'true');
40 define('DB_SERVER', 'localhost');
41 define('DB_USERNAME', 'sysadmin');
42 define('DB_PASSWORD', 'password' );
43 define('DB_DATABASE', 'emart');
44 define('PDO_DSN', 'mysql:host=' . DB_SERVER . ';dbname=' . DB_DATABASE);

45
46 // Server HTTP port (can omit if the default 80 is used)
47 define('HTTP_SERVER_PORT', '80');

48 /* Name of the virtual directory the site runs in, for example:
49 '/emart/' if the site runs at http://www.example.com/emart/
50 '' if the site runs at http://www.example.com/ */
51 define('VIRTUAL_LOCATION', '/emart/');

52
53 // Configure product lists display options
54 define('SHORT_PRODUCT_DESCRIPTION_LENGTH', 80);
55 define('PRODUCTS_PER_PAGE', 6);
```

## Update products displayed per page:

In the config file, the products displayed by page can be changed:

```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
config.php

46 // Server HTTP port (can omit if the default 80 is used)
47 define('HTTP_SERVER_PORT', '80');

48
49 /* Name of the virtual directory the site runs in, for example:
50 '/emart/' if the site runs at http://www.example.com/emart/
51 '' if the site runs at http://www.example.com/ */
52 define('VIRTUAL_LOCATION', '/emart/');

53
54 // Configure product lists display options
55 define('SHORT_PRODUCT_DESCRIPTION_LENGTH', 80);
56 define('PRODUCTS_PER_PAGE', 6);
```

## Access error logs:

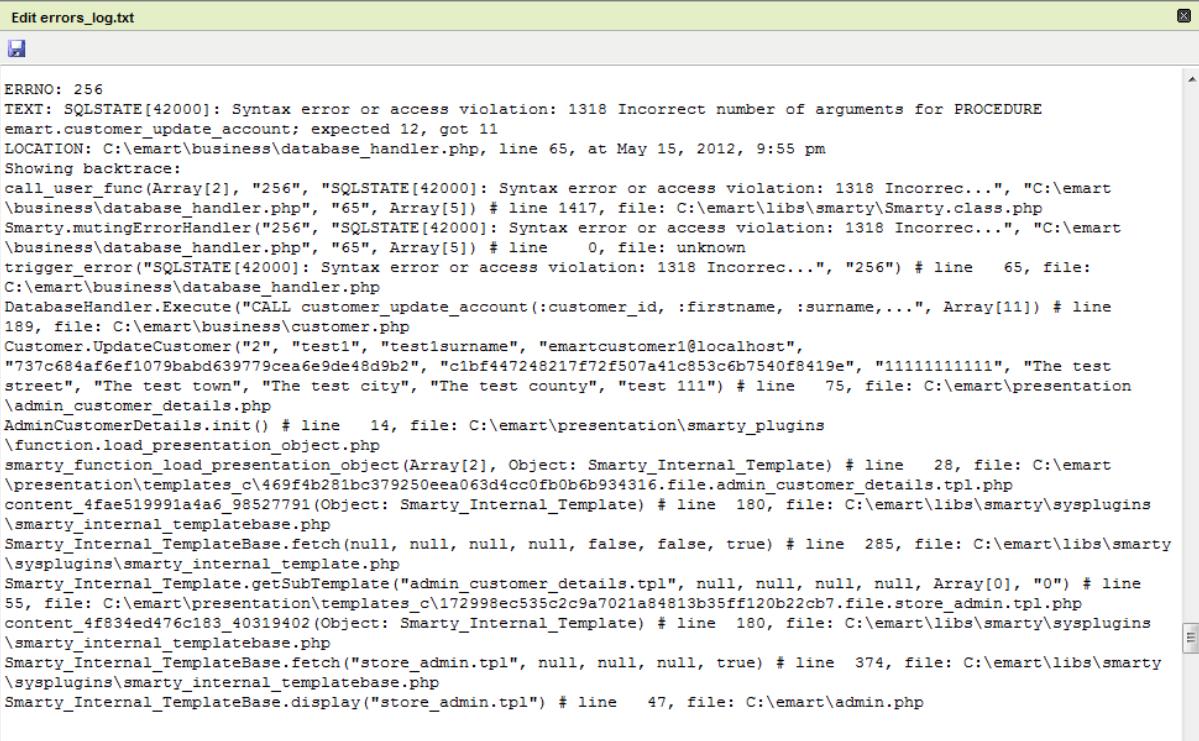
Root directory of your hosting account >>

<a href="#">Filename</a>	<a href="#">Size</a>	<a href="#">Date Modified</a>	<a href="#">File Type</a>	<a href="#">Permissions</a>
[Parent Directory]	--			
business	--	5/16/2012 5:22:24 pm	Directory	
images	--	5/16/2012 5:22:43 pm	Directory	
include	--	5/16/2012 9:25:53 pm	Directory	
libs	--	5/16/2012 5:22:46 pm	Directory	
logs	--	5/16/2012 5:23:15 pm	Directory	
presentation	--	5/17/2012 8:59:35 am	Directory	
product_images	--	5/16/2012 5:23:57 pm	Directory	
scripts	--	5/16/2012 5:23:58 pm	Directory	
styles	--	5/16/2012 5:23:59 pm	Directory	
.htaccess	2.45 KB	5/16/2012 5:24:00 pm	File	
404.php	1.01 KB	5/16/2012 5:24:00 pm	Script	
500.php	810 B	5/16/2012 5:24:00 pm	Script	
admin.php	1.61 KB	5/16/2012 5:24:00 pm	Script	
index.php	2.21 KB	5/16/2012 5:24:00 pm	Script	
logserrors_log.txt	21.6 KB	5/16/2012 8:19:10 pm	Text File	
test_authorize_net.php	1.67 KB	5/16/2012 5:24:01 pm	Script	
test_card.php	1.29 KB	5/16/2012 5:24:01 pm	Script	
test_encryption.php	623 B	5/16/2012 5:24:01 pm	Script	
test_hasher.php	647 B	5/16/2012 5:24:01 pm	Script	

logs >>

<a href="#">Filename</a>	<a href="#">Size</a>	<a href="#">Date Modified</a>	<a href="#">File Type</a>	<a href="#">Permissions</a>
[Parent Directory]				
errors_log.txt	58.1 KB	5/16/2012 5:23:15 pm	Text File	

## Open errors\_log.txt



```
ERRNO: 256
TEXT: SQLSTATE[42000]: Syntax error or access violation: 1318 Incorrect number of arguments for PROCEDURE emart.customer_update_account; expected 12, got 11
LOCATION: C:\emart\business\database_handler.php, line 65, at May 15, 2012, 9:55 pm
Showing backtrace:
call_user_func(Array[2], "256", "SQLSTATE[42000]: Syntax error or access violation: 1318 Incomplete... ", "C:\emart\business\database_handler.php", "65", Array[5]) # line 1417, file: C:\emart\libs\smarty\Smarty.class.php
Smarty.mutingErrorHandler("256", "SQLSTATE[42000]: Syntax error or access violation: 1318 Incomplete... ", "C:\emart\business\database_handler.php", "65", Array[5]) # line 0, file: unknown
trigger_error("SQLSTATE[42000]: Syntax error or access violation: 1318 Incomplete... ", "256") # line 65, file: C:\emart\business\database_handler.php
DatabaseHandler.Execute("CALL customer_update_account(:customer_id, :firstname, :surname,...", Array[11]) # line 189, file: C:\emart\business\customer.php
Customer.UpdateCustomer("2", "test1", "test1surname", "emartcustomer1@localhost",
"737c684aff6ef1079abd639779ceae6e9de48d9b2", "c1bf447248217f72f507a41c853c6b7540f8419e", "111111111111", "The test
street", "The test town", "The test city", "The test county", "test 111") # line 75, file: C:\emart\presentation
\admin_customer_details.php
AdminCustomerDetails.init() # line 14, file: C:\emart\presentation\smarty_plugins
\function.load_presentation_object.php
smarty_function_load_presentation_object(Array[2], Object: Smarty_Internal_Template) # line 28, file: C:\emart\presentation\templates_c\469f4b281bc379250eea063d4cc0fb0b6b934316.file.admin_customer_details.tpl.php
content_4fae519991a4a6_98527791(Object: Smarty_Internal_Template) # line 180, file: C:\emart\libs\smarty\sysplugins
\smarty_internal_templatebase.php
Smarty_Internal_TemplateBase.fetch(null, null, null, null, false, false, true) # line 285, file: C:\emart\libs\smarty\sysplugins\smarty_internal_template.php
Smarty_Internal_Template.getSubTemplate("admin_customer_details.tpl", null, null, null, null, Array[0], "0") # line 55,
file: C:\emart\presentation\templates_c\172998ec535c2c9a7021a84813b35ff120b22cb7.file.store_admin.tpl.php
content_4f834ed476c183_40319402(Object: Smarty_Internal_Template) # line 180, file: C:\emart\libs\smarty\sysplugins
\smarty_internal_templatebase.php
Smarty_Internal_TemplateBase.fetch("store_admin.tpl", null, null, null, true) # line 374, file: C:\emart\libs\smarty\sysplugins\smarty_internal_templatebase.php
Smarty_Internal_TemplateBase.display("store_admin.tpl") # line 47, file: C:\emart\admin.php
```

## Maintaining eMart

eMart holds a massive amount of information. The database contains not only information on the products which stocked but also sensitive information, e.g. customer credit card details. Therefore, it is extremely important that a strict and reliable backup strategy is in place. eMart's web hosting is with Godaddy who offer a range of hosting packages that support MySQL, these packages include full MySQL backup and optimisation options. eMart intend to make full use of this function to ensure the backup of their valuable data. A dependable backup adds insurance to the business. In the event that an incident occurs where data loss could be a factor, the company can return to a fully functional state quickly.

## **Appendix F**

# **SUMMARY OF CHANGES**

---

## Summary of Changes

During initial discussions with the client, eMart, the requirements document was written. Several changes were made to the specification during meetings following various stages of the design. Similar integrated revisions were made to the design specification during the implementation stage of the project. Here we detail any major changes made to the specification of the product.

*Please note: These revisions are not evident in the Requirements documentation (Appendix A) or Design documentation (Appendix B) since these changes represent future extensions that could be made available in subsequent releases of the product. Hence, use cases and any other relevant design sections may be useful at a later date.*

Initial revisions were made to the problem and solution to show a clearer view of what indeed the system needed to accomplish. After a review with the client it was determined that the system would now only cover Stock Management and eCommerce Management. During discussions all parties came to the conclusion that given the time sensitive nature of this project the HR Management and Accounts facilities could be completed as an entirely separate sub-system thus, could be contracted out to a separate development team, or possibly be included in a subsequent version of the product. Additionally, modifications were made to the requirements to mirror the fact that the system was no longer to provide multiple levels of authorisation on the back-end of the system. Use cases were added to facilitate users forgetting login credentials. A forgotten password feature was implemented.

As a result of the system no longer supporting HR Management and Accounts amendments were made to the set of use cases to reflect the removal of rota, payroll and account requirements. The product now utilises a custom made order processing facility instead of the specified PayPal gateway. This was the biggest amendment made transitioning from the design specification to the system realisation. Use cases and requirements were updated to support this modification.

### Added Use Cases:

Ref	Name	Reason
UC57	Forgotten Password	The system now provides users and customers the ability to retrieve lost or forgotten passwords.

### Redundant Use Cases:

Ref	Name	Reason
UC1	Create User	The system back-end no longer supports multiple levels of access.
UC3	Delete User	Same as above.
UC5	Access Transaction Log	The system no longer keeps a transaction log. This is made available in MySQL maintenance plan.
UC6	Add Employee	The system no longer supports rota and payroll processes.
UC7	Locate Employee	Same as above.
UC8	Update Employee	Same as above.
UC9	Delete Employee	Same as above.

<b>UC10</b>	Add Rota Data	Same as above.
<b>UC11</b>	Update Rota Data	Same as above.
<b>UC12</b>	Delete Rota Data	Same as above.
<b>UC13</b>	Show Rota Data	Same as above.
<b>UC14</b>	Run Payroll	Same as above.
<b>UC15</b>	Create Payroll Report	Same as above.
<b>UC43</b>	Fire Expiry Warning	It was determined that the system no longer tracks product expiry information.
<b>UC45</b>	Add Account	The system no longer supports account functionality.
<b>UC46</b>	Locate Account	Same as above.
<b>UC47</b>	Update Account	
<b>UC48</b>	Delete Account	

### Modified Use Cases:

Ref	Name	Reason
<b>UC26</b>		Amended to depict order placement and processing.
<b>UC27</b>		Amended to locate and view orders and history.

## **Appendix G**

# **MINUTES AND AGENDAS**

---

# AGENDA

<p><b>Agenda Meeting 1</b></p> <p><b>Date and Time :</b> 1<sup>st</sup> February 2012, 12pm</p> <p><b>Duration :</b> 2 hours</p> <p><b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Discuss and decide on idea</li> <li>• Determine requirements and set tasks for each member of the team</li> </ul>
<p><i>Topic/Item</i></p> <p>1. Brainstorm together to determine idea for project</p>	<p><i>Time Allocated</i></p> <p>30 minutes</p>
<p>2. Discuss requirements and delegate introduction and objectives to members of the team to complete.</p>	<p>1 hour</p> <p>Kirsty</p>
<p>3. Discuss what other requirements are important to be completed by next meeting and delegate tasks evenly to team.</p>	<p>30 minutes</p> <p>Kirsty</p>
<p>4.</p>	
<p>5.</p>	

<b>Meeting Duties</b>
<b>Chair :</b> Kirsty
<b>Secretary :</b> Sam
<b>Participants :</b> Sara, Carl

<b>Issues carried forward</b>
N/A

# Minutes

## Meeting 1

**Date and Time : 1<sup>st</sup> February 2012**

**Location : Common Room**

**Members Present : Carl, Sam, Sara, Kirsty**

**Members Absent : None**

Purpose of Meeting :

- Discuss and decide on idea
- Determine requirements and set tasks for each member of the team

Item	Issue	By When?	By Whom?
1.	By brainstorming together we finalised on the idea of an online convenience store and database for stock management called E-Mart.		
2.	From reading our specification Sam noted that the first step in our planning was to write an introduction. It was agreed Kirsty would write the introduction.	Next meeting	Kirsty
3.	Also a mission statement and objectives needed to be written ASAP, to show the purpose of the project. Sam agreed to do this.	Next meeting	Sam
4.	Kirsty stated that it was important for the diagrams of the tables and the boundaries of the system/user views to be created before moving any further. Sara and Carl agreed to collaborate together on this.	Next meeting	Sara and Carl
5.	Sara suggested that the best way to determine the requirements of the company was to write a transcript of an interview with the manager of the company. Sam agreed to do this.	Next meeting	Sam
6.	Carl stated that time management was important for us to complete the project to a high standard, and with the use of a Gantt chart this would help aide us to see our targets and ensure we make them. As Kirsty has previous experience using Gantt charts she agreed to do this.	Next meeting	Kirsty

Members invited : All

Date & Time : Monday 6<sup>th</sup> February 2012

Location : Common Room

# AGENDA

<p><b>Agenda Meeting 2</b></p> <p><b>Date and Time :</b> 6<sup>st</sup> February 2012, 11pm</p> <p><b>Duration :</b> 1.5 hours</p> <p><b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Look over and discuss completed tasks</li> <li>• Assign tasks left from looking at completed Gantt chart</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase our work completed so far and discuss improvements if any.	30 minutes	All
2. Use Gantt chart created to assign tasks left to be completed	50 minutes	Sara
3. Discuss any other issues	10 minutes	Sara
4.		
5.		

Meeting Duties
Chair : Sara
Secretary : Carl
Participants : Kirsty, Sam

Issues carried forward
None

# Minutes

<p><b>Meeting 2</b></p> <p><b>Date and Time : 6<sup>th</sup> February 2012</b></p> <p><b>Location : Guild of Students</b></p> <p><b>Members Present : Carl, Sam, Sara, Kirsty</b></p> <p><b>Members Absent : None</b></p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Look over and discuss completed tasks</li> <li>• Assign tasks left on Gantt chart for Phase 1</li> </ul>
---	---

Item	Issue	By When?	By Whom?
1.	Discussed and amended tasks done this week. Discussed todays lecture and any tasks we think we may have overlooked. Decided on a font to use consistently through the project		
2.	From todays lecture we need to research any issues concerning eMart and the DPA. Sara has experience in researching the DPA so she has agreed to do this	Next meeting	Sara
3.	From the Gantt chart, Carl pointed out that stakeholders need to be identified to see who has access to the system and how much access they can have	Next meeting	Carl
4.	After looking at the Gantt chart we need to do the use case descriptions and the actors and their goals.	Next meeting	Kirsty
5.	After completing the above task, the use case diagrams need to be completed	Next meeting	Sara
6.	After 3 & 4 are completed, the system requirements will be done. Whilst waiting for the above task to be completed, research will be done on how to set the system requirements out.	Next meeting	Carl
7.	Problem domain research is needed to get an idea on what operations are needed for the system	Next meeting	Sam
8.	From the Gantt chart, in order to finish off the functional requirements, the non-functional requirements will be done	Next meeting	Sam
9.	No further issues were reported		

Next meeting
<p>Members Invited : All</p> <p>Date &amp; Time : Monday 13<sup>th</sup> February 2012</p> <p>Location : Guild of Students</p>

# AGENDA

<p><b><u>Agenda Meeting 3</u></b></p> <p><b>Date and Time :</b> 13<sup>st</sup> February 2012, 11pm</p> <p><b>Duration :</b> 1 hour</p> <p><b>Location :</b> Lab 1</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Proof read report so far</li> <li>• See what is incomplete on the Gantt chart</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase our work completed so far and discuss improvements if any.	15 minutes	All
2. Use Gantt chart created to assign tasks left to be completed	20 minutes	Sam
3. Discuss any other issues	25 minutes	All
4.		
5.		

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Sam</p> <p><b>Secretary :</b> Sara</p> <p><b>Participants :</b> Carl</p>	<p>Discuss layout of system requirements</p>

# Minutes

<p><b>Meeting 3</b>  <b>Date and Time : 13<sup>th</sup> February 2012, 11pm</b>  <b>Location : Lab 1</b>  <b>Members Present : Carl, Sam, Sara</b>  <b>Members Absent :Kirsty</b></p>	<p>Purpose of Meeting:</p> <ul style="list-style-type: none"> <li>• Proof read report so far</li> <li>• See what else is incomplete on the Gantt chart</li> </ul>
---	---

Item	Issue	By When?	By Whom?
1.	<p>Kirsty apologises for being absent. She is very ill.</p> <p>After reading the report so far, we realised that some of the system requirements have been included in other parts so we decided to put all of them into one document.</p> <p>Carl agreed to do this.</p>	Next Meeting	Carl
2.	<p>Sara noted that we needed to create the final functional requirements now the use case diagrams and descriptions are completed.</p> <p>Sara stated she felt confident enough with her knowledge to create these.</p>	Next meeting	Sara
3.	<p>For the presentation, Sara suggested to create a logo to maintain ethos throughout the project. We all are going to brainstorm and compare ideas in the next meeting.</p>	Next meeting	Everyone
4.	<p>Sam reported that she finished the non functional requirements, however upon overlooking them yesterday realised she had not saved them. They will be completed and uploaded to BaseCampHQ by tomorrow evening.</p>	Next meeting	Sam
5.	<p>A final report shall be created using all of the separate documents to create the report which shall be handed in.</p> <p>We all at the meeting agreed to allocate this task to Kirsty.</p>	Next meeting	Kirsty
6.	<p>Minutes shall be forwarded to Kirsty upon completion.</p>	Next meeting	Sara

<b>Next meeting</b>
<p>Members Invited : All</p> <p>Date &amp; Time : Thursday 16<sup>th</sup> February 2012, 1.30 pm</p> <p>Location : Guild of Students</p>

# AGENDA

<p><b><u>Agenda Meeting 4</u></b></p> <p><b>Date and Time :</b> 15<sup>th</sup> February 2012, 11pm</p> <p><b>Duration :</b> 1.5 hour</p> <p><b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p><b>Purpose of Meeting :</b></p> <ul style="list-style-type: none"> <li>• Make sure report is finished</li> <li>• Organize presentation</li> </ul>
--	--

<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase our work completed from last meeting.	21 minutes	All
2. Look at Gantt chart to check all tasks have been completed	20 minutes	Carl
3. Assign somebody to put the report together to create the final version	10 minutes	All
4. Talk about presentation.	40 minutes	Carl
5. Discuss any other issues.	10 minutes	Carl

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Carl</p> <p><b>Secretary :</b> Kirsty</p> <p><b>Participants :</b> Sam, Sara</p>	<p>None</p>

# Minutes

**Meeting 4**

**Date and Time : 15<sup>th</sup> February 2012**

**Location : Common Room**

**Members Present : Carl, Sam, Sara, Kirsty**

**Members Absent : None**

Purpose of Meeting:

- Make sure report is finished
- Organize presentation

Item	Issue	By When?	By Whom?
1.	The report has been put together, and the non functional and functional requirements have been finished and proof read. All of our work is now together and finished in a final report.		
2.	Before submitting the report, a contents page and numbering needs to be created. Kirsty and Carl have agreed to collaborate and work together to finalise the presentation of the final report.	Next meeting	Kirsty and Carl
3.	We discussed who would like to lead our presentation and Sara said she felt confident enough to do this. She has agreed to put together an example transcript and practice the presentation.	Next meeting	Sara
4.	For the presentation Carl thought it'd be a good idea to create a PowerPoint presentation to showcase and bullet point for the audience. Sam has agreed to do this.	Next meeting	Sam
5.	Kirsty suggested to meet for a mock presentation before the end of the week. She is going to book a study room in the Sydney Jones library for Friday afternoon.	End of the day	Kirsty
6.			

**Next meeting**

Members Invited : All

Date & Time : Monday 21<sup>th</sup> February  
2012

Location : Guild of Students

# AGENDA

<p><b><u>Agenda Meeting 5</u></b>  <b>Date and Time :</b> 21<sup>st</sup> February 2012, 11pm  <b>Duration :</b> 2 hours  <b>Location :</b> Guild of Students  <b>Members invited :</b> All</p>	<p>Purpose of Meeting :  <ul style="list-style-type: none"> <li>• Discuss design phase</li> <li>• Look at Gantt chart and assign tasks</li> </ul> </p>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Generally discuss what we would all like to achieve by the end of the design phase.	20 minutes	Sara
2. Look at Gantt chart to check all tasks that need to be completed are on it.	20 minutes	Sara
3. Discuss initial steps and assign tasks for next meeting.	1.2 minutes	All
4.		
5.		

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Sara  <b>Secretary :</b> Sam  <b>Participants :</b> Kirsty, Carl</p>	<p>None</p>

# Minutes

<p><b>Meeting 5</b>  <b>Date and Time : 21<sup>st</sup> February 2012</b>  <b>Location : Guild of Students</b>  <b>Members Present : Carl, Sam, Sara, Kirsty</b>  <b>Members Absent : None</b></p>	<p>Purpose of Meeting:</p> <ul style="list-style-type: none"> <li>• Discuss design phase</li> <li>• Look at Gantt chart and assign tasks</li> </ul>
--	---

Item	Issue	By When?	By Whom?
1.	We read through the specification, Gantt chart and the recommended text book together and started to speak about what steps as a team we need to take.		
2.	From the database book Kirsty noted that the first step is creating ER Models. Carl has agreed to create the models and Sara has agreed to check over them once created.	ASAP	Sara and Carl
3.	The dataflow diagram is extremely important as it shows what data is input and output from the system. Sam has also agreed to research the business rules. Sam has agreed to do this.	Next meeting	Sam
4.	Carl noted that in the book the table structures need to be implemented. Sara has agreed to look at these, and do as much as she can on them. Then will then be uploaded to BaseCamp and we will all look through them and add to them.	Next meeting	Sara initially, and everyone
5.	The data dictionary needs to be created and continually added to throughout the design. Sara has agreed to start this initially.	Next meeting	Sara
6.	Kirsty has suggested she shall start working on the implementation. She will create the sketches and start work on the interface.	Next meeting	Kirsty
7.	We received feedback from our project manager after our presentation that our project may be a bit too optimistic and we should consider getting rid of the HR Management section of our database. After discussing this we have agreed and our removing our plans to create a payroll and rota system. From now on all documents and planning shall not mention payroll or rotas.		All
8.	We all mentioned our concern about the amount of work from this subject and other subjects and the little amount of time we have to complete this section. We have all agreed to just do as much as we can and ensure we don't fall behind in other modules.		

## Next meeting

Members Invited : All  
Date & Time : Monday 28<sup>th</sup> February 2012  
Location : Common Room

# AGENDA

<p><b><u>Agenda Meeting 6</u></b>  <b>Date and Time :</b> 28<sup>th</sup> February 2012, 11pm  <b>Duration :</b> 1 hour  <b>Location :</b> Guild of Students  <b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• See how everyone has progressed from last meeting.</li> <li>• Assign tasks left to do</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase our work completed so far and discuss improvements if any.	15 minutes	Kirsty
2. Discuss what else needs doing and prioritise work left	20 minutes	Kirsty
3. Discuss any other issues	25 minutes	All
4.		
5.		

Meeting Duties
<b>Chair :</b> Kirsty
<b>Secretary :</b> Carl
<b>Participants :</b> Sam Sara

Issues carried forward
None

# Minutes

## Meeting 6

Date and Time : 28<sup>st</sup> February 2012

Location : Guild of Students

Members Present : Carl, Sam, Sara, Kirsty

Members Absent : None

Purpose of Meeting:

- Discuss design phase
- Look at Gantt chart and assign tasks

Item	Issue	By When?	By Whom?
1.	Carl has drawn up a rough sketch for the ER diagram showing all of the relationships between entities he has thought up so far. Sara has agreed to take over and finalise the ER diagram and create a final copy using SmartDraw. She has agreed to put a final copy up by the end of the week on BaseCamp.	2/03/2012	Sara
2.	Sara has stated that in the past week she has made a good attempt at the data dictionary and included as much as she can think of so far. She has uploaded this to BaseCamp and we will all add to it when necessary.	Continually	All
3.	Kirsty stated that she thinks it will be a good idea to create activity diagrams to show examples of user activities. Carl has agreed to do these.	Next meeting	Carl
4.	Sam has been continuing to look at examples to create the business rules. She has stated that these are well underway and should hopefully be finished by the next meeting	Next meeting	Sam
5.	Kirsty said she has finished the transaction matrix and is halfway through the physical database design. She stated that these will both be completed by the next meeting.	Next meeting	Kirsty
6.	We all agreed that we were doing well with the workload(as we had previously stated we were worried about falling behind In other modules) and this sort of work pace shall continue and we will all feel comfortable about getting the work completed to a good standard on time.		
7.			
8.			

## Next meeting

Members Invited : All

Date & Time : Tuesday 6<sup>th</sup> March 2012

Location : Common Room

# AGENDA

<p><b><u>Agenda Meeting 7</u></b>  <b>Date and Time :</b> 6<sup>th</sup> March 2012, 1pm  <b>Duration :</b> 45 minutes  <b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Start to put report together</li> <li>• Source missing work</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase work from previous meeting and discuss if any errors are noted.	15 minutes	Carl
2. Discuss how report should be structured.	20 minutes	Carl
3. Discuss any other issues.	25 minutes	All
4.		
5.		

Meeting Duties
Chair : Carl
Secretary : Sara
Participants : Kirsty, Sam

Issues carried forward
None

# Minutes

## Meeting 7

**Date and Time : 6th March 2012**

**Location : Common Room**

**Members Present : Carl, Sam, Sara, Kirsty**

**Members Absent : None**

Purpose of Meeting:

- Discuss design phase
- Look at Gantt chart and assign tasks

Item	Issue	By When?	By Whom?
1.	Sara has uploaded the complete ER diagram and every member of the team has confirmed they are happy with the final ER diagram.	-	
2.	Sam has completed the business rules and uploaded them for confirmation off the rest of the team. Carl has agreed to proof read these and add any if he deems necessary.	13 <sup>th</sup> March	Carl
3.	Sam has also made a start on the sequence diagrams and will finish this by the end of the week so the team can evaluate them before the next meeting.	10 <sup>th</sup> March	Everyone
4.	Sara has identified the table structures still need to be done and will start work on these immediately.	Next meeting	Sara
5.	Kirsty has stated that she would like to now decide the design of the website and will create sketches and ideas for the layout and interface as well as identify the target audience.	Next meeting	Kirsty
6.			
7.			
8.			

## **Next meeting**

**Members Invited : All**

**Date & Time : Wednesday 14<sup>th</sup> March 2012**

**Location : Lab 2**

# AGENDA

<p><b>Agenda Meeting 8</b></p> <p><b>Date and Time :</b> 6<sup>th</sup> March 2012, 1pm</p> <p><b>Duration :</b> 45 minutes</p> <p><b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Start to put report together</li> <li>• Source missing work</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Showcase work from previous meeting and discuss if any errors are noted.	15 minutes	Carl
2. Discuss how report should be structured.	20 minutes	Carl
3. Discuss any other issues.	25 minutes	All
4.		
5.		

Meeting Duties
Chair : Carl
Secretary : Sara
Participants : Kirsty, Sam

Issues carried forward
None

# Minutes

## Meeting 8

Date and Time : 14<sup>th</sup> March 2012

Location : Lab 2

Members Present : Sam, Sara, Kirsty

Members Absent : Carl

Purpose of Meeting:

- Discuss finalising phase

Item	Issue	By When?	By Whom?
1.	Carl apologises for being absent.		
2.	Everything has now been finished and we discussed any final amendments that need to be done before the final report is put together.		
3.	Kirsty said the sequence diagrams need amending. We all discussed what needs to be amended; there is no need to have one for every single use case and so only important use cases need to be finalised and have the data flow defined more in the diagram.  Sam has agreed to do this and will have them uploaded to drop box by the end of the day  No other issues were noted.	End of the day	Sam
4.	The report now needs to be created to hand in. Kirsty and Sara have agreed to do this together and will hand it in on Friday 23 <sup>rd</sup> March.	23 <sup>rd</sup> March	Kirsty and Sara
5.			
6.			
7.			
8.			

## Next meeting

Members Invited : All

Date & Time : Tuesday 21<sup>st</sup> March 2012

Location : Common Room

# AGENDA

<p><b>Agenda Meeting 9</b></p> <p><b>Date and Time :</b> 21<sup>st</sup> March 2012, 1pm</p> <p><b>Duration :</b> 30 minutes</p> <p><b>Location :</b> Common Room</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Plan for presentation</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Discuss what shall be included in the presentation.	15 minutes	Sam
2. Discuss any questions that need to be asked.	15 minutes	Sam
3.		
4.		
5.		

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Sam</p> <p><b>Secretary :</b> Sara</p> <p><b>Participants :</b> Kirsty</p>	<p>None</p>

# Minutes

## Meeting 9

**Date and Time : 21st March 2012**

**Location : Lab 2**

**Members Present : Sam, Sara, Kirsty**

**Members Absent : None**

Purpose of Meeting:

- Plan for presentation

Item	Issue	By When?	By Whom?
1.	<p>Carl has unfortunately dropped out of university. We all would like to thank him for all of the contribution he has put into the project thus far and he will be missed.</p> <p>Although we have lost a team member we are all confident going forward that we can complete the project on time and to our best capability.</p>		
2.	The presentation has been booked for 23 <sup>rd</sup> March. Kirsty has stated that because of a test on that day she will be unable to attend the presentation so Sara and Sam shall do it.		
3.	Sara and Sam shall meet on Thursday to practise and go over the presentation and prepare any questions we may have.	Thursday	Sam and Sara
4.			
5.			
6.			
7.			
8.			

## Next meeting

Members Invited : All

Date & Time : Friday 30<sup>th</sup> March 2012

Location : Guild of Students

# AGENDA

## Agenda Meeting 10

**Date and Time : 30<sup>th</sup> March 2012, 11pm**

**Duration : 45 mins**

**Location : Guild of Students**

**Members invited : All**

**Purpose of Meeting :**

- Plan for implementation

1. Discuss implementation,	15 minutes	Sara
2. Discuss other work required.	20 minutes	Sara
3. Any other issues.	10 minutes	All
4.		
5.		

## **Meeting Duties**

**Chair : Sara**

**Secretary : Kirsty**

**Participants : Sam**

## **Issues carried forward**

None

# Minutes

<p><b><u>Meeting 10</u></b></p> <p><b>Date and Time :</b> 30th March 2012</p> <p><b>Location :</b> Guild of Students</p> <p><b>Members Present :</b> Sam, Sara, Kirsty</p> <p><b>Members Absent :</b> None</p>	<p>Purpose of Meeting:</p> <ul style="list-style-type: none"> <li>• Plan for implementation</li> </ul>
--	--

Item	Issue	By When?	By Whom?
1.	Implementation will now begin. Kirsty has agreed to do this over the Easter break and we will all discuss progress after.	Easter	Kirsty
2.	After implementation work for testing, user guide and the report will need to be done. Sara and Sam have agreed to do the bulk of this.		Sara and Sam
3.	No other issues were reported.		
4.			
5.			
7.			
8.			

## Next meeting

Members Invited : All  
 Date & Time : 20/04/2012  
 Location : Student Guild

# AGENDA

**Agenda Meeting 11**

**Date and Time : 20<sup>th</sup> April 2012, 11pm**

**Duration : 45 mins**

**Location : Guild of Students**

**Members invited : All**

**Purpose of Meeting :**

- Check on progress of implementation

<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Discuss implementation thus far	15 minutes	Sam
2. Discuss other work required.	20 minutes	Sam
3. Any other issues.	10 minutes	All
4.		
5.		

**Meeting Duties**

**Chair : Sam**

**Secretary : Sara**

**Participants : Kirsty**

**Issues carried forward**

None

# Minutes

## Meeting 11

Date and Time : 20th April 2012

Location : Guild of Students

Members Present : Sam, Sara, Kirsty

Members Absent : None

Purpose of Meeting:

- Check on progress of implementation

Item	Issue	By When?	By Whom?
1.	Kirsty has reported that implementation is going on time and as planned. No issues or complications have occurred thus far.  She will continue to progress and be able to show Sam and Sara what is done so far at the next meeting.	Next meeting	Kirsty
2.	Sara supplied sample data for the database over Easter to aid implementation.		
3.	No other issues were reported.		
4.			
5.			
7.			
8.			

## **Next meeting**

Members Invited : All

Date & Time : 27/04/2012

Location : Student Guild

# AGENDA

<p><b>Agenda Meeting 12</b></p> <p><b>Date and Time :</b> 27<sup>th</sup> April 2012, 11pm</p> <p><b>Duration :</b> 45 mins</p> <p><b>Location :</b> Guild of Students</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Plan for implementation</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Discuss implementation progress and get feedback off the rest of the group.	15 minutes	Kirsty
2. Discuss other work required to begin and research.	20 minutes	Kirsty
3. Any other issues.	10 minutes	All
4.		
5.		

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Kirsty</p> <p><b>Secretary :</b> Sam</p> <p><b>Participants :</b> Sara</p>	<p>None</p>

# Minutes

## Meeting 12

Date and Time : 27th April 2012

Location : Guild of Students

Members Present : Sam, Sara, Kirsty

Members Absent : None

Purpose of Meeting:

- Check on progress of implementation

Item	Issue	By When?	By Whom?
1.	Kirsty showcased the website thus far to Sam and Sara who are extremely impressed. No changes were noted and Kirsty will continue to complete implementation. She wishes to have it 90 % complete by the next meeting.	Next meeting	Kirsty
2.	Sam and Sara will now begin researching the final report, user guide and test document and make templates for these documents	Next meeting	Sam and Sara
3.	No other issues were reported		
4.			
5.			
7.			
8.			

## **Next meeting**

Members Invited : All

Date & Time : 4/05/2012

Location : Student Guild

# AGENDA

<p><b>Agenda Meeting 13</b></p> <p><b>Date and Time :</b> 4<sup>th</sup> May 2012, 11pm</p> <p><b>Duration :</b> 45 mins</p> <p><b>Location :</b> Guild of Students</p> <p><b>Members invited :</b> All</p>	<p>Purpose of Meeting :</p> <ul style="list-style-type: none"> <li>• Plan for implementation</li> </ul>	
<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Discuss and observe final product and any last minute changes that need to occur.	15 minutes	Sara
2. Delegate and start report documentation.	20 minutes	Sara
3. Any other issues.	10 minutes	All
4.		
5.		

<b>Meeting Duties</b>	<b>Issues carried forward</b>
<p><b>Chair :</b> Sara</p> <p><b>Secretary :</b> Kirsty</p> <p><b>Participants :</b> Sam</p>	<p>None</p>

# Minutes

<p><b>Meeting 13</b></p> <p><b>Date and Time : 4th April 2012</b></p> <p><b>Location : Guild of Students</b></p> <p><b>Members Present : Sam, Sara, Kirsty</b></p> <p><b>Members Absent : None</b></p>	<p>Purpose of Meeting:</p> <ul style="list-style-type: none"> <li>•</li> </ul>
--	--

Item	Issue	By When?	By Whom?
1.	<p>The website has been finished and Kirsty did a thorough demonstration to the team.</p> <p>Everybody is exceptionally happy! Kirsty will now comment the code ready for hand in.</p>	Next meeting	Kirsty
2.	<p>The user guide now needs to be created. Kirsty has uploaded the necessary files to Dropbox for Sara to implement the website on her own laptop. She will use this to create the user guide.</p>	Next meeting	Sara
3.	<p>The report needs to be started. Sam will use the template she created to do as much as she can.</p> <p>She will also make a start on the testing documentation.</p>	Next meeting	Sam
4.			
5.			
7.			
8.			

## Next meeting

Members Invited : All  
 Date & Time : 11/05/2012  
 Location : Student Guild

# AGENDA

**Agenda Meeting 13**

**Date and Time :** 11<sup>th</sup> May 2012, 11pm

**Duration :** 15 mins

**Location :** Guild of Students

**Members invited :** All

**Purpose of Meeting :**

- Plan for implementation

<i>Topic/Item</i>	<i>Time Allocated</i>	<i>Topic Leader</i>
1. Discuss and decide on plan for demo presentation.	15 minutes	Sam
2. Progress on user guide.	10 minutes	Sam
3. Progress on report.	10 minutes	Sam
4. Progress on testing document	10 minutes	Sam
5. Any other issues	15 minutes	All

**Meeting Duties**

**Chair :** Sam

**Secretary :** Sara

**Participants :** Kirsty

**Issues carried forward**

None

# Minutes

## Meeting 14

Date and Time : 11th April 2012

Location : Guild of Students

Members Present : Sam, Sara, Kirsty

Members Absent : None

Purpose of Meeting:

•

Item	Issue	By When?	By Whom?
1.	<p>The demo presentation has been booked for Wednesday. All team members feel confident talking about the final product. As there has been a struggle hosting the website it shall be done using the local host on Kirsty's laptop.</p> <p>A plan of action has been created listing the important aspects to be showcased during the demo. Kirsty has agreed to lead the demo as she feels the most confident using the system.</p> <p>Every team member needs to ensure they are in for the presentation on Wednesday 16<sup>th</sup>, 10pm.</p>	Wednesday	All
2.	Sara reported that the user guide is now complete. Kirsty has checked the user guide and is happy with it.		
3.	The report is now nearly complete. The strengths and weaknesses need to be finalised and Kirsty will finish these.	Wednesday	Kirsty
4.	The testing document needs to be finished. Sara and Sam are going to ask for feedback on what needs to be included in this, split the work load and complete it by Thursday ready for hand in the day after.	Thursday	Sara and Sam
5.	<p>A message will be posted to the Facebook group signifying any other issues that may occur between now and the hand in date; as this is the last meeting.</p> <p>It is each member's responsibility to keep checking between now and then to make sure they do not miss any important information.</p> <p>If nothing is heard Kirsty and Sara will hand in the final documentation on Friday 18<sup>th</sup> May.</p>	Friday 18 <sup>th</sup> May	
6.			
7.			
8.			

## Next meeting

Members Invited : All

Date & Time : 11/05/2012

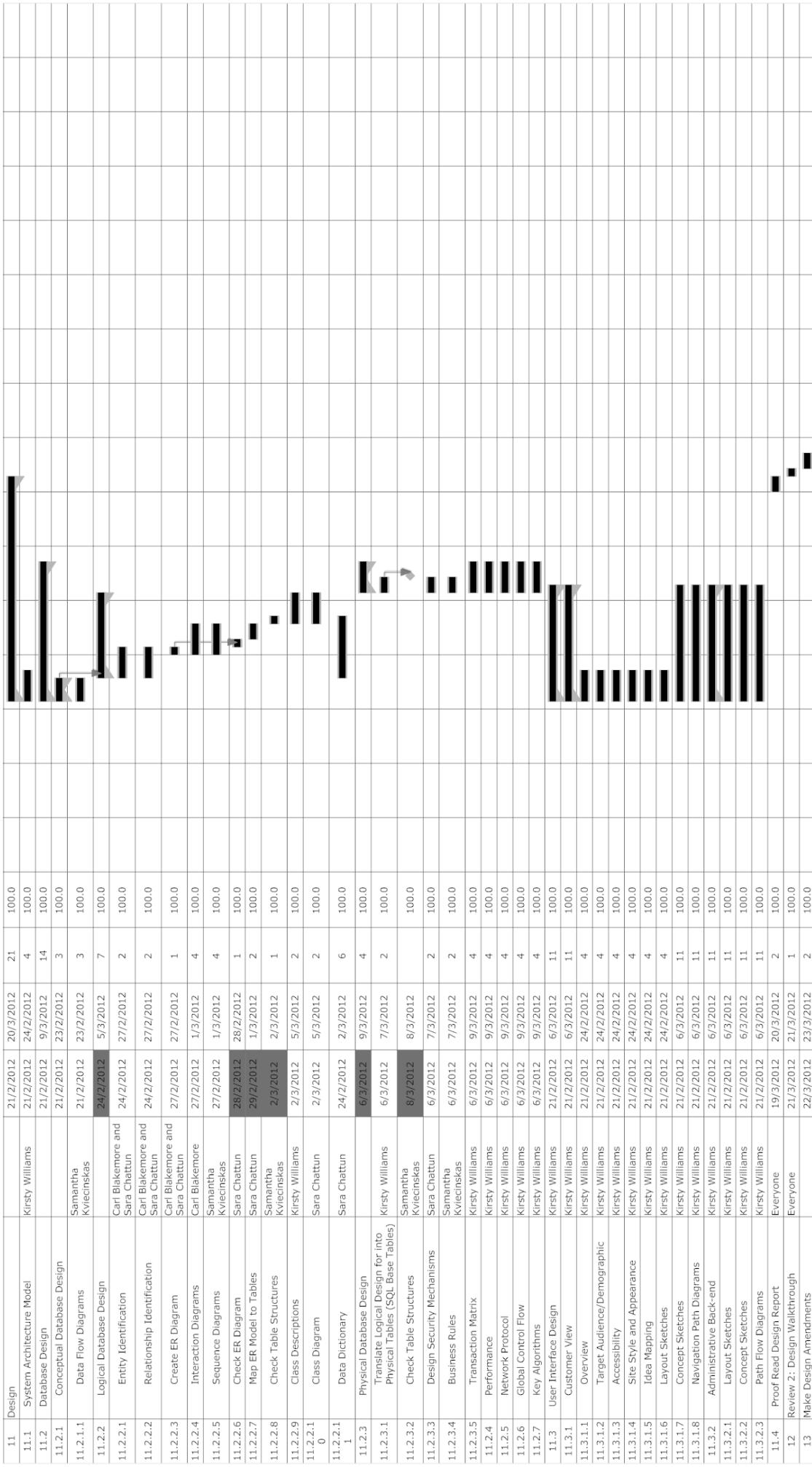
Location : Student Guild

## Appendix H

# GANNT CHART

---

Number	Task	Resource	Start	End	Duration	% Complete	30/1	6/2	13/2	20/2	27/2	5/3	12/3	19/3	26/3	2/4	9/4	16/4	23/4	30/4	7/5	14/5
1	Brainstorming/Research	Everyone	30/1/2012	1/2/2012	3	100.0																
2	Gantt Chart	Kirsty Williams	30/1/2012	31/1/2012	2	100.0																
3	Introduction		1/2/2012	3/2/2012	3	100.0																
3.1	Project Statement	Kirsty Williams	1/2/2012	3/2/2012	3	100.0																
3.2	Feasibility Study	Kirsty Williams	1/2/2012	3/2/2012	3	100.0																
3.3	Governing Standards	Sara Chatton	1/2/2012	3/2/2012	3	100.0																
4	Database Planning		1/2/2012	3/2/2012	3	100.0																
4.1	Mission Statement	Samantha Kwiecińska	1/2/2012	3/2/2012	3	100.0																
4.2	Mission Objectives	Samantha Kwiecińska	1/2/2012	3/2/2012	3	100.0																
5	System Definition		1/2/2012	3/2/2012	3	100.0																
5.1	Candidate Class Identification	Sara Chatton	1/2/2012	2/2/2012	2	100.0																
5.2	System Scope and Boundaries	Carl Blakemore	3/2/2012	3/2/2012	1	100.0																
5.3	User Views	Carl Blakemore	3/2/2012	3/2/2012	1	100.0																
6	Requirements Collection and Analysis		6/2/2012	17/2/2012	10	100.0																
6.1	Fact Finding		6/2/2012	7/2/2012	2	100.0																
6.1.1	Interview Transcripts	Samantha Kwiecińska	6/2/2012	7/2/2012	2	100.0																
6.1.2	eCommerce Walkthrough	Kirsty Williams	6/2/2012	7/2/2012	2	100.0																
6.1.3	Domain Research (i.e. understanding tasks completed in stock management and payroll)	Samantha Kwiecińska	6/2/2012	7/2/2012	2	100.0																
6.1.4	Organisational Structure	Kirsty Williams	6/2/2012	7/2/2012	2	100.0																
6.1.5	Context Diagram	Kirsty Williams	6/2/2012	7/2/2012	2	100.0																
6.2	Analysis		8/2/2012	17/2/2012	8	100.0																
6.2.1	Stakeholders	Kirsty Williams	8/2/2012	8/2/2012	1	100.0																
6.2.2	Actors and Goals	Kirsty Williams	8/2/2012	8/2/2012	1	100.0																
6.2.3	Use Cases		9/2/2012	15/2/2012	5	100.0																
6.2.3.1	Descriptions	Kirsty Williams	9/2/2012	13/2/2012	3	100.0																
6.2.3.2	Diagrams	Sara Chatton	14/2/2012	15/2/2012	2	100.0																
6.2.4	Transaction Requirements	Sara Chatton	9/2/2012	10/2/2012	2	100.0																
6.2.5	System Specification	Carl Blakemore	15/2/2012	16/2/2012	2	100.0																
6.2.6	Functional Requirements	Sara Chatton	15/2/2012	16/2/2012	2	100.0																
6.2.7	Non-Functional Requirements	Samantha Kwiecińska	9/2/2012	16/2/2012	6	100.0																
6.2.8	Prioritisation of Requirements	Kirsty Williams	17/2/2012	17/2/2012	1	100.0																
6.2.9	Success Criteria	Kirsty Williams	17/2/2012	17/2/2012	1	100.0																
6.3	Report Proof Read	Everyone	17/2/2012	17/2/2012	1	100.0																
7	Powerpoint Presentation	Samantha Kwiecińska	16/2/2012	17/2/2012	2	100.0																
8	Presentation Content	Everyone	16/2/2012	17/2/2012	2	100.0																
9	Review 1: Requirements Walkthrough	Everyone	20/2/2012	20/2/2012	1	100.0																
10	Summary of Changes	Kirsty Williams	21/2/2012	22/2/2012	2	100.0																



14	Implementation	Kirsty Williams	23/3/2012	10/5/2012	32	100.0
14.1	Create template PSD files	Kirsty Williams	23/3/2012	23/3/2012	1	100.0
14.2	Implement Database in MySQL	Kirsty Williams	23/3/2012	23/3/2012	1	100.0
14.3	Install Database on GoDaddy Server	Kirsty Williams	25/3/2012	26/3/2012	1	100.0
14.4	Implement Administrative Back-end	Kirsty Williams	26/3/2012	16/4/2012	14	100.0
14.5	Implement Storefront	Kirsty Williams	16/4/2012	10/5/2012	18	100.0
15	Testing	Sara Chattan and Samantha Kvieciukas	5/5/2012	14/5/2012	5	100.0
16	User Documentation	Sara Chattan	15/5/2012	15/5/2012	1	100.0
17	Installation Instructions	Kirsty Williams	11/5/2012	11/5/2012	1	100.0
18	Application Description	Samantha Kvieciukas	15/5/2012	15/5/2012	1	100.0
19	System Achievements	Kirsty Williams	13/5/2012	15/5/2012	2	100.0
20	Screen Shots	Sara Chattan	16/5/2012	16/5/2012	1	100.0
21	Realisation	Kirsty Williams	16/5/2012	16/5/2012	1	100.0
22	Professional Issues	Samantha Kvieciukas	16/5/2012	17/5/2012	2	100.0
23	Evaluation	Everyone	16/5/2012	17/5/2012	2	100.0
24	Extensions	Sara Chattan	16/5/2012	17/5/2012	2	100.0
25	Summary of changes	Kirsty Williams	17/5/2012	17/5/2012	1	100.0
26	Bibliography	Kirsty Williams	17/5/2012	17/5/2012	1	100.0
27	Review 3: Demonstration	Everyone	16/5/2012	16/5/2012	1	100.0
28	Complete Portfolio	Everyone	17/5/2012	17/5/2012	1	100.0
29	Portfolio Submissions	Everyone	18/5/2012	18/5/2012	1	100.0