

Business Name: My Vitality



Contents

Assignment 1.....	5
Client	5
Business Name	5
Description of client.....	5
Description of the proposed system.....	5
Scope of the proposed system.....	6
Community.....	7
Who is the community?	7
Why this community?	7
Describe the community.....	7
Why do the community needs this software system?	8
Assignment 2.....	9
Preliminary Literature Review	9
Introduction	9
Justification	9
Past and current research [systems] in the community area.....	10
How does the system fit in the community?	10
Development: what approach will be followed for the system?.....	12
How will the system be tested?.....	12
What would be the scope for future development for this system?	13
Reference list	13
Assignment 3.....	14
System Proposal.....	14

Outcomes.....	14
Objectives	14
Outline deliverables	16
Scope.....	17
Approach.....	19
Exclusions.....	25
Constraints	26
Outline business case.....	26
Reasons for selecting this solution for the system	30
Outline project plan	31
Resources.....	32
Quality expectations	33
Risk assessment	34
Assignment 4.....	36
Database	36
ERD	36
Development Database.....	38
Screen shots of the database.....	58
Assignment 5.....	66
MIS Reporting	69
Query writing	69
Assignment 6.....	94
Analysis and Design Documentation	94
Screenshots.....	94
Dataflow diagrams	104

UML's diagrams.....	109
Implementation Methodology.....	113
Implementation and Requirements document	117
Requirements.....	117
Technical performance	118
In serves support.....	119
Security	120
Backup procedure	121
Testing and evaluation of system	122
Implementation	125
Assignment 7.....	130
User manual.....	130
Installation manual	29

Assignment 1

Client

Business Name

The business name is **My Vitality**.

The business name was chosen to evoke feelings of a user taking control of their own energy and strength. This fits with the ethos of the company.

Description of client

My Vitality is a small up and coming business based in the heart of Shanghai, China. It offers a variety of services relating to Chinese herbal medicine. These include herbal therapy, acupuncture, and dietary therapy.

Employing two staff, both of whom are bi-lingual in Chinese and English, allows them to specifically target the large expatriate community in Shanghai who are unable to speak Chinese. This also allows for the local population to use their services.

The client would like to create an enterprise-level system to replace its current system which consists of manual files, spread sheets, and personal diary entries. The new system will act as a foundation for the future expansion of the company. It is the company's vision to expand its client base within the Shanghai area, while creating new revenue streams via an online presence.

Description of the proposed system

The goals of the My Vitality system are to modernize the technology, functionality, and security of the current system by replacing the current spreadsheet and paper-based storage with a relational database. My Vitality will create an online presence for the company via a Web page interface and online ordering system. This will allow customers to view and access an online storefront as an alternative to in-store purchases, as well as a portal through which they can interact with the company and find out more of what you do.

The database will serve as a central store for the company's data. It will connect with Web page interface and online ordering system allowing the customers to be able to see up to date stock availability and prices, purchase products and receive invoices for the purchase. The HCP will be given comprehensive MIS reporting from the data stored in the database.

Current Web standards will be followed while building the Web page interface and online ordering system, and a mobile first design approach will be adopted for the Web site interface. Furthermore, the system will adopt a responsive design philosophy and react to and change to a variety of device screen sizes.

The Web site and online ordering system will be programmed in a variety of languages:

- HTML5 will be used to build the structure of the site
- CSS will be used to add style to the site
- JavaScript will be used to add interactivity to the site
- SQL will be used to create and maintain the database
- PHP will be used as the language to interface with the Web site and the database. It will also perform server-side processing and validation

Scope of the proposed system

The new system should be capable of:

- Collecting and storing information about supplements
- Collecting and storing information contained in invoices
- Creating MIS reports
- Connecting via the Internet to transmit and receive data
- Connecting to an online relational database
- Connecting to an external providers banking application
- Running on at least the Windows 7 Operating System
- Running an open-source and license fee free relational database
- Supporting different levels of access
- Supporting a graphical user interface
- Supporting a Web interface for online customers
- Allowing online customers to purchase supplements
- Allowing online customers to view available stock levels
- Allowing patients to view selected account data and view and receive notifications via a mobile application
- Automation of messages (SMS or email; Mobile application notification)

Community

Who is the community?

The community is the people who live in the city of Shanghai, China.

More specifically, the company will focus on the expatriate community of Shanghai who do not speak Chinese.

Why this community?

I currently live just outside of Shanghai and my experiences here will allow me to tailor this project towards a need of a sub-section of its residents.

Shanghai (and China as a whole), has a large amount of expat residents. Of these residents a high proportion are unable to speak Chinese. Conversely, most Chinese residents cannot speak English. This lack of communication leaves many expats unable to experiment with and try out the traditional remedies used by the local population.

Furthermore, though China has implemented many initiatives to curb pollution, the air quality in Shanghai is still quite poor. Since prolonged exposure to poor air will likely lead to health problems, an alternate health care provider would be greatly appreciated by the non-Chinese speaking portion of the city.

Describe the community

Shanghai is China's largest city and is also the country's business capital, it is estimated to have around 25.8 million residents. This gives many foreigners the opportunity to relocate and work in the city. From the 2010 census, an estimated 160,000 expats were resident in Shanghai. Many who move to China cannot speak Chinese. According to an article in Beijing relocation, 73% of expats in China cannot speak Chinese. Many expats use English as the Lingua Franca to communicate between different languages.

(<http://www.beijingrelocation.com/blog/expats-in-china-nationalities-and-in-which-cities-they-settle/>)

(<http://www.stats-sh.gov.cn/tjnj/nje11.htm?d1=2011tjnj/E0214.htm?d1=2010tjnj/E0214.htm>)

(<http://worldpopulationreview.com/world-cities/shanghai-population/>)

Why do the community needs this software system?

This software would give the non-Chinese speaking community in Shanghai, and the rest of China, an opportunity to find out and use services relating to traditional Chinese herbal medicine.

It will allow the community to combine eastern and western medicine to boost their health, fight the health implications of living with poor air quality, and boost their immune systems to fight common airborne diseases which are easily transmitted throughout such a populous city.

Assignment 2

Preliminary Literature Review

Introduction

The My Vitality system is primarily a customer facing system. It will consist of a Web site and an online ordering system, that connects with a central store of data, contained within a relational database. This central data store will also allow the HCP of the business to receive enhanced MIS reporting capabilities.

The new system should be capable of:

- Collecting and storing information about supplements
- Collecting and storing information contained in invoices
- Creating MIS reports
- Connecting via the Internet to transmit and receive data
- Connecting to an online relational database
- Connecting to an external providers banking application
- Running on at least the Windows 7 Operating System
- Running an open-source and license fee free relational database
- Supporting different levels of access
- Supporting a graphical user interface
- Supporting a Web interface for online customers
- Allowing online customers to purchase supplements
- Allowing online customers to view available stock levels
- Allowing patients to view selected account data and view and receive notifications via a mobile application
- Automation of messages (SMS or email; Mobile application notification)

Justification

Shanghai is a city like no other, it does not sleep, it does not stand still. Its residents have grown accustomed to constant changes in their environment, and to innovation around every corner.

- Since the year 2000 the population has almost doubled, leaving Shanghai as the one of the most populous cities on the planet. (Year 2000: 13,959,000, Year 2018: 25,888,000).
- Opened its first subway line in 1993. It now has 15 lines covering 673 kilometers.

- It has over 200 sky scrappers
- Online shopping in China has grown from 33.57 million users in 2006 to 466.7 million users in 2016

But possibly the biggest change that has happened in Shanghai, and most Urban areas in China, has occurred over the last 3 years. Most people now use their smart phones to pay for everything; shopping, taxis, the cigarette seller on the street, even buskers can have a QR code ready for you to scan to transfer them some money. Physical cash, paper and coins, is quickly becoming obsolete.

My Vitality is needed to address the needs of its community. They embrace technology and expect efficiency. Customers want the option to purchase online, the busy lives they lead mean that time is at a premium; they want to know that what they ordered is available; they want to know about and feel connected with the company. In today's technological world, customer service is not only done in the physical world, but also the virtual world too. My Vitality will allow the company to enter the E-commerce market in a move to attract new customers, but it is equally important to enter the market to satisfy its existing customer's needs.

Chinese herbal medicine has slowly evolved over thousands of years, gradually adjusting its practices to the changing clinical conditions of its patients. My Vitality will provide an immediate overhaul of the current system, to quickly and effectively meet the rapidly changing needs of its clients.

For the company to compete in today's marketplace, it must imitate the city where it lives. It must not sleep, it must not stand still.

Past and current research [systems] in the community area.

I reviewed two websites for Chinese Herbal medicine, one of which is in English and the other in Chinese.

The www.chongjingtang.cn website has all the information in Chinese but can be translated via Google to get an idea of its content. It is a single page website with navigation links on the side which will place you at various sections throughout the page. It contains images throughout the site that promote the feeling of health and wellness. The Web site resizes gracefully across a variety of screen sizes. From a westerner's perspective, the use of traditional Chinese calligraphy throughout the site enhances the idea of traditional eastern medicine. Though in reality, this is 'normal' in a Chinese site as this is their language. However, it could be used sporadically throughout the customer facing sections of the proposed system to enhance the look, feel and integrity of the site from a westerner's perspective.

The house of harmony has a simple website. I feel that their use of color impedes the usability of the site. The text on some pages is hard to read. The site does not respond well to changing screen sites and does not have a responsive design. Users viewing this site on a mobile device would not be treated to a

good user experience. However, the simple black and white images used to display the treatments look appealing and the information throughout the site is written in a short, clear and concise manner.

Neither site seems to provide an online ordering system.

Note: I have accessed the site (www.chongjingtang.cn) from in China, with the firewall restrictions within China I am unsure whether the site is accessible from outside. If I access it via an American VPN connection, it does not load. If necessary I can provide screen shots of the site upon request.

How does the system fit in the community?

This system is designed to improve the company's business processes and to develop an online presence. This in turn, helps improve the services that the company offers its users, and allows more user to know about and use the company. Since the company's services are based around health and wellness, this has the knock-on effect of contributing to the improved health of its users, and in-turn, the community.

The clients need this system to be able to access the range of supplements that the company provides without the need to go to the physical store. This provides the client with the opportunity to read about the benefits of the supplements that are offered and then allows them to purchase them. This benefits the client in the following ways:

- Saves travel time to and from the store
- Ensures that the product is in-stock when you order (may not be in-stock if you travel to store)
- Allows clients who do not live close enough to travel to store to purchase
- Allows clients who have disabilities the option to buy, if their disability impedes them travelling
- Allows clients to purchase products anonymously (some customers may be embarrassed buying certain products in person. So even though they give their name and address, they conduct business online and do not have face-to-face contact)
- Allows clients to research the products at in their own time
- Allows clients to know the company ethos and goals

Enhanced MIS reporting will give the HCP better information and will improve the decision-making process. This could lead to lower costs, more competitive pricing schemes, promotions, or even changing the opening hours of the company. This will also serve as a benefit to the client in the medium to long term.

Development: what approach will be followed for the system?

The system will be developed by using an adaptive approach to the SDLC. The adaptive approach will allow the system to be planned, designed and implemented according to the agreed upon deadlines for phase completion.

After the initial project proposal phases have been completed (Assignments 1,2,3), the adaptive approach will allow the system to be separated and developed using iterations. This will allow the database to be analyzed, designed and developed (assignment 4), the MIS reporting system to be analyzed, designed and developed (assignment 5), the system to be analyzed, designed and developed (assignment 6), and the user manual and documentation to be analyzed, developed and designed (assignment 7).

Core processes	Iterations			
	1	2	3	4
Identify problem				
Plan				
Analysis				
Design				
Implementation				
Test and Integrate				

Key	System
1	Database
2	MIS Reporting
3	Chosen System
4	User manual and documentation

How will the system be tested?

The system will be tested according to a test plan. The plan will specify how, when the testing will be performed, who will conduct it, and what data will be used. The plan will consist of four phases.

Phase One: The first phase involves the programmer compiling the program during coding to remove any syntax errors and to desk check the program code to spot and remove logic errors.

Phase Two: The second phase involves testing an individual unit or module of code before it is integrated into the software. This will include testing that methods produce the correct results and that classes model the data accurately and function as stand-alone units. This phase is called unit testing.

Phase Three: The third phase involves testing that the programs created can work together. This includes testing that the individual units tested in phase two can work in unison with each other. This is called integration testing.

Phase Four: The fourth phase involves testing the entire system. This is to ensure that the system has delivered the required level of functionality and has satisfied the agreed upon business needs. Once system testing has been completed in-house, the system will be demonstrated to the clients for the final acceptance tests.

What would be the scope for future development for this system?

The system could allow customers to create personal accounts and become users of the system. The user account could then allow them access to book their own appointments online or to store payment details.

The online payment system could be improved from allowing only EFT payments. A common form of payment in China is through WeChat wallet. This allows a user to scan a QR code and transfer money to the owner of the QR code via the application. This would be relatively simple to implement and would make the system more accessible for the Chinese market.

The system could be expanded to allow for online consultations. The user could book an online appointment and the HCP could connect with the user via an online video chat software such as Skype.

Reference list

Statista. (2018). *China: online shoppers 2016 / Statistic*. [online] Available at: <https://www.statista.com/statistics/277391/number-of-online-buyers-in-china/> [Accessed 16 Mar. 2018].

Chongjingtang.cn. (2018). *Zhongjingtang (Chinese Meridian Church) - Meridian Health, Healthy Self-Healing, Zhongjingtang official Website*. [online] Available at: <http://www.chongjingtang.cn/> [Accessed 9 Mar. 2018].

Thehouseofharmony.com. (2018). *acupuncture - thehouseofharmony*. [online] Available at: <http://www.thehouseofharmony.com/acupuncture/4567605250> [Accessed 16 Mar. 2018].

Satzinger, J., Jackson, R. and Burd, S. (2016). *Systems analysis and design in a changing world*. 7th ed. Boston: Cengage Learning.

Tilley, S. and Rosenblatt, H. (n.d.). *Systems analysis and design*. 11th ed. Cengage Learning.

Assignment 3

System Proposal

Outcomes

Shanghai (and China as a whole), has a large amount of expat residents. Of these residents a high proportion are unable to speak Chinese. Conversely, most Chinese residents cannot speak English. This lack of communication leaves many expats unable to experiment with and try out the traditional remedies used by the local population. This system would give the non-Chinese speaking community in Shanghai, and the rest of China, an opportunity to find out and use services relating to traditional Chinese herbal medicine.

Furthermore, though China has implemented many initiatives to curb pollution, the air quality in Shanghai is still quite poor. Since prolonged exposure to poor air will likely lead to health problems, an alternate health care provider would be greatly appreciated by the non-Chinese speaking portion of the city. Hence, this system will also allow the community to reduce the use of western medical practices, or to combine the western medicine that they are accustomed to, with the eastern medicine that the company provides, to fight the health implications of living with poor air quality, and to boost their overall health and immune systems, to fight common airborne diseases which are easily transmitted throughout such a populous city.

Finally, the system will also allow the community to integrate the company into their digital lives. Users digital and physical lives are becoming ever closer. Today, users need and expect a digital entry point into a company. The My Vitality system will give users this option via the Web interface and allow them to purchase products via the online ordering system.

Objectives

The goals of the My Vitality system are to modernize the technology, functionality, and security of the current system and to create an online presence for the company via a Web page interface and online ordering system.

However, the purpose of the system is to enhance the company's relationship with its customers, to attract new customers, and to create a new revenue stream from entering the E-commerce arena. For the system to achieve these goals, it must merge with its customers use of the Web and provide ways to measure if it is achieving it.

For the site to successfully integrate into a customer's life, firstly it must be accessible to the customer.

- We must measure the uptime capacity of the service provider. A provider with less than a 99.9% up-time should not be used.
- We need to ensure that a customer receives quality and equivalent user experiences across a variety of devices.
- We need to ensure that accessibility controls are implemented throughout the site. A customer with a visual, mobility, cognitive or auditory disability, should have the same experiences as any other user

Secondly, the customer must be able to easily find what they are looking for:

- We must ensure that the site provides an easy to use navigation system. The company should consider using analytics to see the user's clickstream.
- The primary goal of the online ordering system is to be able for a customer to purchase products. Thus, the navigation system on the site should be set up so that a customer finds it easily. This can be measured by the number of users who purchase products on the system.

Furthermore, the site must provide interest to the customer and entice the user to return to the site in future. This is where the site can try to integrate with a user's digital life. To achieve this the site must offer more than general information about the company and the online ordering system. The company must invest in complementary assets. Fortunately, the complementary assets for this system, do not necessarily require additional monetary investment.

The Web is a two-way interaction medium, with both parties being able to interact with each other. This gives the company an opportunity to connect with potential customers in a more meaningful way. Fortunately, the company does not have to build a platform for this interaction to occur, they can join existing social media platforms and connect via here. These platforms now serve as a mainstay in people's lives, so as well as serving as a communication platform, it will also serve as vital entry points to go to the site and an advertising tool.

Producing articles or running a blog would also increase the stickiness of the site and likelihood that a user would return. The articles/blog would be hosted on the site, but links could be posted on the social media platforms, used to entice the users to come to the site. From here the user is more likely to make a purchase.

- Analytics can track where the customer entered the site from, how long they stayed, and where they went
- Social media use will give feedback onto how users feel about the company
- Social media will provide a means to provide quick and effective customer service to an individual person, while also allowing anyone else to see how professionally you dealt with that customer. This will enhance the trust in the company.
- Increased followers on social media can give an indication that the brand is growing, conversely, the opposite will also indicate that something is going wrong

In conclusion, the system is a means to enhance customer relations and provide a new revenue stream for the company. Since enhanced customer relations often provides intangible benefits which are difficult to measure, the company must try to track web site statistics such as up-time capacity, visitor numbers, and gauge customers attitude to the company via feedback from customer relations. However, the online ordering system provides tangible benefits, which can be viewed by increased sales via the online platform, and in-store customer numbers increasing.

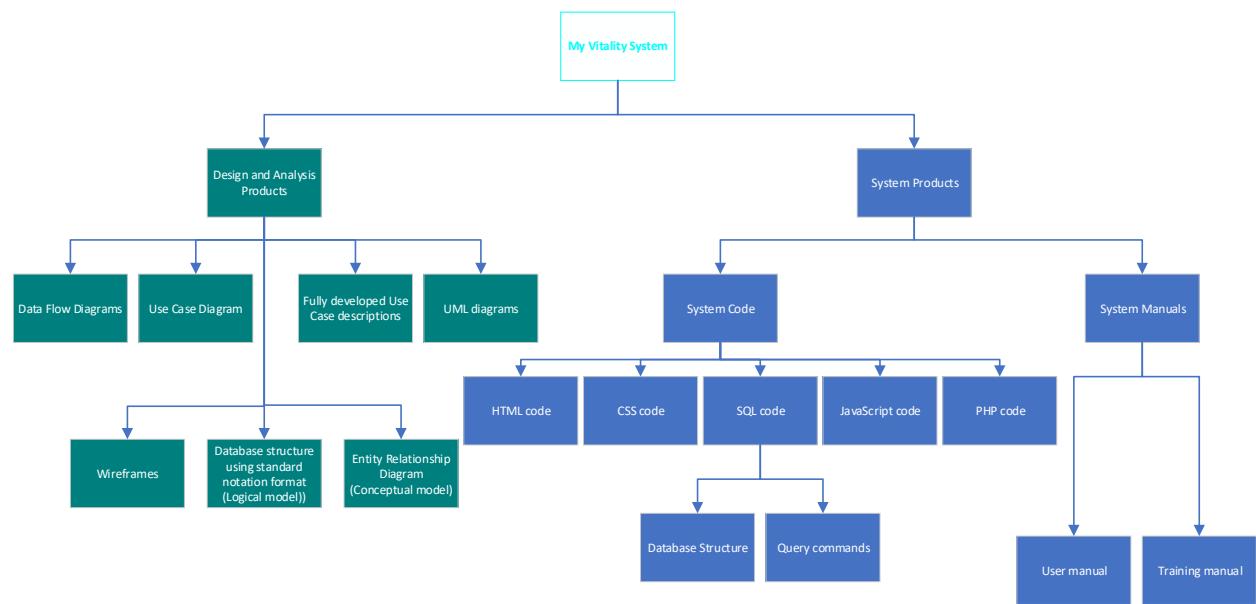
Outline deliverables

The My Vitality system will provide the following deliverables:

- Design and Analysis Products
 - Use Case Diagram
 - Fully Developed Use Case Descriptions
 - Data Flow Diagrams
 - UML Diagrams
 - Entity Relationship Diagrams
 - Database Structure using Standard Notation Format
- System Products
 - System Code
 - HTML Code
 - CSS Code
 - JavaScript Code
 - PHP Code
 - SQL Code
 - Database Structure
 - Query Commands
 - System Manuals
 - User Manuals (Customer and Administration)
 - Installation Manual

Below is a graphical representation of the structured list above.

Product breakdown Structure

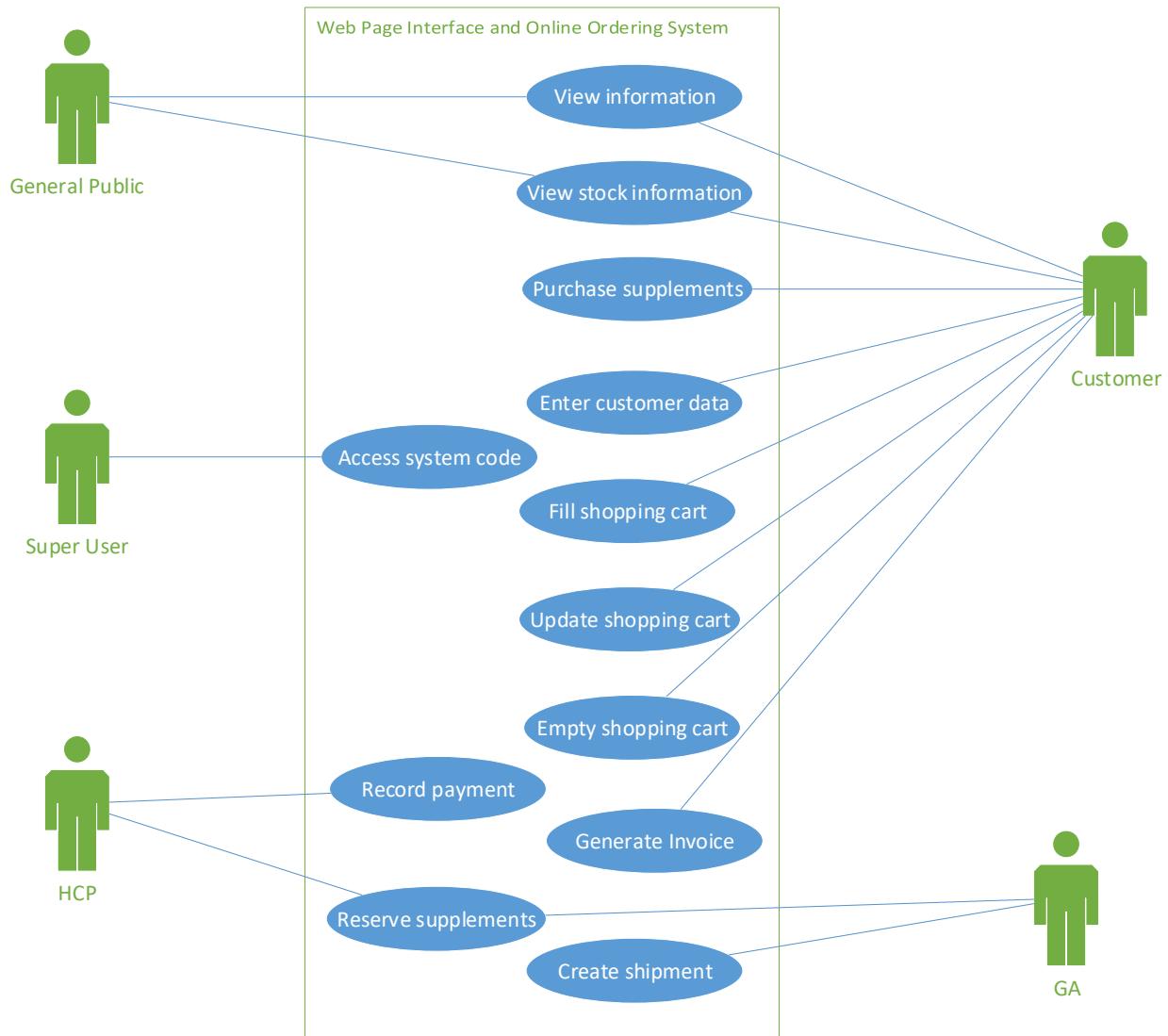


Scope

To view the scope of the system please view the use case diagram and fully developed use case descriptions below.

The use case diagram shows a high-level view of the systems capabilities and scope. It shows the actors (users) of the system and how they will interact with it, and what functionality is available to each actor.

To get a better understanding of each use case, please consult the fully developed use cases below the use case diagram.



Use case name:	View information	
Scenario:	Actor browses the online site	
Triggering event:	Actor goes to online site	
Brief description:	Display general info about the business and supplements, images and prices	
Actors:	General public, customers	
Pre-conditions:	System is online and Web site navigation works	
Post conditions:	Content is displayed	
Flow of activities:	Actor	System
	1. Actor access site via URL	1.1 System requests files from server
		1.2 Server provides files and system displays them
Exception conditions:	1.1. Server is down	
	1.1. Links are broken on Web site	
	1.2 Users browser is not compatible	

Use case name:	View stock information	
Scenario:	Actor wants to see stock levels	
Triggering event:	Actor selects stock levels on stock item	
Brief description:	Display and indicate if stock is available for each supplement	
Actors:	General public, customers, patients	
Pre-conditions:	Database and Web site must be online and available	
Post conditions:	Stock levels are shown	
Flow of activities:	Actor	System
	1 Actor goes to product page	1.1 System retrieves the stock levels from database
		1.2 System displays stock level with product
Exception conditions:	1.1 Database is offline	
	1.1 Scripting language files are missing	
	1.1 SQL query is incorrect	

Use case name:	Purchase supplements									
Scenario:	Actor purchases supplements online									
Triggering event:	Actor leaves shopping cart with intention to complete transaction									
Brief description:	Actor has filled shopping cart with desired products and proceeds to payment section									
Actors:	Customers									
Pre-conditions:	Shopping cart functionality is working, database is online									
Post conditions:	System has stored transaction temporarily and directs user to enter personal information									
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Actor clicks complete transaction button</td> <td>1.1 System processes transaction</td> </tr> <tr> <td></td> <td>1.2 Verifies that required stock is available</td> </tr> <tr> <td></td> <td>1.3 System requests actor information</td> </tr> </tbody> </table>		Actor	System	1. Actor clicks complete transaction button	1.1 System processes transaction		1.2 Verifies that required stock is available		1.3 System requests actor information
Actor	System									
1. Actor clicks complete transaction button	1.1 System processes transaction									
	1.2 Verifies that required stock is available									
	1.3 System requests actor information									
Exception conditions:	1.1 User leaves site 1.2 Required stock levels not available 1.3 Form for entering user information fails to load									

Use case name:	Enter customer data													
Scenario:	Actor enters their data to be able to complete transaction													
Triggering event:	Actor leaves shopping cart with intention to complete transaction													
Brief description:	Actor enters their information and the system records it before going to payment section													
Actors:	Customers													
Pre-conditions:	System has verified transaction is ready to proceed													
Post conditions:	User information is stored and generate invoice use case called													
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. User enters personal information</td> <td>1.1 System validates user information</td> </tr> <tr> <td></td> <td>1.2 System confirms with user information validity</td> </tr> <tr> <td>2. User confirms validity</td> <td>2.1 System stored information</td> </tr> <tr> <td></td> <td>2.2 System calls generate invoice use case</td> </tr> <tr> <td></td> <td>2.3 Informs user of transaction completion</td> </tr> </tbody> </table>		Actor	System	1. User enters personal information	1.1 System validates user information		1.2 System confirms with user information validity	2. User confirms validity	2.1 System stored information		2.2 System calls generate invoice use case		2.3 Informs user of transaction completion
Actor	System													
1. User enters personal information	1.1 System validates user information													
	1.2 System confirms with user information validity													
2. User confirms validity	2.1 System stored information													
	2.2 System calls generate invoice use case													
	2.3 Informs user of transaction completion													
Exception conditions:	1.1 User fails to enter valid data 2.1 System fails to store data 2.2. System fails to generate invoice													

Use case name:	Fill shopping cart							
Scenario:	Adds items to shopping cart							
Triggering event:	User selects a product and enters the quantity required							
Brief description:	User selects quantity of product and system remembers selection for session							
Actors:	Customer							
Pre-conditions:	Shopping cart is empty							
Post conditions:	System has stored the actors selections							
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Actor selects a product and quantity to buy</td> <td>1.1 System stores the selection and quantity</td> </tr> <tr> <td></td> <td>1.2 Asks actor whether they want to check out or continue shopping</td> </tr> </tbody> </table>	Actor	System	1. Actor selects a product and quantity to buy	1.1 System stores the selection and quantity		1.2 Asks actor whether they want to check out or continue shopping	
Actor	System							
1. Actor selects a product and quantity to buy	1.1 System stores the selection and quantity							
	1.2 Asks actor whether they want to check out or continue shopping							
	2. Actor makes selection	2.1 Directs user to selected place						
Exception conditions:	1.1 Actor selects invalid quantity 1.2 Actor leaves site 1.2 Actor empties shopping cart							

Use case name:	Update shopping cart							
Scenario:	User updates the quantity's held inside the shopping cart							
Triggering event:	User clicks on the update quantity button							
Brief description:	User updates the quantity's held inside the shopping cart and the system stores the updated data							
Actors:	Customer							
Pre-conditions:	Shopping cart has users items stored							
Post conditions:	Quantities of items stored have been altered and saved							
Flow of activities:	<table border="1"> <thead> <tr> <th>Actor</th> <th>System</th> </tr> </thead> <tbody> <tr> <td>1. Actor increases or decrease quantity in cart</td> <td>1.1 System stores the selection and quantity</td> </tr> <tr> <td></td> <td>1.2 Asks actor whether they want to check out or continue shopping</td> </tr> </tbody> </table>	Actor	System	1. Actor increases or decrease quantity in cart	1.1 System stores the selection and quantity		1.2 Asks actor whether they want to check out or continue shopping	
Actor	System							
1. Actor increases or decrease quantity in cart	1.1 System stores the selection and quantity							
	1.2 Asks actor whether they want to check out or continue shopping							
	2. Actor makes selection	2.1 Carries out user choice						
Exception conditions:	1.1 Actor selects invalid quantity 1.2 Actor leaves site 1.2 Actor empties shopping cart							

Use case name:	Empty shopping cart	
Scenario:	Remove products from cart	
Triggering event:	User deselects an item in the shopping cart	
Brief description:	User removes a product from the cart and the system stores the updated data	
Actors:	Customer	
Pre-conditions:	Items are stored in the shopping cart	
Post conditions:	Actors selections have been removed from cart	
Flow of activities:	Actor	System
	1. Actor removes items from cart	1.1 System removes selection choices
		1.2 Asks actor whether they are sure
Exception conditions:	1.1 Actor decides to keep items in cart	

Use case name:	Generate invoice	
Scenario:	Create invoice to document customer transaction	
Triggering event:	Customer completes an online order	
Brief description:	Create invoice of customers data and supplements purchased, and email to HCP/GA	
Actors:	Customer, GA, HCP	
Pre-conditions:	Transaction has been initiated	
Post conditions:	System has stored transaction data and created and sent invoices	
Flow of activities:	Actor	System
	1. Actor completes transaction	1.1 System gets customer and transaction data
		1.2 Generates invoice from data
		1.3 Places a hold on requested supplements
		1.3. Stores invoice data
		1.4 Sends invoice to customer and HCP email
		1.5 Displays invoice on screen
		1.6 Thanks customer for purchase
Exception conditions:	1.1 system cannot retrieve data 1.3 Database system is offline 1.4 Email mails system is down	

Use case name:	Reserve Stock	
Scenario:	Reserve stock after online sale	
Triggering event:	Online sale is made	
Brief description:	Reserve supplements that have been sold online while payment is processed	
Actors:	GA, HCP	
Pre-conditions:	Invoice has been created	
Post conditions:	Stock requested has been removed from normal operations	
Flow of activities:	Actor	System
	1. Invoice has been created	1.1 System records the items purchase
		1.2 Updates the quantity for sale
		1.3. Stores invoice data
		1.4 Updates the reserved items table
		1.5 Updates the available stock levels on the Web Site
Exception conditions:	1.1 System cannot retrieve data 1.3 Database system is offline	

Use case name:	Record Payment	
Scenario:	Actor approves the transaction	
Triggering event:	Customer pays for order and payment company sends email to approve payment	
Brief description:	Enter and store what the patient has paid	
Actors:	HCP	
Pre-conditions:	Actor receives payment confirmation from payment company and supplements are on hold	
Post conditions:	Transaction has been approved, stock is approved to be released	
Flow of activities:	Actor	System
	1. Actor approves the transaction	1.1 System initiates create shipment use case
		1.2 System releases on-hold items from stock
		1.3 System updates order status
Exception conditions:	1. Payment has not been approved 1. Customer never paid or did not pay in full 1 Actor cannot access payment approvals	

Use case name:	Create shipment									
Scenario:	Create the package for the online shipment									
Triggering event:	HCP approves transaction									
Brief description:	Transaction approval confirmation is received, and the actor create the package for shipment									
Actors:	GA									
Pre-conditions:	Packaging and supplements are available									
Post conditions:	Package is shipped									
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Actor receives email containing shipment details</td> <td></td> </tr> <tr> <td>2. Actor creates shipment</td> <td></td> </tr> <tr> <td>3. Actor enters shipment outgoing details</td> <td>3.1 Stores the shipment tracking details</td> </tr> </tbody> </table>	Actor	System	1. Actor receives email containing shipment details		2. Actor creates shipment		3. Actor enters shipment outgoing details	3.1 Stores the shipment tracking details	
Actor	System									
1. Actor receives email containing shipment details										
2. Actor creates shipment										
3. Actor enters shipment outgoing details	3.1 Stores the shipment tracking details									
Exception conditions:	3. Database is down									

Use case name:	Access system code									
Scenario:	Actor performs system maintenance									
Triggering event:	System malfunctions, requires scheduled maintenance or upgrade									
Brief description:	Actor accesses system code and performs maintenance task									
Actors:	Super User									
Pre-conditions:	System is turned on									
Post conditions:	Maintenance task is complete									
Flow of activities:	<table border="1"> <thead> <tr> <th style="text-align: center;">Actor</th> <th style="text-align: center;">System</th> </tr> </thead> <tbody> <tr> <td>1. Actor logs on via super user interface</td> <td>1.1 System grant user access</td> </tr> <tr> <td>2. Actor performs maintenance</td> <td></td> </tr> <tr> <td>3. Actor logs out of system</td> <td></td> </tr> </tbody> </table>	Actor	System	1. Actor logs on via super user interface	1.1 System grant user access	2. Actor performs maintenance		3. Actor logs out of system		
Actor	System									
1. Actor logs on via super user interface	1.1 System grant user access									
2. Actor performs maintenance										
3. Actor logs out of system										
Exception conditions:	1. Actor cannot gain access to system 1. Code has been destroyed 2. Actor cannot perform maintenance takes									

Approach

The system will be developed by using an adaptive approach to the SDLC. The adaptive approach will allow the system to be planned, designed and implemented according to the agreed upon deadlines for phase completion.

After the initial project proposal phases have been completed (Assignments 1,2,3), the adaptive approach will allow the system to be separated and developed using iterations. This will allow the database to be analyzed, designed and developed (assignment 4), the MIS reporting system to be analyzed, designed and developed (assignment 5), the system to be analyzed, designed and developed (assignment 6), and the user manual and documentation to be analyzed, developed and designed (assignment 7).

Core processes	Iterations			
	1	2	3	4
Identify problem				
Plan				
Analysis				
Design				
Implementation				
Test and Integrate				

Key	System
1	Database
2	MIS Reporting
3	Chosen System
4	User manual and documentation

Exclusions

The system will not:

- allow customers to create accounts
- allow alternative payment methods (other than EFT)
- allow any user access to the system code (other than the Super user)

Constraints

A constraint is a requirement or condition that the system must satisfy or an outcome that the system must achieve. The system has the following constraints:

- The system must use a relational database to store the data.
- The system must use the current data as a point of entry.
- The current data must be imported into the database.
- The database must be open source and not require a license fee.
- The system must be able to operate on the Windows 7 operating system.
- The HCP must be able to continue immediately with the system.
- The super user is the only user who has access to the database and system code.
- Only the HCP can update the pricing structures for:
 - percentage markup for new supplements
 - Discounts for a specific patient or customer
- The system must be able to generate MIS reports regarding patients, supplements and sales
- The MIS reports are only visible to HCP
- The system must indicate whether stock is available
- The system must indicate the retail price of the stock
- The system must allow customers to purchase supplements online
- The system must only allow payment via EFT payment
- The system must be able to immediately reserve stock that has been purchased
- The system must be able to generate and automatically send emails
- The database must be operational by 21 May 2018
- The MIS system must be operational by 25 June 2018
- The user and installation manual to be completed by 01 October 2018
- The system must be operational by 22 October 2018 (No presentation date given yet)

For a list of exception conditions that will prevent an individual use case from completion, please view the fully developed use case diagrams on pages 19 – 24.

Outline business case

This system is designed to improve the company's business processes and to develop an online presence. This in turn, helps improve the services that the company offers its users, and allows more user to know about and use the company. Since the company's services are based around health and wellness, this has the knock-on effect of contributing to the improved health of its users, and in-turn, the community.

The clients need this system to be able to access the range of supplements that the company provides without the need to go to the physical store. This provides the client with the opportunity to read about

the benefits of the supplements that are offered and then allows them to purchase them. This benefits the client in the following ways:

- Saves travel time to and from the store
- Ensures that the product is in-stock when you order (may not be in-stock if you travel to store)
- Allows clients who do not live close enough to travel to store to purchase
- Allows clients who have disabilities the option to buy, if their disability impedes them travelling
- Allows clients to purchase products anonymously (some customers may be embarrassed buying certain products in person. So even though they give their name and address, they conduct business online and do not have face-to-face contact)
- Allows clients to research the products at in their own time
- Allows clients to know the company ethos and goals

Enhanced MIS reporting will give the HCP better information and will improve the decision-making process. This could lead to lower costs, more competitive pricing schemes, promotions, or even changing the opening hours of the company. This will also serve as a benefit to the client in the medium to long term.

My Vitality's marketing objectives are as follows:

- 1) To increase our brand awareness on social media platforms by 100 users per month for 12 months. The company currently does not have a presence on social media, so it will start at zero.
- 2) To obtain a 50% return rate of customers within three months of completing a purchase for the next 12 months
- 3) To increase online sales to new customers by 25% when compared to the corresponding month in the previous year for the next 12 months

My Vitality will meet these objectives by following the following strategies and tactics:

1)

To increase our brand awareness on social media platforms by 100 users per month for 12 months, we have chosen to use a cross-platform marketing strategy, utilizing Facebook, Twitter and a company WeChat account

To increase users, we will focus on creating relevant content that is of use to our target customer. The content will be uploaded to our web site and then shared on Facebook, Twitter and WeChat.

Content will include:

- creating well written articles on health, nutrition and wellbeing
- regular updates to our blogs

We will also look to interact with our followers. Social media users like to feel a part of a brand. To accomplish this, we will look to share and like our follower's posts or tweets if they are uplifting, motivational, related to health, nutrition or wellbeing. Any negative reaction towards our brand will be

quickly addressed to see why they are upset and how it can be fixed.

To increase brand awareness, we need to get what we do and how we do it to our customers, The first step in achieving this is to set up full and detailed profile pages on Facebook, Twitter and WeChat. Each of these pages will have details and links to our main site.

Secondly, each piece of content that is created must have our brand incorporated consistently, must include links to our social media pages and it must be easy for a user to share or like our content and brand.

To ensure that we are reaching our target customers, we must post our content at the times of the day when our customers are most likely to see it, while being mindful that we do not post too much, for us to be perceived as spamming their news feeds. According to various research studies:

- Facebook users have a 32% higher engagement on a Saturday and a Sunday
- Facebook users have a 18% higher engagement on a Thursday and a Friday
- Facebook posts at 1pm get the most shares
- Facebook posts at 3pm get the most clicks
- Posting to Facebook once per day is best or you'll start to feel like spam to a user
- Twitter users are more likely to be on Twitter on their commute to work
- The best time to tweet is between 1700 and 1900 or noon

The results of recommended tweeting frequency research show large variations in their results sets. We have chosen to go with the 'less is more approach' and limit our tweets to no more than 3 per day.

To measure whether we are achieving our objective, we will record statistics on the first and last day of each month and compare the results. The statistics that we will be monitoring are follower numbers on each platform, the number of total number of re-tweets, likes and shares that our content has received on each platform.

2)

To obtain a 50% return rate of customers within three months of completing a purchase for the next 12 months, we plan to ensure that:

- the supplements are shipped on the first working day after purchase
- the customer is made aware of any delays in shipments at earliest opportunity
- shipment is packaged appropriately with the right quantity and correct items

For any user to re-purchase a product, they must be happy with the product that they purchased. To ensure that the user gets off to a good start, they will receive a welcome email after payment has been approved. In this email, we will provide the invoice of the product, the shipment details, contact details and thank them for their purchase. In this email we will also ask if they would like to sign up to our mailing list for discounts and promotions.

To measure whether we are achieving our objective, we will record and analyze, when a user re-purchases.

3)

To increase online sales to new customers by 25% when compared to the corresponding month in the previous year for the next 12 months, we plan to:

- ensure that our site has a 99.9% up time
- ensure that new content is developed each week and that it is distributed on our social media platforms and posted onto our own web site
- engage with customers by liking/sharing/commenting on their social media posts that fit in with our own ethos
- for 'slow' selling months where targets are not being met by the end of the second month, offer a limited number of discount codes via our social media outlets
- offer referral discounts to existing patients
- allow potential and existing customers the chance to join our monthly newsletter

To measure whether we are achieving our objective, we will compare the numbers of new customers in the current month to the same period in the previous year. We will track data:

- from the mailing list and customers who access our web site to see if they are making a purchase
- to see if discount codes in 'slow' selling months attracts new customers
- to see if offering referral discounts to existing customers attracts new customers
- to see the correlation between new customers and building a community of users and increasing our brand awareness on social media platforms

The funding sources for this project will be provided via the owner (HCP) of the company, so income generation is not necessary. Since the marketing strategy is based around social media use and content creation the company does not need a large marketing budget; most of the content will be created in house, but the company will set aside, a budget of 250RMB per month.

Shanghai is China's largest city and is also the countries business capital, it is estimated to have around 25.8 million residents. This gives many foreigners the opportunity to relocate and work in the city. From the 2010 census, an estimated 160,000 expats were resident in Shanghai. Many who make move to China cannot speak Chinese. Furthermore, online shopping in China has grown from 33.57 million users in 2006 to 466.7 million users in 2016. When combining the large number of expats in the area and the fact that online shopping in China is the 'norm' in this community we predict that we can grow the online business from zero to 1000 monthly transactions in 12 months.

Reasons for selecting this solution for the system

The reasons for selecting the solution for this system are:

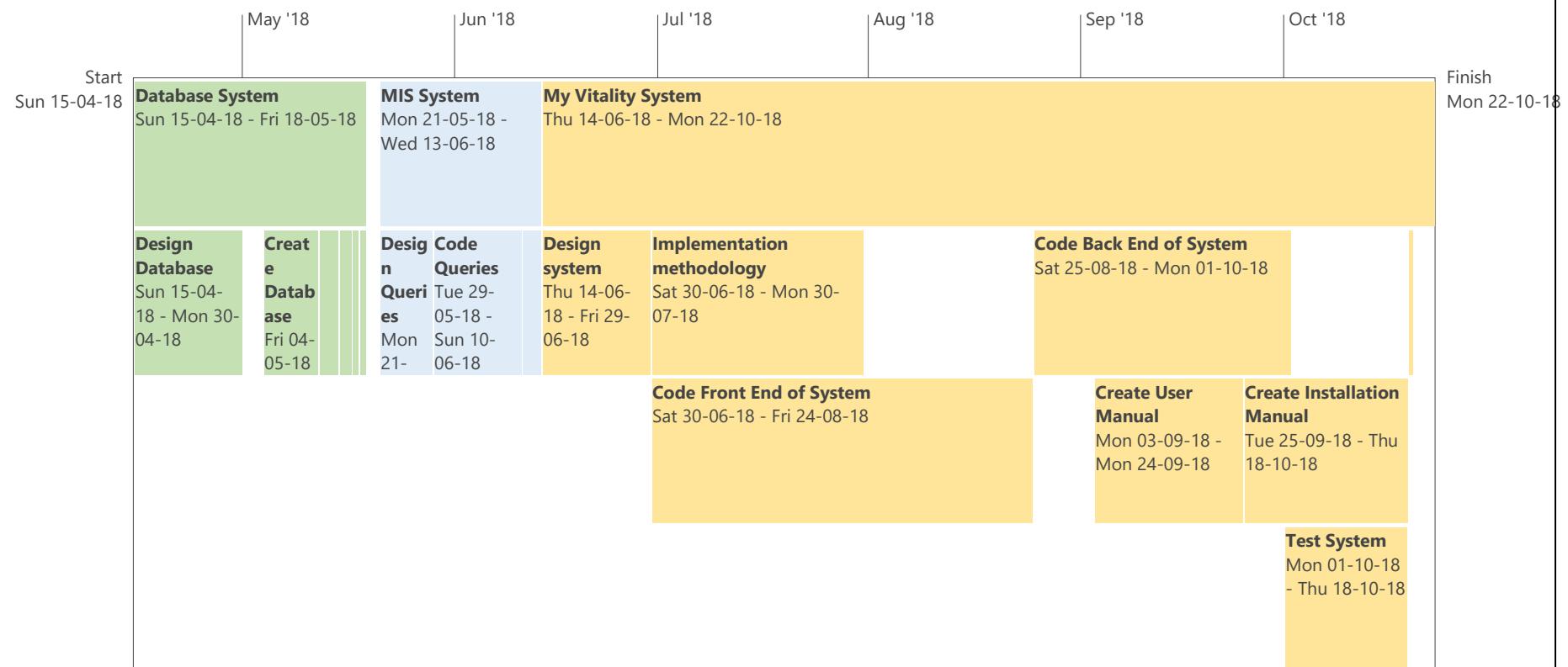
Improved service for patients and potential new customers. The system will aim to improve the service that the company gives to its customers. It will allow customers access to information about the company and its products and services. It will be integrated with social media platforms, so that customers have more ways of interacting with us.

Support for new services. The proposed solution will allow the company to offer online ordering and take advantage of the growth in e-commerce in the community.

Better performance. The upgrade from a spread sheet to a database will improve the speed of data retrieval, accuracy, and security of the company's data. Without the upgrade in data storage, it is unlikely that the company could implement the new system effectively.

More information. The new system will provide the HCP will more accurate and up-to-date information. This will allow them to improve the quality of the decisions throughout the company.

Outline project plan



Tabular format of the above project schedule:

Task Name	Duration	Start	Finish
Database System	26 days	Sun 15-04-18	Fri 18-05-18
Design Database	12 days	Sun 15-04-18	Mon 30-04-18
Create Database	6 days	Fri 04-05-18	Fri 11-05-18
Import Data into Database	2 days	Sat 12-05-18	Mon 14-05-18
Test Database	2 days	Tue 15-05-18	Wed 16-05-18
Take Screen Shots	1 day	Thu 17-05-18	Thu 17-05-18
Database Design, implementation and testing complete	1 day	Fri 18-05-18	Fri 18-05-18
MIS System	18 days	Mon 21-05-18	Wed 13-06-18
Design Queries	6 days	Mon 21-05-18	Mon 28-05-18
Code Queries	10 days	Tue 29-05-18	Sun 10-06-18
Test Queries	3 days	Mon 11-06-18	Wed 13-06-18
MIS Design, implementation and testing complete	0 days	Wed 13-06-18	Wed 13-06-18
My Vitality System	93 days	Thu 14-06-18	Mon 22-10-18
Design system	12 days	Thu 14-06-18	Fri 29-06-18
Implementation methodology	22 days	Sat 30-06-18	Mon 30-07-18
Code Front End of System	41 days	Sat 30-06-18	Fri 24-08-18
Code Back End of System	27 days	Sat 25-08-18	Mon 01-10-18
Test System	14 days	Mon 01-10-18	Thu 18-10-18
Create User Manual	16 days	Mon 03-09-18	Mon 24-09-18
Create Installation Manual	18 days	Tue 25-09-18	Thu 18-10-18
My Vitality System Design, implementation and testing complete	1 day	Fri 19-10-18	Fri 19-10-18

Resources

The resources that will be used are:

- Atom editor - Atom is a free and open-source text and source code editor
- phpMyAdmin - phpMyAdmin is a free and open source administration tool for MySQL and MariaDB
- Microsoft Visio - Microsoft Visio is a diagramming and vector graphics application and is part of the Microsoft Office family
- Google Chrome and Mozilla Firefox – Web Browsers with Developer tools
- W3C Markup and Validation service - [https://validator.w3.org/n](https://validator.w3.org/)

The client will be provided with the following:

- Use Case Diagrams
- Fully Developed Use Case Descriptions
- Data Flow Diagrams
- UML Diagrams
- Entity Relationship Diagrams
- Database Structure using Standard Notation Format
- System Code
 - HTML Code
 - CSS Code
 - JavaScript Code
 - PHP Code
 - SQL Code
- System Manuals
 - User Manual
 - Training Manual
- Fully operation system

Quality expectations

The system will deliver the agreed upon level of functionality that is specified throughout the project proposal. To deliver this level of functionality, system code will have tested throughout the coding process. The testing will include the following:

Phase One: The first phase involves the programmer compiling the program during coding to remove any syntax errors and to desk check the program code to spot and remove logic errors.

Phase Two: The second phase involves testing an individual unit or module of code before it is integrated into the software. This will include testing that methods produce the correct results and that classes model the data accurately and function as stand-alone units. This phase is called unit testing.

Phase Three: The third phase involves testing that the programs created can work together. This includes testing that the individual units tested in phase two can work in unison with each other. This is called integration testing.

Phase Four: The fourth phase involves testing the entire system. This is to ensure that the system has delivered the required level of functionality and has satisfied the agreed upon business needs. Once the system test is complete, the system will undergo a stress test. This will involve at least two people ordering supplements at the same time to ensure that the system can cope with multiple users.

Once system testing has been completed in-house, the system will be demonstrated to the clients for the final acceptance tests.

The system will also be tested for usability. This will be done by testing the user interface to ensure that it is easy to use and facilitates the functional requirements of the system.

The system must ensure that it handles all exceptions gracefully. The system should not crash. Code must use exception handling.

The response time for software responses to SQL queries should be measured and checked against system wide guidelines.

The service provider should be monitored for the up-time capacity it provides.

Risk assessment

Table containing the risk and its contingency / risk reduction technique.

Risk	Risk Reduction Technique
Code lost due to hardware failure	Back up code to an external hard drive
Code module fail to integrate with another module	Test modules during the coding process. Not just at the end
Code produces incorrect results or does not work	Log each result of failure and how it was fixed. Follow a test plan
User interface is not fit for purpose	Develop wireframes; task analysis
Staff sickness	Include some slack time in project schedule
Server failure	Choose a provider with multiple servers and a good reputation
Human error	Back up data to external hard drive
Software attack	Use an anti-virus program
Code vulnerability	Code using current Web standards. Incorporate defensive coding

Snapshot of the form for recording problems throughout the project:

#	ISSUE	TYPE	PRIORITY	OPENED ON	OPENED BY	CLOSED ON	CLOSED BY	NOTES
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								

Assignment 4

Database

ERD

Additional Notes for some specific entities within the ERD.

PURCHASE_JOURNAL is used to keep a historical record of the supplements added to the system.

SUPPLEMENT_COST is used to keep a historical record of any changes in the price of the supplements.

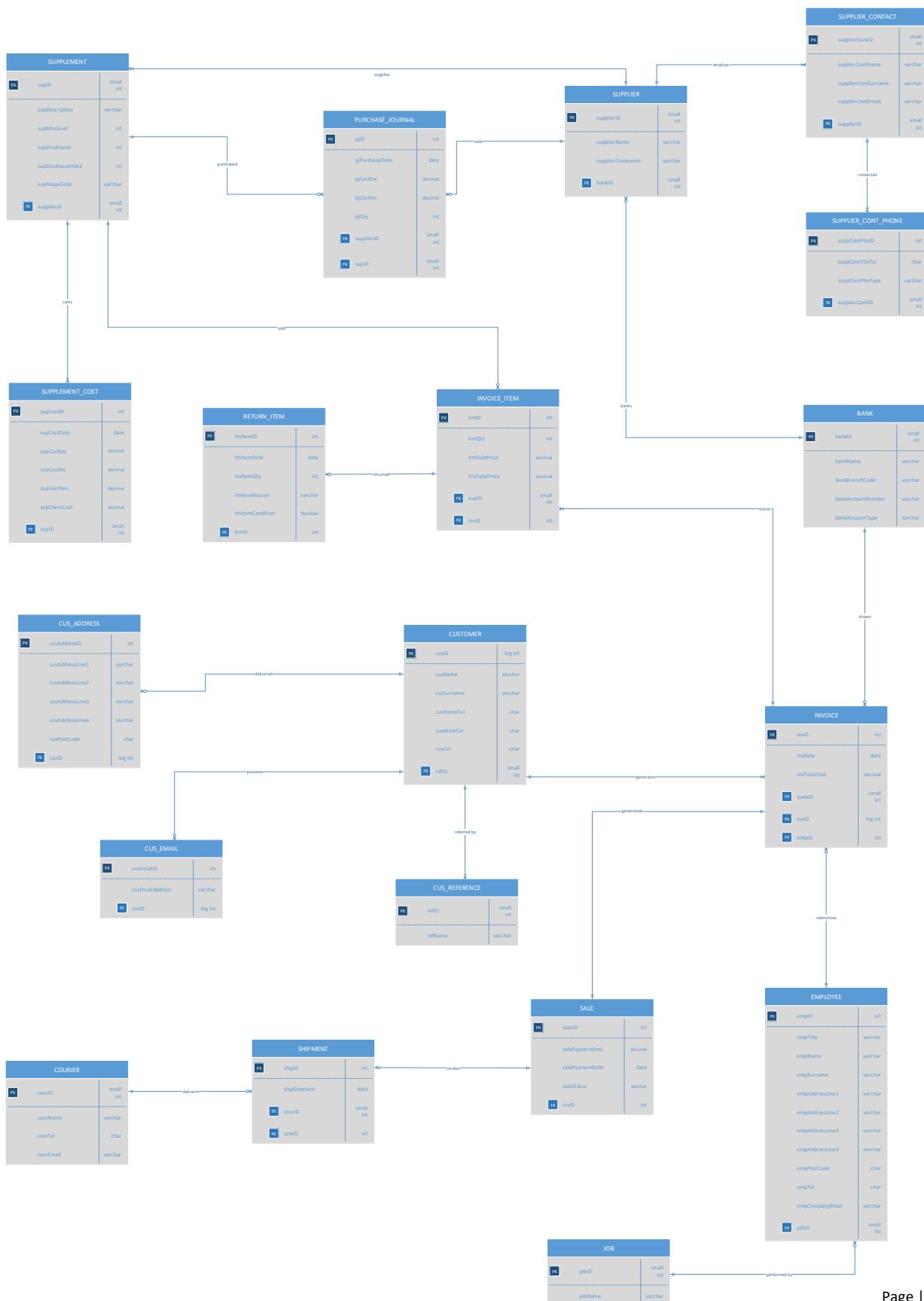
For example, if a supplement re-stock occurs and the supplement has been purchased at a different price to the current cost of the supplement, these two entities will track the costs and keep accurate records.

BANK is used to store bank account information. This is used to store the bank account information for suppliers as well as the bank account information for My Vitality (the account where customers must deposit money to complete their purchase)

SALE is used to track the status of transactions.

EMPLOYEE is used to track the data for employees. For the online ordering system, this entity is used to retrieve data for the outgoing invoice for who to make the payment to. Extra attributes such as address and contact information were included as it would likely be needed for future expansion of the system.

CUS_ADDRESS is used to store the address of the customer. Though each customer only inputs the delivery address for each order now, this separate entity allows for expansion of the system in future. For example, this would allow both a delivery and billing address for a customer to be stored without any modification to the database system.



Development | Database

The selected Database Management System is phpMyAdmin.

Logical Model

BANK (bankID, bankName, bankBranchCode, bankAccountNumber, bankAccountType)

PRIMARY KEY: bankID

CONSTRAINTS:

bankID SMALLINT PRIMARY KEY AUTO INCREMENT

bankName VARCHAR(35) NOT NULL

bankBranchCode VARCHAR(6) NOT NULL DEFAULT '000000'

bankAccountNumber VARCHAR(12) NOT NULL DEFAULT '0000000000'

bankAccountType VARCHAR(12) NOT NULL CHECK(bankAccountType IN ('CHEQUE', 'CURRENT', 'TRANSMISSION', 'SAVINGS', 'MZANSI'))

JOB (jobID, jobName)

PRIMARY KEY: jobID

CONSTRAINTS:

jobID SMALLINT PRIMARY KEY AUTO INCREMENT

jobName VARCHAR(3) NOT NULL CHECK(jobName IN ('HCP', 'GA', 'SU'))

COURIER (courID, courName, courTel, courEmail)

PRIMARY KEY: courID

CONSTRAINTS:

courID SMALLINT PRIMARY KEY AUTO INCREMENT

courName VARCHAR(35) NOT NULL

courTel CHAR(18) NOT NULL CHECK(courTel LIKE '--- --- -(----)')
DEFAULT '(000)-(000)-(0000)'

courEmail VARCHAR(50) NOT NULL CHECK(courEmail LIKE '%@%.---' OR
courEmail LIKE '%@%.--.--') DEFAULT 'xx@xx.xxx'

CUS_REFERENCE (refID, refName)

PRIMARY KEY: refID

CONSTRAINTS:

refID SMALLINT PRIMARY KEY AUTO INCREMENT

refName VARCHAR(30) NOT NULL

SUPPLIER (supplierID, supplierName, supplierComments, bankID)

PRIMARY KEY: supplierID

FOREIGN KEY: bankID REFERENCES BANK

CONSTRAINTS:

supplierID SMALLINT PRIMARY KEY AUTO INCREMENT

supplierName VARCHAR(35) NOT NULL

supplierComments VARCHAR(200) NOT NULL DEFAULT 'NONE'

bankID REFERENCES BANK(bankID)

SUPPLIER_CONTACT (supplierContID, supplierContName, supplierContSurname, supplierContEmail, supplierID)

PRIMARY KEY: supplierContID

FOREIGN KEY: supplierID REFERENCES SUPPLIER

CONSTRAINTS:

supplierContID SMALLINT PRIMARY KEY AUTO INCREMENT

supplierContName VARCHAR(20) NOT NULL

supplierContSurname VARCHAR(20) NOT NULL

supplierContEmail VARCHAR(50) NOTNULL CHECK(supplierContEmail LIKE '%@%.---' OR courEmail LIKE '%@%.---') DEFAULT 'xx@xx.xxx'

supplierID REFERENCES SUPPLIER(supplierID)

SUPPLIER_CONTACT_PHONE (suppContPhoID, suppContPhoTel, suppContPhoType, supplierContID)

PRIMARY KEY: suppContPhoID

FOREIGN KEY: supplierContID REFERENCES SUPPLIER_CONTACT

CONSTRAINTS:

suppContPhoID INT PRIMARY KEY AUTO INCREMENT

suppContPhoTel CHAR(18) NOT NULL CHECK(suppContPhoTel LIKE '(- -) - (- -) - (- -)') DEFAULT '(000)-(000)-(0000)'

suppContPhoType VARCHAR(4) NOT NULL

supplierContID REFERENCES SUPPLIER_CONTACT(supplierContID)

SUPPLEMENT (supID, supDescription, supMinLevel, supStockLevel, supStockLevelHeld, supNappiCode, supplierID)

PRIMARY KEY: supID

FOREIGN KEY: supplierID REFERENCES SUPPLIER

CONSTRAINTS:

supID SMALLINT PRIMARY KEY AUTO INCREMENT

supDescription VARCHAR(35) NOT NULL DEFAULT 'NONE'

supMinLevel INT NOT NULL CHECK(supMinLevel >= 0 AND supMinLevel <= 99999)

supStockLevel INT NOT NULL DEFAULT 0 CHECK(supStockLevel >= 0 AND supStockLevel <= 99999)

supStockLevelHeld INT NOT NULL CHECK(supStockLevelHeld >= 0 AND supStockLevelHeld <= 99999)

supNappiCode VARCHAR(7) NOT NULL DEFAULT '0000000'

supplierID REFERENCES SUPPLIER(supplierID)

SUPPLEMENT_COST (supCostID, supCostDate, supCostExc, supCostInc, supPercInc, supClientCost, supID)

PRIMARY KEY: supCostID

FOREIGN KEY: supID REFERENCES SUPPLEMENT

CONSTRAINTS:

supCostID INT PRIMARY KEY AUTO INCREMENT

supCostDate DATE NOT NULL

supCostExc DECIMAL(8,2) NOT NULL CHECK(supCostExc >= 0.00)

supCostInc AS (supCostExc * 1.14) NOT NULL CHECK(supCostInc >= 0.00)

supPercInc DECIMAL(8,2) NOT NULL CHECK(supPercInc >= 0.00)

supClientCost AS ((supCostExc * 1.14) + supPercInc)

supID REFERENCES SUPPLEMENT(supID)

EMPLOYEE (empID, empTitle, empName, empSurname, empAddressLine1, empAddressLine2, empAddressLine3, empAddressLine4, empPostCode, empTel, empCompanyEmail, jobID)

PRIMARY KEY: empID

FOREIGN KEY: jobID REFERENCES JOB

CONSTRAINTS:

empID INT PRIMARY KEY AUTO INCREMENT

empTitle VARCHAR(10) NOT NULL

empName VARCHAR(20) NOT NULL

empSurname VARCHAR(20) NOT NULL

empAddressLine1 VARCHAR(40) NOT NULL DEFAULT 'NONE'

empAddressLine2 VARCHAR(40) NOT NULL DEFAULT 'NONE'

empAddressLine3 VARCHAR(40) NOT NULL DEFAULT 'NONE'

empAddressLine4 VARCHAR(40) NOT NULL DEFAULT 'NONE'

empPostCode CHAR(4) NOT NULL DEFAULT '0000',

```
empTel CHAR(18) NOT NULL CHECK(courTel LIKE '--- --- -(----)')
DEFAULT '(000)-(000)-(0000)'

EmpCompanyEmail VARCHAR(50) NOT NULL CHECK(courEmail LIKE '%@%.---' OR
courEmail    LIKE '%@%.--.--') DEFAULT 'xx@xx.xxx'

jobID REFERENCES JOB(jobID)
```

CUSTOMER (cusID, cusName, cusSurname, cusHomeTel, cusWorkTel, cusCel, refID)

PRIMARY KEY: cusID

FOREIGN KEY: refID REFERENCES CUS_REFERENCE

CONSTRAINTS:

```
cusID BIGINT PRIMARY KEY AUTO INCREMENT

cusName VARCHAR(20) NOT NULL

cusSurname VARCHAR(20) NOT NULL

cusHomeTel CHAR(18) NOT NULL CHECK(cusHomeTel LIKE '--- --- -(----')
) DEFAULT '(000)-(000)-(0000)'

cusWorkTel CHAR(18) NOT NULL CHECK(cusWorkTel LIKE '--- --- -(----')
) DEFAULT '(000)-(000)-(0000)'

cusCel CHAR(18) NOT NULL CHECK(cusCel LIKE '--- --- -(----')
DEFAULT '(000)-(000)-(0000)'

refID REFERENCES CUS_REFERENCE (refID)
```

**CUS_ADDRESS (cusAddressID, cusAddressLine1, cusAddressLine2, cusAddressLine3,
cusAddressLine4, cusPostCode, cusID)**

PRIMARY KEY: cusAddressID

FOREIGN KEY: cusID REFERENCES CUSTOMER

CONSTRAINTS:

```
cusAddressID INT PRIMARY KEY AUTO INCREMENT

cusAddressLine1 VARCHAR(40) NOT NULL

cusAddressLine2 VARCHAR(40) NOT NULL DEFAULT 'NONE'

cusAddressLine3 VARCHAR(40) NOT NULL DEFAULT 'NONE'
```

```
cusAddressLine4 VARCHAR(40) NOT NULL DEFAULT 'NONE'  
cusPostCode CHAR(4) NOT NULL DEFAULT '0000',  
cusID REFERENCES CUSTOMER(cusID)
```

CUS_EMAIL (cusEmailID, cusEmailAddress, cusID)

PRIMARY KEY: cusEmailID

FOREIGN KEY: cusID REFERENCES CUSTOMER

CONSTRAINTS:

```
cusEmailID INT PRIMARY KEY AUTO INCREMENT
```

```
cusEmailAddress VARCHAR(50) NOT NULL CHECK(cusEmail LIKE '%@%.---' OR  
courEmail LIKE '%@%.---.') DEFAULT 'xx@xx.xxx'
```

```
cusID REFERENCES CUSTOMER(cusID)
```

INVOICE (invID, invDate, invTotalCost, bankID, cusID, empID)

PRIMARY KEY: invID

FOREIGN KEY: bankID REFERENCES BANK

FOREIGN KEY: cusID REFERENCES CUSTOMER

FOREIGN KEY: empID REFERENCES EMPLOYEE

CONSTRAINTS:

```
invID INT PRIMARY KEY AUTO INCREMENT
```

```
invDate DATE NOT NULL
```

```
invTotalCost DECIMAL(9,2) NOT NULL CHECK(invTotalCost >= 0.00)
```

```
bankID REFERENCES BANK(bankID)
```

```
cusID REFERENCES CUSTOMER(cusID)
```

```
empID REFERENCES EMPLOYEE(empID)
```

SALE (saleID, salePaymentAmt, salePaymentDate, saleStatus, invID)

PRIMARY KEY: saleID

FOREIGN KEY: invID REFERENCES INVOICE(invID)

CONSTRAINTS:

saleID INT PRIMARY KEY AUTO INCREMENT

salePaymentAmt DECIMAL(9,2) NOT NULL CHECK(salePaymentAmt >= 0.00) DEFAULT 0.00

salePaymentDate DATE NOT NULL CHECK(salePaymentDate >= invDate)

saleStatus VARCHAR(8) NOT NULL CHECK(saleStatus IN ('APPROVED', 'CANCELED', 'HISTORIC', 'PENDING')) DEFAULT 'PENDING'

invID REFERENCES INVOICE(invID)

INVOICE_ITEM (itmID, itmQty, itmSoldPrice, itmTotalPrice, supID, invID)

PRIMARY KEY: itmID

FOREIGN KEY: supID REFERENCES SUPPLEMENT

FOREIGN KEY: invID REFERENCES INVOICE

CONSTRAINTS:

itmID INT PRIMARY KEY AUTO INCREMENT

itmQty INT NOT NULL CHECK(itmQty >= 0 AND itmQty <= 99999)

itmSoldPrice DECIMAL(9,2) NOT NULL CHECK(itmSoldPrice >= 0.00)

itmTotalPrice AS (itmQty * itmSoldPrice) CHECK(itmTotalPrice >= 0.00)

supID REFERENCES SUPPLEMENT(supID)

invID REFERENCES INVOICE(invID)

RETURN_ITEM (rtnItemID, rtnItemDate, rtnItemQty, rtnItemReason, rtnItemCondition, itmID)

PRIMARY KEY: rtnItemID

FOREIGN KEY: itmID REFERENCES INVOICE_ITEM(itmID)

CONSTRAINTS:

rtnItemID INT PRIMARY KEY AUTO INCREMENT

rtnItemDate DATE NOT NULL

```
rtnItemQty INT NOT NULL CHECK (rtnItemQty >= 0 AND rtnItemQty <= 99999)
rtnItemReason VARCHAR(50) NOT NULL DEFAULT 'NONE'
rtnItemCondition BOOLEAN NOT NULL DEFAULT FALSE
itmID REFERENCES INVOICE_ITEM(itmID)
```

PURCHASE_JOURNAL (pjID, pjPurchaseDate, pjCostExc, pjCostInc, pjQty,
supplierID, supID)

PRIMARY KEY: pjID

FOREIGN KEY: supplierID REFERENCES SUPPLIER

FOREIGN KEY: supID REFERENCES SUPPLEMENT

CONSTRAINTS:

```
pjID INT PRIMARY KEY AUTO INCREMENT
```

```
pjPurchaseDate DATE NOT NULL
```

```
pjCostExc DECIMAL(8,2) NOT NULL CHECK(pjCostExc >= 0.00)
```

```
pjCostInc DECIMAL(8,2) AS (pjCostExc * 1.14) NOT NULL CHECK(pjCostInc  
>= 0.00)
```

```
pjQty INT NOT NULL CHECK (pjQty >= 0 AND pjQty <= 99999)
```

supplierID REFERENCES SUPPLIER(supplierID)

supID REFERENCES SUPPLEMENT(supID)

SHIPMENT (shipID, shipDateSent, courID, saleID)

PRIMARY KEY: shipID

FOREIGN KEY: courID REFERENCES COURIER

FOREIGN KEY: saleID REFERENCES SALE

CONSTRAINTS:

```
shipID INT PRIMARY KEY AUTO INCREMENT
```

```
shipDateSent DATE NOT NULL
```

courID REFERENCES COURIER(courID)

```
saleID REFERENCES SALE (saleID)
```

SQL Code

The SQL code to create the database and tables

```
DROP DATABASE IF EXISTS my_vitality;
CREATE DATABASE IF NOT EXISTS my_vitality;

DROP TABLE IF EXISTS PURCHASE_JOURNAL;
DROP TABLE IF EXISTS RETURN_ITEM;
DROP TABLE IF EXISTS INVOICE_ITEM;
DROP TABLE IF EXISTS SALE;
DROP TABLE IF EXISTS INVOICE;
DROP TABLE IF EXISTS CUS_PHONE;
DROP TABLE IF EXISTS CUSTOMER;
DROP TABLE IF EXISTS EMPLOYEE;
DROP TABLE IF EXISTS SUPPLEMENT_COST;
DROP TABLE IF EXISTS SUPPLEMENT;
DROP TABLE IF EXISTS SUPPLIER_CONTACT;
DROP TABLE IF EXISTS SUPPLIER_CONT_PHONE;
DROP TABLE IF EXISTS SUPPLIER;
DROP TABLE IF EXISTS CUS_REFERENCE;
DROP TABLE IF EXISTS COURIER;
DROP TABLE IF EXISTS JOB;
DROP TABLE IF EXISTS BANK;

CREATE TABLE BANK (
    bankID SMALLINT AUTO_INCREMENT,
    bankName VARCHAR(35) NOT NULL,
    bankBranchCode VARCHAR(6) NOT NULL DEFAULT '000000',
    bankAccountNumber VARCHAR(12) NOT NULL DEFAULT '000000000000',
    bankAccountType VARCHAR(12) NOT NULL DEFAULT 'NONE',
    CONSTRAINT bank_bankID_pk PRIMARY KEY(bankID),
    CONSTRAINT bank_bankAccountType_ck CHECK(bankAccountType IN ('CHEQUE',
    'CURRENT', 'TRANSMISSION', 'SAVINGS', 'MZANSI', 'NONE'))
);

CREATE TABLE JOB (
    jobID SMALLINT AUTO_INCREMENT,
    jobName VARCHAR(3) NOT NULL,
    CONSTRAINT job_jobID_pk PRIMARY KEY(jobID),
    CONSTRAINT job_jobName_ck CHECK(jobName IN ('HCP', 'GA', 'SU'))
);

CREATE TABLE COURIER (
    courID SMALLINT AUTO_INCREMENT,
    courName VARCHAR(35) NOT NULL,
    courTel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
    courEmail VARCHAR(50) NOT NULL DEFAULT 'xx@xx.xxxx',
    CONSTRAINT courier_courID_pk PRIMARY KEY(courID),
```

```

CONSTRAINT courier_courTel_ck CHECK(courTel LIKE '--- --- (----) '),
CONSTRAINT courier_courEmail_ck CHECK(courEmail LIKE '%@%.---' OR
courEmail LIKE '%@%.---')
);

CREATE TABLE CUS_REFERENCE (
refID SMALLINT AUTO_INCREMENT,
refName VARCHAR(30) NOT NULL,
CONSTRAINT cus_reference_refID_pk PRIMARY KEY(refID)
);

CREATE TABLE SUPPLIER (
supplierID SMALLINT AUTO_INCREMENT,
supplierName VARCHAR(35) NOT NULL,
supplierComments VARCHAR(200) NOT NULL DEFAULT 'NONE',
bankID SMALLINT,
CONSTRAINT supplier_supplierID_pk PRIMARY KEY(supplierID),
CONSTRAINT supplier_bankID_fk FOREIGN KEY (bankID) REFERENCES
BANK(bankID)
);

CREATE TABLE SUPPLIER_CONTACT (
supplierContID SMALLINT AUTO_INCREMENT,
supplierContName VARCHAR(20) NOT NULL,
supplierContSurname VARCHAR(20) NOT NULL DEFAULT 'NO SURNAME PROVIDED',
supplierContEmail VARCHAR(50) NOT NULL DEFAULT 'xx@xx.xxx',
supplierID SMALLINT,
CONSTRAINT supplier_contact_supplierContID_pk PRIMARY
KEY(supplierContID),
CONSTRAINT supplier_contact_supplierID_fk FOREIGN KEY(supplierID)
REFERENCES SUPPLIER(supplierID),
CONSTRAINT supplier_contact_supplierContEmail_ck CHECK(supplierContEmail
LIKE '%@%.---' OR courEmail LIKE '@%.---')
);

CREATE TABLE SUPPLIER_CONT_PHONE (
suppContPhoID INT AUTO_INCREMENT,
suppContPhoTel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
suppContPhoType VARCHAR(4) NOT NULL,
supplierContID SMALLINT,
CONSTRAINT suppler_cont_phone_suppContPhoID_pk PRIMARY
KEY(suppContPhoID),
CONSTRAINT suppler_cont_phone_supplierContID_fk FOREIGN
KEY(supplierContID) REFERENCES SUPPLIER_CONTACT(supplierContID),
CONSTRAINT suppler_cont_phone_suppContPhoTel_ck CHECK(suppContPhoTel LIKE
'--- --- (----) ')
);

CREATE TABLE SUPPLEMENT (
supID SMALLINT AUTO_INCREMENT,
supDescription VARCHAR(35) NOT NULL DEFAULT 'NONE',
supMinLevel INT NOT NULL,

```

```

supStockLevel INT NOT NULL,
supStockLevelHeld INT NOT NULL DEFAULT 0,
supNappiCode VARCHAR(7) NOT NULL DEFAULT '000000',
supplierID SMALLINT,
CONSTRAINT supplement_supID_pk PRIMARY KEY(supID),
CONSTRAINT supplement_supplierID_fk FOREIGN KEY(supplierID) REFERENCES SUPPLIER(supplierID),
CONSTRAINT supplement_supMinLevel_ck CHECK(supMinLevel >= 0 AND supMinLevel <= 99999),
CONSTRAINT supplement_supStockLevel_ck CHECK(supStockLevel >= 0 AND supStockLevel <= 99999),
CONSTRAINT supplement_supStockLevelHeld_ck CHECK(supStockLevelHeld >= 0 AND supStockLevelHeld <= 99999)
);

CREATE TABLE SUPPLEMENT_COST (
supCostID INT AUTO_INCREMENT,
supCostDate DATE NOT NULL,
supCostExc DECIMAL(8,2) NOT NULL,
supCostInc DECIMAL(8,2) GENERATED ALWAYS AS (supCostExc * 1.14) PERSISTENT,
supPercInc DECIMAL(8,2) NOT NULL,
supClientCost DECIMAL(8,2) GENERATED ALWAYS AS ((supCostExc * 1.14) + supPercInc) PERSISTENT,
supID SMALLINT,
CONSTRAINT supplement_cost_supCostID_pk PRIMARY KEY(supCostID),
CONSTRAINT supplement_cost_supID_fk FOREIGN KEY(supID) REFERENCES SUPPLEMENT(supID),
CONSTRAINT supplement_cost_supCostExc_ck CHECK(supCostExc >= 0.00),
CONSTRAINT supplement_cost_supCostInc_ck CHECK(supCostInc >= 0.00),
CONSTRAINT supplement_cost_supPercInc_ck CHECK(supPercInc >= 0.00),
CONSTRAINT supplement_cost_supClientCost_ck CHECK(supClientCost >= 0.00)
);

CREATE TABLE EMPLOYEE (
empID INT AUTO_INCREMENT,
empTitle VARCHAR(10) NOT NULL,
empName VARCHAR(20) NOT NULL,
empSurname VARCHAR(20) NOT NULL,
empAddressLine1 VARCHAR(40) NOT NULL DEFAULT 'NONE',
empAddressLine2 VARCHAR(40) NOT NULL DEFAULT 'NONE',
empAddressLine3 VARCHAR(40) NOT NULL DEFAULT 'NONE',
empAddressLine4 VARCHAR(40) NOT NULL DEFAULT 'NONE',
empPostCode CHAR(4) NOT NULL DEFAULT '0000',
empTel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
EmpCompanyEmail VARCHAR(50) NOT NULL DEFAULT 'xx@xx.xxx',
jobID SMALLINT,
CONSTRAINT employee_empID_pk PRIMARY KEY(empID),
CONSTRAINT employee_jobID_fk FOREIGN KEY(jobID) REFERENCES JOB(jobID),
CONSTRAINT employee_courTel_ck CHECK(courTel LIKE '--- --- ---'),
CONSTRAINT employee_courEmail_ck CHECK(courEmail LIKE '%@%.---' OR courEmail LIKE '%@%.---')
);

```

```

-- The majority of customerID's have a length of 13. Some records contain 11
and 12 characters
-- Data uploaded as BIGINT to allow for the large integer value needed to
store this large number
-- auto increment used to generate the number, the number will not go below
the last number created, so will not drop down to an 11 or 12-digit number
-- there are 88,909,205,916 vacant numbers before the last order and the
limit for 13 length digits.
-- From MariaDB 10.2.6 autoincrement columns are no longer permitted in CHECK
constraints, DEFAULT value expressions and virtual columns
-- above line from - https://mariadb.com/kb/en/library/auto_increment/#check-
constraints-default-values-and-virtual-columns
-- if necessary for a 13 length cusID, include in application code
CREATE TABLE CUSTOMER (
    cusID BIGINT AUTO_INCREMENT,
    cusName VARCHAR(20) NOT NULL,
    cusSurname VARCHAR(20) NOT NULL,
    cusHomeTel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
    cusWorkTel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
    cusCel CHAR(18) NOT NULL DEFAULT '(000)-(000)-(0000)',
    refID SMALLINT DEFAULT 7,
    CONSTRAINT customer_cusID_pk PRIMARY KEY(cusID),
    CONSTRAINT customer_refID_fk FOREIGN KEY(refID) REFERENCES
CUS_REFERENCE(refID),
    CONSTRAINT customer_cusHomeTel_ck CHECK(cusHomeTel LIKE '(- -) - (- -) - (- - -) '),
    CONSTRAINT customer_cusWorkTel_ck CHECK(cusWorkTel LIKE '(- -) - (- -) - (- - -) '),
    CONSTRAINT customer_cusCel_ck CHECK(cusCel LIKE '(- -) - (- -) - (- - -) ')
);

CREATE TABLE CUS_ADDRESS (
    cusAddressID INT AUTO_INCREMENT,
    cusAddressLine1 VARCHAR(40) NOT NULL,
    cusAddressLine2 VARCHAR(40) NOT NULL DEFAULT 'NONE',
    cusAddressLine3 VARCHAR(40) NOT NULL DEFAULT 'NONE',
    cusAddressLine4 VARCHAR(40) NOT NULL DEFAULT 'NONE',
    cusPostCode CHAR(4) NOT NULL DEFAULT '0000',
    cusID BIGINT,
    CONSTRAINT cus_address_cusAddressID_pk PRIMARY KEY(cusAddressID),
    CONSTRAINT cus_address_cusID_fk FOREIGN KEY(cusID) REFERENCES
CUSTOMER(cusID)
);

CREATE TABLE CUS_EMAIL (
    cusEmailID INT AUTO_INCREMENT,
    cusEmailAddress VARCHAR(50) NOT NULL DEFAULT 'xx@xx.xxxx',
    cusID BIGINT,
    CONSTRAINT cusEmail_cusEmailID_pk PRIMARY KEY(cusEmailID),
    CONSTRAINT cusEmail_cusID_fk FOREIGN KEY(cusID) REFERENCES
CUSTOMER(cusID),
    CONSTRAINT cusEmail_cusEmailAddress_ck CHECK(cusEmailAddress LIKE '%@%.-- -' OR cusEmailAddress LIKE '%@%.---.--')
);

```

```

CREATE TABLE INVOICE (
    invID INT AUTO_INCREMENT,
    invDate DATE NOT NULL,
    invTotalCost DECIMAL(9,2) NOT NULL,
    bankID SMALLINT,
    cusID BIGINT,
    empID INT,
    CONSTRAINT invoice_invID_pk PRIMARY KEY(invID),
    CONSTRAINT invoice_bankID_fk FOREIGN KEY(bankID) REFERENCES BANK(bankID),
    CONSTRAINT invoice_cusID_fk FOREIGN KEY(cusID) REFERENCES
    CUSTOMER(cusID),
    CONSTRAINT invoice_empID_fk FOREIGN KEY(empID) REFERENCES
    EMPLOYEE(empID),
    CONSTRAINT invoice_invTotalCost_ck CHECK(invTotalCost >= 0.00)
);

CREATE TABLE SALE (
    saleID INT AUTO_INCREMENT,
    salePaymentAmt DECIMAL(9,2) NOT NULL DEFAULT 0.00,
    salePaymentDate DATE NOT NULL,
    saleStatus VARCHAR(8) NOT NULL DEFAULT 'PENDING',
    invID INT,
    CONSTRAINT sale_saleID_pk PRIMARY KEY(saleID),
    CONSTRAINT sale_invID_fk FOREIGN KEY(invID) REFERENCES INVOICE(invID),
    CONSTRAINT sale_salePaymentAmt_ck CHECK(salePaymentAmt >= 0.00),
    CONSTRAINT sale_salePaymentDate_ck CHECK(salePaymentDate >= invDate),
    CONSTRAINT sale_saleStatus_ck CHECK(saleStatus IN ('APPROVED',
    'CANCELED', 'HISTORIC', 'PENDING'))
);

CREATE TABLE INVOICE_ITEM (
    itmID INT AUTO_INCREMENT,
    itmQty INT NOT NULL,
    itmSoldPrice DECIMAL(9,2) NOT NULL,
    itmTotalPrice DECIMAL(9,2) GENERATED ALWAYS AS (itmQty * itmSoldPrice)
PERSISTENT,
    supID SMALLINT,
    invID INT,
    CONSTRAINT invoice_item_itmID_pk PRIMARY KEY(itmID),
    CONSTRAINT invoice_item_supID_fk FOREIGN KEY(supID) REFERENCES
    SUPPLEMENT(supID),
    CONSTRAINT invoice_item_invID_fk FOREIGN KEY(invID) REFERENCES
    INVOICE(invID),
    CONSTRAINT invoice_item_itmQty_ck CHECK(itmQty >= 0 AND itmQty <= 99999),
    CONSTRAINT invoice_item_itmSoldPrice_ck CHECK(itmSoldPrice >= 0.00),
    CONSTRAINT invoice_item_itmTotalPrice_ck CHECK(itmTotalPrice >= 0.00)
);

CREATE TABLE RETURN_ITEM (
    rtnItemID INT AUTO_INCREMENT,
    rtnItemDate DATE NOT NULL,

```

```

rtnItemQty INT NOT NULL,
rtnItemReason VARCHAR(50) NOT NULL DEFAULT 'NONE',
rtnItemCondition BOOLEAN NOT NULL DEFAULT FALSE,
itmID INT,
CONSTRAINT return_item_rtnItemID_pk PRIMARY KEY(rtnItemID),
CONSTRAINT return_item_itmID_fk FOREIGN KEY(itmID) REFERENCES
INVOICE_ITEM(itmID),
CONSTRAINT return_item_rtnItemQty_ck CHECK (rtnItemQty >= 0 AND
rtnItemQty <= 99999)
);

CREATE TABLE PURCHASE_JOURNAL (
pjID INT AUTO_INCREMENT,
pjPurchaseDate DATE NOT NULL,
pjCostExc DECIMAL(8,2) NOT NULL,
pjCostInc DECIMAL(8,2) GENERATED ALWAYS AS (pjCostExc * 1.14) PERSISTENT,
pjQty INT NOT NULL,
supplierID SMALLINT,
supID SMALLINT,
CONSTRAINT purchase_journal_pjID_pk PRIMARY KEY(pjID),
CONSTRAINT purchase_journal_supplierID_fk FOREIGN KEY(supplierID)
REFERENCES SUPPLIER(supplierID),
CONSTRAINT purchase_journal_supID_fk FOREIGN KEY(supID) REFERENCES
SUPPLEMENT(supID),
CONSTRAINT purchase_journal_pjCostExc_ck CHECK(pjCostExc >= 0.00),
CONSTRAINT purchase_journal_pjCostInc_ck CHECK(pjCostInc >= 0.00),
CONSTRAINT purchase_journal_pjQty_ck CHECK (pjQty >= 0 AND pjQty <=
99999)
);

CREATE TABLE SHIPMENT (
shipID INT AUTO_INCREMENT,
shipDateSent DATE NOT NULL,
courID SMALLINT,
saleID INT,
CONSTRAINT shipment_shipID_pk PRIMARY KEY(shipID),
CONSTRAINT shipment_courID_fk FOREIGN KEY(courID) REFERENCES
COURIER(courID),
CONSTRAINT shipment_saleID_fk FOREIGN KEY(saleID) REFERENCES SALE(saleID)
);

-- create the roles for the staff of company
CREATE ROLE ga;
CREATE ROLE hcp;

-- grant the privileges to the roles

-- privileges for adding / updating / viewing stock levels and info
GRANT SELECT, INSERT, UPDATE
ON my_vitality.PURCHASE_JOURNAL
To ga, hcp;

```

```
GRANT SELECT, INSERT, UPDATE
ON my_vitality.SUPPLEMENT
To ga, hcp;

GRANT SELECT, INSERT, UPDATE
ON my_vitality.SUPPLEMENT_COST
To ga, hcp;

GRANT SELECT, INSERT, UPDATE
ON my_vitality.RETURN_ITEM
To ga, hcp;

-- privilege to authorize a transaction
GRANT SELECT, INSERT, UPDATE
ON my_vitality.SALE
To hcp;

-- privilege for creating, updating and viewing shipment info
-- Though this is a job for the GA, it is logical that the HCP would also be
able to complete this task if the GA was not available
GRANT SELECT, INSERT, UPDATE
ON my_vitality.SHIPMENT
To hcp;

-- privileges for viewing data from specific tables
GRANT SELECT
ON my_vitality.INVOICE
To ga, hcp;

GRANT SELECT
ON my_vitality.INVOICE_ITEM
To ga, hcp;

GRANT SELECT
ON my_vitality.CUSTOMER
To ga, hcp;

GRANT SELECT
ON my_vitality.CUS_ADDRESS
To ga, hcp;

GRANT SELECT
ON my_vitality.CUS_EMAIL
To ga, hcp;

GRANT SELECT
ON my_vitality.COURIER
To ga, hcp;

GRANT SELECT
ON my_vitality.SUPPLIER
To ga, hcp;
```

```
GRANT SELECT
ON my_vitality.SUPPLIER_CONTACT
TO ga, hcp;

GRANT SELECT
ON my_vitality.SUPPLIER_CONT_PHONE
TO ga, hcp;

-- privilege used to see the bank account info for sales and for purchases of
stock
GRANT SELECT
ON my_vitality.BANK
TO hcp;

-- privilege used for MIS purposes
GRANT SELECT
ON my_vitality.CUS_REFERENCE
TO hcp;

-- create users and grant roles to them

-- set passwords later when system functions correctly
-- naming convention rolename_uid_empID - where uid stands for user ID
CREATE USER hcp_uid_1; -- mariaDB does not support setting a password expire
option
CREATE USER ga_uid_2; -- mariaDB does not support setting a password expire
option

-- grant roles to users
GRANT hcp
TO hcp_uid_1;

GRANT ga
TO ga_uid_2;
```

Data Dictionary

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
BANK	bankID	ID for bank account entry	small int	#####	0 - 65535	AUTO INCREMENT	Y	PK	
	bankName	Name of bank	varchar (35)	XXXXXX	N/A		Y		
	bankBranchCode	Branch code for bank	varchar (6)	#####	10000 - 999999	000000	Y		
	bankAccountNumber	Account number	varchar (12)	#####	0000000000 to 9999999999	0000000000	Y		
	bankAccountType	Type of bank account	varchar (12)	XXXXXX	N/A	CHEQUE	Y		

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
JOB	jobID	Employees job ID	small int	##	1 - 65535	AUTO INCREMENT	Y	PK	
	jobName	Name of employee job	varchar (3)	XX	N/A		Y		

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
COURIER	courID	ID for courier	small int	##	1 - 65535	AUTO INCREMENT	Y	PK	
	courName	Name of Courier	varchar (35)	XXXXXX	N/A		Y		
	courTel	Telephone for courier	char(18)	(###)-(###)-####	N/A	(000)-(000)-(0000)	Y		
	courEmail	Email for courier	varchar (50)	xx@xx.xxx or xx@xx.xx.xx	N/A	xx@xx.xxx	Y		

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
CUS_REFERENCE	refID	ID for table	small int	###	1 - 65535	AUTO INCREMENT	Y	PK	
	refName	How the customer was referred (search engine, friend, social media, other, ...)	varchar (30)	Xxxxxxx	N/A		Y		

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SUPPLIER	supplierID	supplier ID	small int	#####	1 - 65535	AUTO INCREMENT	Y	PK	
	supplierName	name of supplier	varchar(35)	XXXXXXX	N/A		Y		
	supplierComments	comments about the supplier	varchar(200)	XXXXXXX	N/A	NONE	Y		
	bankID	bank ID	small int				Y	FK	BANK

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SUPPLIER_CONTACT	supplierContID	supplier contact ID	small int	#####	1 -65535	AUTO INCREMENT	Y	PK	
	supplierContName	contact first name	varchar(20)	Xxxxx	N/A	NO SURNAME PROVIDED	Y		
	supplierContSurame	contact last name	varchar(20)	Xxxxx	N/A	None	Y		
	supplierContEmail	email address	varchar(50)	xx@xx.xxx or xx@xx.xx.xx	N/A	xx@xx.xx.xx	Y		
	supplierID	supplier ID	small int				Y	FK	SUPPLIER

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SUPPLEMENT	supID	supplement ID	small int	#####	1 - 65535	AUTO INCREMENT	Y	PK	
	supDescription	supplement description	varchar(35)	XXXXXXX	N/A	NONE	Y		
	supMinLevel	Level of stock before re-order	int	#####	0 - 99999		Y		
	supStockLevel	Amount of stock available for sale	int	#####	0 - 99999		Y		
	supStockLevelHeld	Amount of stock being held for sale	int	#####	0 - 99999		Y		
	supNappiCode	Unique identifier for medical product	varchar(7)	XXXXXXX	100000 - 9999999	000000	Y		
	supplierID	supplier ID	small int				Y	FK	SUPPLIER

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SUPPLEMENT_COST	supCostID	ID for cost entry	int	###	1 - ...	AUTO INCREMENT	Y	PK	
	supCostDate	Stores the starting date for the prices	date	yy-mm-dd	NOT < CURRENT DATE		Y		
	supCostExc	Keep historical record of Cost Exc VAT	Decimal(8, 2)	#####.##	0.00 - 999999.99		Y		
	supCostInc	Keep historical record of Cost Exc VAT	supCostExc * 1.14	#####.##			Y		
	supCostPerInc	Keep historical record of Markup	Decimal(8, 2)	#####.##	0.00 - 999999.99		Y		
	supCostClientCost	Keep historical record of sold price	((supCostExc * 1.14) + supPerInc)	#####.##	0.00 - 999999.99		Y		
	supID	ID of supplement purchased	small int				Y	FK	SUPPLEMENT

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
EMPLOYEE	empID	Employee ID	int	###	1 - ...	AUTO INCREMENT	Y	PK	
	empTitle	Employee's title	varchar(10)	XXXXXX	N/A		Y		
	empName	First name	varchar (20)	XXXXXX	N/A		Y		
	empSurname	Surname	varchar (20)	XXXXXX	N/A		Y		
	empAddressLine1	First line of address	varchar (40)	XXXXXX	N/A	NONE	Y		
	empAddressLine2	Second line of address	varchar (40)	XXXXXX	N/A	NONE	Y		
	empAddressLine3	Third line of address	varchar (40)	XXXXXX	N/A	NONE	Y		
	empAddressLine4	Fourth line of address	varchar (40)	XXXXXX	N/A	NONE	Y		
	empPostCode	Postal Code	char(4)	####	0001 - 9999	0000	Y		
	empTel	Employee company telephone number	char(18)	(###)-(###)-(####)	N/A	(000)-(000)-(0000)	Y		
	empCompanyEmail	Employee company email address	varchar (50)	xx@xx.xxx or xx@xx.xx.xx	N/A	xx@xx.xxx	Y		
	jobID	Employees job ID	small int				Y	FK	JOB

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
CUSTOMER	cusID	Customer ID	big int	###	1 - ...	AUTO INCREMENT	Y	PK	
	cusName	Customers first name	varchar (20)	XXXXXX	N/A		Y		
	cusSurname	Customers surname	varchar (20)	XXXXXX	N/A		Y		
	cusTel	Telephone for customer	char (18)	(###)-(###)-(####)	N/A	(000)-(000)-(0000)	Y		
	cusWorkTel	Work telephone for customer	char (18)	(###)-(###)-(####)	N/A	(000)-(000)-(0000)	Y		
	cusCel	Cellphone for customer	char (18)	(###)-(###)-(####)	N/A	(000)-(000)-(0000)	Y		
	cusEmail	Email for customer	varchar (50)	xx@xx.xxx or xx@xx.xx.xx	N/A	xx@xx.xxx	Y		
	refID	How the customer was referred to company	small int	###	1 - 65535	7	Y		

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
CUS_ADDRESS	cusAddressID	customer address identifier	int	###	1 - ...	AUTO INCREMENT	Y	PK	
	cusAddressLine1	First line of address	VARCHAR(40)	XXXXXX	N/A		Y		
	cusAddressLine2	Second line of address	VARCHAR(40)	XXXXXX	N/A	NONE	Y		
	cusAddressLine3	Third line of address	VARCHAR(40)	XXXXXX	N/A	NONE	Y		
	cusAddressLine4	Fourth line of address	VARCHAR(40)	XXXXXX	N/A	NONE	Y		
	cusPostCode	Postal Code	CHAR(4)	####	0001 - 9999	0000	Y		
	cusID	Customer ID	BIG INT				Y	FK	CUSTOMER

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
CUS_EMAIL	cusEmailID	customer email identifier	INT	###	1 - ...	AUTO INCREMENT	Y	PK	
	cusEmailAddress	Customers email address	VARCHAR(50)	xx@xx.xxx or xx@xx.xx.xx	N/A	xx@xx.xxx	Y		
	cusID	Customer ID	BIG INT				Y	FK	CUSTOMER

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
INVOICE	invID	ID for their invoice	int	### but output to application as INV#####	0001 - ...	AUTO INCREMENT		PK	
	invDate	Date invoice generated	date	yy-mm-dd	NOT < CURRENT DATE		Y		
	intTotalCost	Total cost of the sale	Decimal	#####.##	0.00 - 9999999.98		Y		
	bankID	Account customers should pay into	small int	###	1 - 65535		Y	FK	BANK
	cusID	Customer who made the transaction	big int	###	1 - ...		Y	FK	CUSTOMER
	emplID	Employee ID who is listed on the invoice	int	###	1 - ...		Y	FK	EMPLOYEE

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SALE	saleID	The sale ID number	int	####	1 - ...	AUTO INCREMENT	Y	pk	
	saleStatus	Whether the sale has been approved	varchar(8)	XXXXXX		PENDING	Y		
	salePaymentAmount	The amount the customer paid	decimal	#####.##	0.00 - 9999999.98	0.00	Y		
	salePaymentDate	The date the customer paid	date	yy-mm-dd	N/A		Y		
	invID	the invoice associate with the Sale	int				Y	FK	INVOICE

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
INVOICE_ITEM	itmID	ID for one invoice line	int	####	1001 - ...	AUTO INCREMENT	Y	PK	
	itmQty	Amount of item purchased	int	####	1 - 99999		Y		
	itmSoldPrice	Keep historical record of price sold	Decimal (9, 2)	#####.##	0.00 - 999999.99		Y		
	ItmTotalPrice	Total cost of items purchased	itmQty * itmSoldPrice	#####.##	0.00 - 999999.99		Y		
	supID	ID of supplement purchased	small int				Y	FK	SUPPLEMENT
	invID	ID of associated invoice	int				Y	FK	INVOICE

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
RETURN_ITEM	rtnItemID	ID for item(s) returned	int	####	1 - ...	AUTO INCREMENT	Y		
	rtnItemDate	date of the return	date	yy-mm-dd	not < salePaymentDate		Y		
	rtnItemQty	the quantity of items returned	int	####	NOT < itmQty		Y		
	rtnItemReason	the reason for item returned	varchar(50)	XXXXXXX	N/A	NONE	Y		
	rtnItemCondition	determine if items can be restocked	Boolean	TRUE or FALSE	TRUE or FALSE	FALSE	Y		
	itmID	ID for associated invoice line	int				Y	FK	INVOICE_ITEM

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
PURCHASE_JOURNAL	pjID	Purchase Identifier	int	#####	1 - ...	AUTO INCREMENT	Y	PK	
	pjPurchaseDate	Date of purchase	date	yy-mm-dd	N/A		Y		
	pjCostExc	Cost paid for item excluding VAT	decimal	#####.##	0.00 - 99999.99		Y		
	pjCostInc	Cost paid for item including VAT	(pjCostExc * 1.14)	#####.##	0.00 - 99999.99		Y		
	pjQty	Quantity of item purchased	int	####	1 - 99999		Y		
	supplierID	supplier ID	small int				Y	FK	SUPPLIER
	supID	supplement ID	small int				Y	FK	SUPPLEMENT

TABLE	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	DEFAULT VALUE	REQUIRED	PK or FK	FK REFERENCED TABLE
SHIPMENT	shipID	ID of shipment	int	###	1 - ...		Y	PK	
	shipDateSent	date of when shipment was sent	date	yy-mm-dd	not < salePaymentDate		Y		
	courID	ID of couriers used	small int				Y	FK	COURIER

Screen shots of the database

phpMyAdmin

Structure SQL Search Query Export Operations Privileges Routines Events Triggers Tracking More

Table Action Rows Type Collation Size Overhead

Table	Action	Rows	Type	Collation	Size	Overhead
bank	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	16 KB	-
counter	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16 KB	-
customer	Browse Structure Search Insert Empty Drop	44	InnoDB	latin1_swedish_ci	96 KB	-
cus_address	Browse Structure Search Insert Empty Drop	439	InnoDB	latin1_swedish_ci	90 KB	-
cus_email	Browse Structure Search Insert Empty Drop	20	InnoDB	latin1_swedish_ci	32 KB	-
cus_reference	Browse Structure Search Insert Empty Drop	44	InnoDB	latin1_swedish_ci	16 KB	-
employee	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	2 KB	-
invoice	Browse Structure Search Insert Empty Drop	1,024	InnoDB	latin1_swedish_ci	200 KB	-
invoice_item	Browse Structure Search Insert Empty Drop	2,456	InnoDB	latin1_swedish_ci	20 KB	-
job	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 KB	-
purchase_journal	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	4 KB	-
return_item	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32 KB	-
sale	Browse Structure Search Insert Empty Drop	1,913	InnoDB	latin1_swedish_ci	144 KB	-
shipment	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	40 KB	-
supplement	Browse Structure Search Insert Empty Drop	24	InnoDB	latin1_swedish_ci	32 KB	-
supplement_cost	Browse Structure Search Insert Empty Drop	24	InnoDB	latin1_swedish_ci	32 KB	-
supplier	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32 KB	-
supplier_contact	Browse Structure Search Insert Empty Drop	7	InnoDB	latin1_swedish_ci	32 KB	-
supplier_cont_phone	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32 KB	-
Sum		7,120	InnoDB	latin1_swedish_ci	1.2 MB	0.8

Console Bookmarks Options History Clear

Showing rows 0 - 6 (7 total. Query took 0.0013 seconds.)

SELECT * FROM `bank`

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Options bankID bankName bankBranchCode bankAccountNumber bankAccountType

bankID	bankName	bankBranchCode	bankAccountNumber	bankAccountType
0	NO BANK DETAILS	000000	0000000000	NONE
1	ABSA	000000	45124561254	CHEQUE
2	STANDARD BANK	11813	01887092	CHEQUE
3	STANDARD BANK	23460	470315985	CHEQUE
4	CAPITEC	470010	4063452826	CHEQUE
5	ABSA	000000	4063467827	CHEQUE
6	ABSA	000000	4028764343	CHEQUE

Check all With selected Edit Copy Delete Export

Showing rows 0 - 5 (5 total. Query took 0.0013 seconds.)

SELECT * FROM `courier`

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Options courID courName courTel courEmail

courID	courName	courTel	courEmail
1	POST OFFICE SOUTH AFRICA	(012)-(407)-(7000)	customer.services@postoffice.co.za
2	POSTNET	(011)-(207)-(2900)	callcentre@postnetsa.co.za
3	EMS SOUTH AFRICA	(011)-(951)-(6000)	jmccustomerservices@postoffice.co.za
4	DHL EXPRESS	(011)-(921)-(3600)	xx@xx.xx
5	FEDEX EXPRESS SOUTH AFRICA	(087)-(742)-(8000)	xx@xx.xx
6	TNT	(086)-(012)-(2441)	xx@xx.xx

Check all With selected Edit Copy Delete Export

		cusID	cusName	cusSurname	cusHomeTel	cusWorkTel	cusCell	refID
<input type="checkbox"/>	Edit Copy Delete Details	47072706087	Chippa	Shango	(075)-(182)-(976) (051)-(534)-(9247) (081)-(241)-(0799)		16	
<input type="checkbox"/>	Edit Copy Delete Details	47120603088	Itumeleng	Makapela	(040)-(608)-(3750) (022)-(873)-(7626) (099)-(417)-(7121)		16	
<input type="checkbox"/>	Edit Copy Delete Details	82032206087	Wouter	Burger	(085)-(527)-(1284) (087)-(284)-(9774) (062)-(400)-(2276)		16	
<input type="checkbox"/>	Edit Copy Delete Details	151021036088	Shaun	Lesch	(028)-(699)-(2379) (071)-(870)-(0610) (033)-(657)-(6105)		13	
<input type="checkbox"/>	Edit Copy Delete Details	290611026087	Goratamang	Owens	(034)-(452)-(4033) (045)-(584)-(6194) (045)-(536)-(3745)		16	
<input type="checkbox"/>	Edit Copy Delete Details	341216047088	Sandiso	Naidoo	(038)-(261)-(4393) (010)-(680)-(1524) (033)-(458)-(6428)		16	
<input type="checkbox"/>	Edit Copy Delete Details	400422050085	Adriaan	Fourie	(090)-(748)-(3083) (094)-(931)-(1216) (084)-(115)-(5587)		16	
<input type="checkbox"/>	Edit Copy Delete Details	451201089085	Wisanzi	Naidoo	(025)-(886)-(9875) (017)-(791)-(5211) (036)-(720)-(4195)		16	
<input type="checkbox"/>	Edit Copy Delete Details	490705046088	Ayanda	Mlawuli	(083)-(260)-(2972) (022)-(883)-(8524) (091)-(257)-(9320)		7	
<input type="checkbox"/>	Edit Copy Delete Details	491010022086	Kivan	Dambuza	(066)-(940)-(2054) (015)-(865)-(4063) (029)-(665)-(0693)		16	
<input type="checkbox"/>	Edit Copy Delete Details	510523033088	Fazile	Nkabinde	(028)-(562)-(6165) (017)-(582)-(2477) (011)-(289)-(5994)		7	
<input type="checkbox"/>	Edit Copy Delete Details	510915039082	Christiaan	Malemene	(062)-(500)-(1913) (093)-(002)-(8011) (026)-(196)-(7457)		7	
<input type="checkbox"/>	Edit Copy Delete Details	511127046082	Edwin	Dlamini	(081)-(575)-(2418) (041)-(381)-(4846) (092)-(285)-(4922)		16	
<input type="checkbox"/>	Edit Copy Delete Details	521023092081	Edwin	Nkune	(074)-(978)-(8437) (025)-(336)-(4813) (033)-(646)-(1699)		13	
<input type="checkbox"/>	Edit Copy Delete Details	530628013085	Modisaotsile	Van Rensburg	(058)-(153)-(5944) (029)-(585)-(1426) (019)-(588)-(2915)		16	
<input type="checkbox"/>	Edit Copy Delete Details	531127094084	Shaun	Naidoo	(016)-(554)-(7714) (017)-(158)-(6412) (081)-(662)-(9948)		13	
<input type="checkbox"/>	Edit Copy Delete Details	580512061088	Bernhardt	Magutshwa	(078)-(471)-(8574) (049)-(195)-(2355) (056)-(458)-(1618)		16	
<input type="checkbox"/>	Edit Copy Delete Details	6307101012087	Francois	Mosehane	(036)-(100)-(8445) (014)-(544)-(2727) (044)-(670)-(8746)		16	
<input type="checkbox"/>	Edit Copy Delete Details	640428082086	Shane	Matshazini	(011)-(405)-(4850) (029)-(114)-(2425) (087)-(364)-(9226)		16	
<input type="checkbox"/>	Edit Copy Delete Details	650402056081	Ruduehwan	Khuluse	(001)-(620)-(1011) (013)-(327)-(1182) (071)-(816)-(8080)		13	
<input type="checkbox"/>	Edit Copy Delete Details	660620054086	Cliff	Kunane	(038)-(398)-(9879) (039)-(970)-(7777) (036)-(753)-(2557)		16	
Console								

		cusAddressID	cusAddressLine1	cusAddressLine2	cusAddressLine3	cusAddressLine4	cusPostCode	cusID
<input type="checkbox"/>	Edit Copy Delete Details	1	06 Baileybridge	Unit 9	Stonebridge	Phoenix	9068	6305240548084
<input type="checkbox"/>	Edit Copy Delete Details	2	1 Kliprivier Avenue	Secunda	Secunda	NONE	2302	2605150205088
<input type="checkbox"/>	Edit Copy Delete Details	3	11 Gallimble Avenue	Rooihuiskraal	NONE	NONE	0157	3602254350086
<input type="checkbox"/>	Edit Copy Delete Details	4	11 Mayibuya House	No 19326 Joburg Ivory Park Ext 12	NONE	NONE	1685	5206110625085
<input type="checkbox"/>	Edit Copy Delete Details	5	11 Rusteln Avenue	Ben Kamma	NONE	NONE	6025	9907200115082
<input type="checkbox"/>	Edit Copy Delete Details	6	1158 Manchester Crescent	Lanseria South	NONE	NONE	1829	500208660085
<input type="checkbox"/>	Edit Copy Delete Details	7	116 Selbourne Avenue	Parow	NONE	NONE	7500	100325239088
<input type="checkbox"/>	Edit Copy Delete Details	8	11399 Mphophoma Street	Klipfontein View	NONE	NONE	1685	5304200232082
<input type="checkbox"/>	Edit Copy Delete Details	9	12 Threevingergras	Dansville	NONE	NONE	0183	9706130545082
<input type="checkbox"/>	Edit Copy Delete Details	10	1206 Kuskorial Avenue	Moregoed	NONE	NONE	0186	4003100885084
<input type="checkbox"/>	Edit Copy Delete Details	11	129 Himalaya Street	Shellcross	NONE	NONE	9093	7300050161086
<input type="checkbox"/>	Edit Copy Delete Details	12	13 Garrick Road	University Estate	Woodstock	NONE	7525	2805200609084
<input type="checkbox"/>	Edit Copy Delete Details	13	13 Philips Road	Ecombe	NONE	NONE	9093	3410210326084
<input type="checkbox"/>	Edit Copy Delete Details	14	1359 Mahlangu Stand	Winterveldt	NONE	NONE	0198	2511120529083
<input type="checkbox"/>	Edit Copy Delete Details	15	138 Tarragon Two Road	3 Weltevredenpark	NONE	NONE	1709	5608010379088
<input type="checkbox"/>	Edit Copy Delete Details	16	15 Beaufort Place	39 Beaufort Avenue	Goodwood Park	NONE	7960	9003280238085
<input type="checkbox"/>	Edit Copy Delete Details	17	15 Circle Road	Capetown	Table View	NONE	7991	751103931085
<input type="checkbox"/>	Edit Copy Delete Details	18	15 Magnolia Way	Ridgeworth	Bellville	NONE	7530	4004220578084
<input type="checkbox"/>	Edit Copy Delete Details	19	15 Sekwinya	Seulsville	NONE	NONE	0125	8807130418086
<input type="checkbox"/>	Edit Copy Delete Details	20	1529 Khakhu Avenue	Daveyton	NONE	NONE	1520	8405280542088
Console								

+ Options		cusEmailID	cusEmailAddress	cusID	+ Options	
<input type="checkbox"/>		1	Mkuludi@mweb.co.za	6398240548084	<input type="checkbox"/>	
<input type="checkbox"/>		2	Moyahabo@mweb.co.za	3602250350086	<input type="checkbox"/>	
<input type="checkbox"/>		3	Mkuludi@btworld.co.za	5206110625085	<input type="checkbox"/>	
<input type="checkbox"/>		4	Tlou@btworld.co.za	5002080880085	<input type="checkbox"/>	
<input type="checkbox"/>		5	Brian@btworld.co.za	103320239088	<input type="checkbox"/>	
<input type="checkbox"/>		6	Nthatong@hotmail.com	5304280232082	<input type="checkbox"/>	
<input type="checkbox"/>		7	Josaya@gmail.com	9706130545082	<input type="checkbox"/>	
<input type="checkbox"/>		8	Mitasaka@mweb.co.za	2809260609084	<input type="checkbox"/>	
<input type="checkbox"/>		9	Seturuumanne@frs.co.za	3410210326084	<input type="checkbox"/>	
<input type="checkbox"/>		10	Tshilidzi@anglocoal.co.za	2511130529083	<input type="checkbox"/>	
<input type="checkbox"/>		11	Emil@mystudysmart.com	5610010370088	<input type="checkbox"/>	
<input type="checkbox"/>		12	Leashag@missionliving.co.za	8409280542088	<input type="checkbox"/>	
<input type="checkbox"/>		13	Ndlovuwo@actionford.co.za	2709110830083	<input type="checkbox"/>	
<input type="checkbox"/>		14	Selson@n-counter.com	6509050392085	<input type="checkbox"/>	
<input type="checkbox"/>		15	Nkhetheni@nashuaisp.co.za	6404210377085	<input type="checkbox"/>	
<input type="checkbox"/>		16	Louis@con.co.za	681014062082	<input type="checkbox"/>	
<input type="checkbox"/>		17	Shaun@intekom.co.za	531127094084	<input type="checkbox"/>	
<input type="checkbox"/>		18	Shephard@otreses.co.za	7805110427084	<input type="checkbox"/>	

- Server: 12/0.0.1 - Database: my_vitality - Table: employee

Employee Details													
+ Options		empID	empTitle	empName	empSurname	empAddressLine1	empAddressLine2	empAddressLine3	empAddressLine4	empPostCode	empTel	EmpCompanyEmail	jobID
<input type="checkbox"/>		1	MR	CASEY	MILLAN	NONE	NONE	NONE	NONE	0000	(082)-471-2929	cmlan@myvitality.co.za	1
<input type="checkbox"/>		2	MRS	JANE	DOE	NONE	NONE	NONE	NONE	0000	(082)-479-1234	jdoe@myvitality.co.za	2

+ Options		invID	invDate	invTotalCost	bankID	cusID	empID
<input type="checkbox"/>		1	2010-01-28	355.00	1	2012190284086	1
<input type="checkbox"/>		2	2010-04-07	1050.00	1	7309060944088	1
<input type="checkbox"/>		3	2010-05-06	3430.00	1	9003200235085	1
<input type="checkbox"/>		4	2010-07-10	455.00	1	7212170290085	1
<input type="checkbox"/>		5	2010-07-19	570.00	1	8807150418086	1
<input type="checkbox"/>		6	2010-08-01	540.00	1	4708110702087	1
<input type="checkbox"/>		7	2010-08-05	1220.00	1	8610120779086	1
<input type="checkbox"/>		8	2010-08-13	1390.80	1	9003200235085	1
<input type="checkbox"/>		9	2010-12-22	905.63	1	4104200140083	1
<input type="checkbox"/>		10	2011-01-01	1489.94	1	650402056081	1
<input type="checkbox"/>		11	2011-01-06	502.10	1	5212190473083	1
<input type="checkbox"/>		12	2011-01-11	2851.00	1	2703080954088	1
<input type="checkbox"/>		13	2011-01-11	252.14	1	2011240231088	1
<input type="checkbox"/>		14	2011-01-14	276.78	1	2808250506083	1
<input type="checkbox"/>		15	2011-01-16	273.90	1	4804160882084	1
<input type="checkbox"/>		16	2011-01-17	1067.22	1	2008190919082	1
<input type="checkbox"/>		17	2011-01-21	820.80	1	8612040918085	1
<input type="checkbox"/>		18	2011-01-28	467.90	1	5008230024086	1

Invoice Item

Options	itmID	itmQty	itmSoldPrice	itmTotalPrice	supID	invID
Edit Copy Delete	1	3	65.00	195.00	65	1
Edit Copy Delete	2	3	150.00	450.00	150	2
Edit Copy Delete	3	5	320.00	1600.00	104	3
Edit Copy Delete	4	1	65.00	65.00	32	4
Edit Copy Delete	5	2	135.00	270.00	70	5
Edit Copy Delete	6	4	110.00	440.00	24	6
Edit Copy Delete	7	2	90.00	180.00	107	7
Edit Copy Delete	8	3	150.00	450.00	145	8
Edit Copy Delete	9	5	121.13	605.65	181	9
Edit Copy Delete	10	4	236.60	946.40	197	10
Edit Copy Delete	11	5	100.42	502.10	172	11
Edit Copy Delete	12	3	221.34	664.02	132	12
Edit Copy Delete	13	2	125.07	252.14	225	13
Edit Copy Delete	14	3	92.26	276.78	86	14
Edit Copy Delete	15	3	81.30	243.90	84	15
Edit Copy Delete	16	1	203.02	203.02	182	16
Edit Copy Delete	17	4	205.20	820.80	214	17
Edit Copy Delete	18	5	93.58	467.90	32	18

Showing rows 0 - 1 (2 total). Query took 0.1302 seconds.

```
SELECT * FROM `invoice_item` WHERE invID = 1;
```

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Job

Options	jobID	jobName
Edit Copy Delete	1	HCP
Edit Copy Delete	2	GA
Edit Copy Delete	3	SU

Check all | With selected | [Edit](#) [Copy](#) [Delete](#) | [Export](#)

Supplier

Options	supID	supDescription	supMinLevel	supStockLevel	supStockLevelHeld	supNappiCode	supplierID
Edit Copy Delete	1	90	10	28	0	000000	1
Edit Copy Delete	2	60	1	41	0	000000	2
Edit Copy Delete	3	60	1	22	0	000000	3
Edit Copy Delete	4	60	10	13	0	000000	2
Edit Copy Delete	5	60	5	1	0	000000	2
Edit Copy Delete	6	60	5	36	0	000000	1
Edit Copy Delete	7	60	1	34	0	000000	3
Edit Copy Delete	8	60	1	27	0	000000	2
Edit Copy Delete	9	90	8	33	0	000000	1
Edit Copy Delete	10	180g powder	4	2	0	000000	3
Edit Copy Delete	11	NONE	7	49	0	000000	3
Edit Copy Delete	12	NONE	1	13	0	000000	6
Edit Copy Delete	13	NONE	8	44	0	000000	6
Edit Copy Delete	14	113g	6	37	0	000000	6
Edit Copy Delete	15	NONE	7	37	0	000000	6
Edit Copy Delete	16	NONE	5	25	0	000000	6
Edit Copy Delete	17	118ml	8	36	0	000000	6
Edit Copy Delete	18	30	3	38	0	000000	2

Supplier Cost Data								
	supCostID	supCostDate	supCostExc	supCostIncl	supPerInc	supClientCost	supID	Options
<input type="checkbox"/>	1	2018-04-01	313.00	356.82	30.00	386.82	1	
<input type="checkbox"/>	2	2018-04-01	216.00	245.10	40.00	265.10	2	
<input type="checkbox"/>	3	2018-04-01	216.81	247.16	35.00	282.16	3	
<input type="checkbox"/>	4	2018-04-01	222.00	253.08	15.00	268.08	4	
<input type="checkbox"/>	5	2018-04-01	201.00	225.14	40.00	269.14	5	
<input type="checkbox"/>	6	2018-04-01	259.00	295.26	20.00	315.26	6	
<input type="checkbox"/>	7	2018-04-01	221.00	251.94	35.00	286.94	7	
<input type="checkbox"/>	8	2018-04-01	171.00	194.94	40.00	234.94	8	
<input type="checkbox"/>	9	2018-04-01	326.00	371.64	15.00	386.64	9	
<input type="checkbox"/>	10	2018-04-01	195.06	223.28	60.00	283.28	10	
<input type="checkbox"/>	11	2018-04-01	147.85	168.55	60.00	228.55	11	
<input type="checkbox"/>	12	2018-04-01	262.00	298.68	15.00	313.68	12	
<input type="checkbox"/>	13	2018-04-01	136.00	155.04	15.00	170.04	13	
<input type="checkbox"/>	14	2018-04-01	278.13	317.07	25.00	342.07	14	
<input type="checkbox"/>	15	2018-04-01	330.00	376.20	15.00	391.20	15	
<input type="checkbox"/>	16	2018-04-01	245.00	279.30	15.00	294.30	16	
<input type="checkbox"/>	17	2018-04-01	230.33	262.58	25.00	287.58	17	
<input type="checkbox"/>	18	2018-04-01	153.00	174.42	40.00	214.42	18	

Supplier Details				
	supplierID	supplierName	supplierComments	bankID
<input type="checkbox"/>	1	SUPPLIER A	NONE	2
<input type="checkbox"/>	2	SUPPLIER B	NONE	3
<input type="checkbox"/>	3	SUPPLIER C	NONE	4
<input type="checkbox"/>	4	SUPPLIER D	NONE	0
<input type="checkbox"/>	5	SUPPLIER E	NONE	5
<input type="checkbox"/>	6	SUPPLIER F	NONE	6

Supplier Contact Data					
	supplierContID	supplierContName	supplierContSurname	supplierContEmail	supplierID
<input type="checkbox"/>	1	John	Adams	johnadams@supplier.co.za	1
<input type="checkbox"/>	2	Mary	Nkosi	mary@webmail.com	2
<input type="checkbox"/>	3	Ben	NO SURNAME PROVIDED	xx@xxx.xxx	3
<input type="checkbox"/>	4	Freddy	Neil	xx@xx.xx	4
<input type="checkbox"/>	5	Linda	NO SURNAME PROVIDED	supplier@absamail.co.za	5
<input type="checkbox"/>	6	John	NO SURNAME PROVIDED	johnmalan@mweb.co.za	6
<input type="checkbox"/>	7	Gert	NO SURNAME PROVIDED	johnmalan@mweb.co.za	6

Supplier Contact Phone Numbers				
	suppContPhoID	supplierContPhoTel	supComPhoType	supplierContID
<input type="checkbox"/>	1	(011)-(863)-(0056)	TEL	1
<input type="checkbox"/>	2	(011)-(863)-(0051)	FAX	1
<input type="checkbox"/>	3	(011)-(894)-(9004)	TEL	2
<input type="checkbox"/>	4	(011)-(894)-(9004)	TEL	3
<input type="checkbox"/>	5	(012)-(456)-(2346)	FAX	3
<input type="checkbox"/>	6	(081)-(345)-(1268)	CELL	4
<input type="checkbox"/>	7	(011)-(543)-(1136)	TEL	5
<input type="checkbox"/>	8	(012)-(766)-(3333)	TEL	6
<input type="checkbox"/>	9	(012)-(766)-(4102)	FAX	6

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	pjID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	pjPurchaseDate	date			No	None			Change Drop More
3	pjCostExc	decimal(8,2)			No	None			Change Drop More
4	pjCostInc	decimal(8,2)			Yes	None		PERSISTENT	Change Drop More
5	pjQty	int(11)			No	None			Change Drop More
6	supplierID	smallint(6)			Yes	None			Change Drop More
7	supID	smallint(5)			Yes	None			Change Drop More

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

Add 1 column(s) after supID

[Indexes](#)

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	pjID	0	A	No	
Edit Drop	purchase_journal_supplierID_fk	BTREE	No	No	supplierID	0	A	Yes	
Edit Drop	purchase_journal_supID_fk	BTREE	No	No	supID	0	A	Yes	

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	rtnItemID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	rtnItemDate	date			No	None			Change Drop More
3	rtnItemQty	int(11)			No	None			Change Drop More
4	rtnItemReason	varchar(50) latin_swedish_ci			No	NONE			Change Drop More
5	rtnItemCondition	tinyint(1)			No	0			Change Drop More
6	itmID	int(11)			Yes	None			Change Drop More

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

Add 1 column(s) after itmID

[Indexes](#)

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	rtnItemID	0	A	No	
Edit Drop	return_item_itmID_fk	BTREE	No	No	itmID	0	A	Yes	

[Table structure](#) [Relation view](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	shipID	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	shipDateSent	date			No	None			Change Drop More
3	courID	smallint(5)			Yes	None			Change Drop More
4	saleID	int(11)			Yes	None			Change Drop More

[Print](#) [Propose table structure](#) [Track table](#) [Move columns](#) [Normalize](#)

Add 1 column(s) after saleID

[Indexes](#)

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	shipID	0	A	No	
Edit Drop	shipment_courID_fk	BTREE	No	No	courID	0	A	Yes	
Edit Drop	shipment_saleID_fk	BTREE	No	No	saleID	0	A	Yes	

Your SQL query has been executed successfully.

```
show GRANTS for ga
```

+ Options
Grants for ga

```
GRANT USAGE ON *.* TO 'ga'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`supplement_cost` TO 'ga'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`purchase_journal` TO 'ga'  
GRANT SELECT ON `my_vitality`.`cus_address` TO 'ga'  
GRANT SELECT ON `my_vitality`.`invoice` TO 'ga'  
GRANT SELECT ON `my_vitality`.`supplier_contact` TO 'ga'  
GRANT SELECT ON `my_vitality`.`customer` TO 'ga'  
GRANT SELECT ON `my_vitality`.`invoice_item` TO 'ga'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`return_item` TO 'ga'  
GRANT SELECT ON `my_vitality`.`supplier` TO 'ga'  
GRANT SELECT ON `my_vitality`.`cus_email` TO 'ga'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`supplement` TO 'ga'  
GRANT SELECT ON `my_vitality`.`supplier_contact_phone` TO 'ga'  
GRANT SELECT ON `my_vitality`.`courier` TO 'ga'
```

Your SQL query has been executed successfully.

```
show GRANTS for hcp
```

+ Options
Grants for hcp

```
GRANT USAGE ON *.* TO 'hcp'  
GRANT SELECT ON `my_vitality`.`cus_address` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`courier` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`supplier_contact_phone` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`bank` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`cus_reference` TO 'hcp'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`purchase_journal` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`customer` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`invoice_item` TO 'hcp'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`supplier` TO 'hcp'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`supplier_contact` TO 'hcp'  
GRANT SELECT, INSERT, UPDATE ON `my_vitality`.`supplier_contact_phone` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`invoice` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`return_item` TO 'hcp'  
GRANT SELECT ON `my_vitality`.`supplement` TO 'hcp'
```

Assignment 5

MIS Reporting

Query writing

-- most commands are grouped into two sections

-- first section is for the MIS dashboard which gives a summary of data

-- second section is for the extended version of the dashboard where the user can get a detailed view of the data

-- dashboard provides reports relating to stock, sales, financial and pricing data

-- unless otherwise stated, the yearly dates reflect financial years - 01 March to 29 Feb

-- <http://www.sars.gov.za/ClientSegments/Individuals/Need-Help/Pages/Calender.aspx>

-- when using yearly data, application program can replace integer values and loop through the same command using counters and parameter values

-- screen shots taken on the 19th June 2018. For queries that use CURRENT_DATE, this figure would have been used to produce result

-- some queries when using dates, produce an empty result set or null value. The most recent invoice data is 28 June 217. So, queries relating to the last month or current financial year, do not have data to retrieve. These were included for future operation of system

-- Where the result set is greater in size than a single screen shot, I have included a screen shot of the total number of rows the query generated and a shot of the first page of results

```
-- Commands relating to Stock Information
```

```
-----  
-- MIS dashboard
```

```
-- get a list of supplements and its supplier that should be ordered due to  
low stock levels
```

```
SELECT supID, supplierName  
FROM SUPPLEMENT JOIN SUPPLIER USING(supplierID)  
WHERE supStockLevel < supMinLevel;
```

```
+ Options
```

supID	supplierName
34	SUPPLIER A
35	SUPPLIER A
71	SUPPLIER A
91	SUPPLIER A
96	SUPPLIER A
107	SUPPLIER A
124	SUPPLIER A
151	SUPPLIER A
180	SUPPLIER A
195	SUPPLIER A
217	SUPPLIER A
5	SUPPLIER B
45	SUPPLIER B
46	SUPPLIER B
66	SUPPLIER B
82	SUPPLIER B
108	SUPPLIER B
118	SUPPLIER B
143	SUPPLIER B
153	SUPPLIER B
186	SUPPLIER B
10	SUPPLIER C
37	SUPPLIER C
40	SUPPLIER C
109	SUPPLIER F
193	SUPPLIER F
207	SUPPLIER F
222	SUPPLIER F
227	SUPPLIER F

```
-- Detailed view
-- get the contact details for the suppliers of low stock items
-- query may return multiple contact options for each supplier. This gives
the receiver of the query the option to contact via telephone or fax, if both
are available
SELECT DISTINCT supplierName, supplierContName, supplierContSurname,
supplierContEmail, suppContPhoTel, suppContPhoType
FROM SUPPLEMENT JOIN SUPPLIER USING(supplierID)
JOIN SUPPLIER_CONTACT USING(supplierID)
JOIN SUPPLIER_CONT_PHONE USING(supplierContID)
WHERE supStockLevel < supMinLevel;
```

+ Options

supplierName	supplierContName	supplierContSurname	supplierContEmail	suppContPhoTel	suppContPhoType
SUPPLIER A	John	Adams	johnadams@suppliera.co.za	(011)-(863)-(0056)	TEL
SUPPLIER A	John	Adams	johnadams@suppliera.co.za	(011)-(863)-(0051)	FAX
SUPPLIER B	Mary	Nkosi	mary@webmail.com	(011)-(894)-(9004)	TEL
SUPPLIER C	Ben	NO SURNAME PROVIDED	xx@xx.xxx	(011)-(894)-(9004)	TEL
SUPPLIER C	Ben	NO SURNAME PROVIDED	xx@xx.xxx	(012)-(456)-(2346)	FAX
SUPPLIER F	John	NO SURNAME PROVIDED	johnmalan@mweb.co.za	(012)-(766)-(3333)	TEL
SUPPLIER F	John	NO SURNAME PROVIDED	johnmalan@mweb.co.za	(012)-(766)-(4102)	FAX
SUPPLIER F	Gert	NO SURNAME PROVIDED	johnmalan@mweb.co.za	(012)-(766)-(3333)	TEL
SUPPLIER F	Gert	NO SURNAME PROVIDED	johnmalan@mweb.co.za	(012)-(766)-(4102)	FAX

-- End of commands relating to Stock Information

-- Commands relating to sales information

-- MIS dashboard

-- summary of supplements sold during the last month

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)
GROUP BY supID;
```

-- No data stored in the database provided within last month. Query used for future operation of system

✓ MySQL returned an empty result set (i.e. zero rows). (

-- Detailed view

-- summary of supplements sold during previous financial years

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE
GROUP BY supID;
```

-- No data stored in the database provided within current financial year.

Query used for future operation of system

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29')
GROUP BY supID;
```

✓ Showing rows 0 - 49 (55 total, Query took 0.0067 seconds.)

supID	quantity
1	1
10	1
19	2
23	3
27	4
34	3
39	4
40	4
41	1
46	8
50	2
51	5
52	3
53	4
54	1
63	5
67	3
85	5
86	2
90	7
91	4
94	1
98	4
99	5
103	2
105	3
110	2
112	6
115	1
118	5
121	3
124	2
130	3
134	5
140	5
145	5
148	4
155	5
165	5
167	1
171	5
174	1
179	1
183	3
186	5
187	5
192	6
195	1
213	4
219	1

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29')
GROUP BY supID;
```

✓ Showing rows 0 - 49 (215 total, Query took 0.0044 seconds.)

supID	quantity
2	4
3	13
4	11
6	4
7	11
9	6
11	7
12	9
13	7
14	9
16	2
17	2
18	14
19	10
20	8
21	5
22	9
23	2
24	12
25	8
26	19
27	3
28	7
29	13
30	8
31	1
32	2
33	1
34	4
35	9
36	9
38	10
39	14
41	2
42	5
43	1
44	9
45	6
46	11
47	10
48	6
49	5
50	2
51	14
52	13
54	9
55	10
56	3
58	13
59	7

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29')
GROUP BY supID;
```

✓ Showing rows 0 - 49 (192 total. Query took 0.0041 seconds.)

supID	quantity
1	12
2	2
4	5
5	4
7	9
10	4
11	7
12	1
13	2
14	3
15	5
16	13
17	5
18	10
19	2
20	4
21	5
22	15
23	5
26	1
27	13
28	9
29	8
30	7
31	4
32	4
33	2
34	2
35	6
37	9
38	5
39	4
40	8
41	3
42	10
43	3
46	2
47	5
49	4
50	3
51	2
55	7
56	5
57	4
58	4
59	10
61	8
62	11
63	3
64	14

```
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29')
GROUP BY supID;
```

✓ Showing rows 0 - 49 (185 total. Query took 0.0048 seconds.)

supID	quantity
1	2
2	5
3	9
4	8
6	4
7	2
9	5
13	9
14	7
15	4
17	5
19	2
20	14
21	5
22	5
23	2
24	6
25	1
28	5
27	3
28	13
29	4
30	2
31	2
32	4
35	6
36	4
37	10
38	5
40	6
42	3
43	6
45	2
46	6
47	13
48	3
50	18
52	11
54	1
55	4
58	3
59	5
60	9
62	4
63	4
64	11
65	4
66	9
67	10
68	5

```
-- MIS dashboard
-- list of top ten supplements sold during the last month
-- query returns top ten values, not top ten rows
-- subquery returns the 10th value, using the offset parameter to the limit
clause
-- parent query uses value to retrieve all rows equal to or above that value
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)
GROUP BY supID
HAVING quantity >= (
    SELECT SUM(itemQty) AS quantity
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)
    GROUP BY supID
    ORDER BY quantity DESC LIMIT 9,1
)
ORDER BY quantity DESC;
```

MySQL returned an empty result set (i.e. zero rows). (

```
-- Detailed view
-- list of top ten supplements sold during the last year from current date
-- query returns top ten values, not top ten rows
-- subquery returns the 10th value, using the offset parameter to the limit
clause
-- parent query uses value to retrieve all rows equal to or above that value
SELECT supID, SUM(itemQty) AS quantity
FROM INVOICE_ITEM JOIN INVOICE USING(invID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)
GROUP BY supID
HAVING quantity >= (
    SELECT SUM(itemQty) AS quantity
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)
    GROUP BY supID
    ORDER BY quantity DESC LIMIT 9,1
)
ORDER BY quantity DESC;
```

supID	quantity
99	5
145	5
134	5
85	5
40	4
238	4
121	3
67	3
98	3
124	2

```
-- MIS dashboard
-- create a summary of supplements that have not sold during the last month
-- the EXCEPT and INTERSECT operators will be supported in version 10.3 of
MariaDB
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)
)
ORDER BY supID ASC;
-- returns all supplements as last invoice date is not in the last month
```

+ Options

	supID	▲ ▾
<input type="checkbox"/> Edit Copy Delete	1	
<input type="checkbox"/> Edit Copy Delete	2	
<input type="checkbox"/> Edit Copy Delete	3	
<input type="checkbox"/> Edit Copy Delete	4	
<input type="checkbox"/> Edit Copy Delete	5	
<input type="checkbox"/> Edit Copy Delete	6	
<input type="checkbox"/> Edit Copy Delete	7	
<input type="checkbox"/> Edit Copy Delete	8	
<input type="checkbox"/> Edit Copy Delete	9	
<input type="checkbox"/> Edit Copy Delete	10	
<input type="checkbox"/> Edit Copy Delete	11	
<input type="checkbox"/> Edit Copy Delete	12	
<input type="checkbox"/> Edit Copy Delete	13	
<input type="checkbox"/> Edit Copy Delete	14	
<input type="checkbox"/> Edit Copy Delete	15	
<input type="checkbox"/> Edit Copy Delete	16	
<input type="checkbox"/> Edit Copy Delete	17	
<input type="checkbox"/> Edit Copy Delete	18	
<input type="checkbox"/> Edit Copy Delete	19	
<input type="checkbox"/> Edit Copy Delete	20	
<input type="checkbox"/> Edit Copy Delete	21	
<input type="checkbox"/> Edit Copy Delete	22	
<input type="checkbox"/> Edit Copy Delete	23	
<input type="checkbox"/> Edit Copy Delete	24	
<input type="checkbox"/> Edit Copy Delete	25	

```
-- Detailed view
-- create a summary of any supplements that have not sold during previous
financial years
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND
CURRENT_DATE
)
ORDER BY supID ASC;
```

-- no supplements sold during this period. Last invoice date is 51 weeks old

✓ Showing rows 0 - 24 (245 total. Query took 0.0049 seconds.) [supID: 1... - 25...]

```
SELECT supID FROM SUPPLEMENT WHERE NOT EXISTS ( SELECT supID FROM INVOICE_ITEM JOIN
```

1 ▾ > >> | Show all | Number of rows: 25 ▾ Filter

supID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

```
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29')
)
ORDER BY supID ASC;
-- all supplements were sold during this period
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29')
)
ORDER BY supID ASC;
-- all supplements were sold during this period
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29')
)
ORDER BY supID ASC;
-- all supplements were sold during this period
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID
FROM SUPPLEMENT
WHERE NOT EXISTS (
    SELECT supID
    FROM INVOICE_ITEM JOIN INVOICE USING(invID)
    WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29')
)
ORDER BY supID ASC;
-- all supplements were sold during this period
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
-- MIS dashboard
-- number of orders waiting for payment approval
SELECT COUNT(invID) AS "Orders Pending Approval"
FROM sale
WHERE saleStatus = 'PENDING';
-- no orders pending approval. Invoice data is 51 weeks old. Invoice data
imported, stored as HISTORIC in sale table as there is no way to tell if the
order has been paid, rejected, or in process of being sold. Any new order in
the system will not be marked Historic.
```

Orders Pending Approval

0

```
-- Detailed view
-- summary of invoices that needs approval
-- sort by the earliest date
SELECT s.invID, i.invDate, i.invTotalCost
FROM sale s JOIN invoice i USING (invID)
WHERE saleStatus = 'PENDING'
ORDER BY i.invDate;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
-- MIS dashboard
-- number of orders in the last month
SELECT COUNT(invID)
FROM INVOICE
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);
-- No data stored in the database provided within last month. Query used for
future operation of system
```

COUNT(invID)

0

```
-- Detailed view
-- number of orders during previous financial years
SELECT COUNT(invID)
FROM INVOICE
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE;
COUNT(invID)
0

SELECT COUNT(invID)
FROM INVOICE
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');
COUNT(invID)
41

SELECT COUNT(invID)
FROM INVOICE
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29');
COUNT(invID)
351

SELECT COUNT(invID)
FROM INVOICE
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29');
COUNT(invID)
237

SELECT COUNT(invID)
FROM INVOICE
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29');
COUNT(invID)
230
```

```
-- MIS dashboard
-- number of orders cancelled in the last month

-- status of sale is stored using the following four words - ('APPROVED',
'CANCELED', 'HISTORIC', 'PENDING')
-- Invoice data is 51 weeks old. Invoice data imported is stored as HISTORIC
in sale table as there is no way to tell if the order has been paid,
rejected, or in process of being sold. Any new order in the system will not
be marked Historic. All following queries return 0, but in future operation
of system, sales can be logged as cancelled, for example when payment is not
received.
```

```
SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);
Cancel
0

-- Detailed view
-- number of orders cancelled during previous financial years
SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND
CURRENT_DATE;
Cancel
0

SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1
YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');
Cancel
0

SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-
02-29');
Cancel
0

SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-
02-29');
Cancel
0

SELECT COUNT(saleID) AS Cancel
FROM sale
WHERE saleStatus = 'CANCELED'
AND salePaymentDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-
02-29');
```

```
-- MIS dashboard  
-- the following queries relate to supplements returned all return an empty  
result set. The table for returned supplements has been created for the  
future operation of system and it contains no data before the system becomes  
operational
```

```
-- the commands for the previous 5 financial years will come in operation as  
the system becomes older
```

```
-- create a summary of supplements returned in last month
```

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)  
WHERE rtnItemDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)  
GROUP BY supID  
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
-- Detailed view
```

```
-- create a summary of supplements returned in previous financial years
```

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)  
WHERE rtnItemDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND  
CURRENT_DATE  
GROUP BY supID  
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)  
WHERE rtnItemDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1  
YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29')  
GROUP BY supID  
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)  
WHERE rtnItemDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2  
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1  
YEAR)), '-02-29')  
GROUP BY supID  
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)  
WHERE rtnItemDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3  
YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2  
YEAR)), '-02-29')  
GROUP BY supID  
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows). (

```
SELECT supID, COUNT(supID) AS returned  
FROM INVOICE_ITEM II JOIN RETURN_ITEM RI USING(itemID)
```

```
WHERE rtnItemDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29')
GROUP BY supID
ORDER BY returned DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows).

```
-- MIS dashboard
-- create a summary of the supplements sold in the last month, categorized by supplier
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
    JOIN INVOICE USING (invID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH)
GROUP BY supplierID
ORDER BY quantity DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows).

```
-- Detailed view
-- create a summary of the supplements sold in previous financial years,
categorized by supplier
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
    JOIN INVOICE USING (invID)
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE
GROUP BY supplierID
ORDER BY quantity DESC;
```

✓ MySQL returned an empty result set (i.e. zero rows).

```
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
    JOIN INVOICE USING (invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29')
GROUP BY supplierID
ORDER BY quantity DESC;
```

+ Options

supplierID	quantity
1	33
2	19
3	6
5	5
6	3
4	1

```
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
JOIN INVOICE USING (invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29')
GROUP BY supplierID
ORDER BY quantity DESC;
```

supplierID	quantity
1	260
2	184
6	65
3	51
5	19
4	12

```
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
JOIN INVOICE USING (invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29')
GROUP BY supplierID
ORDER BY quantity DESC;
```

supplierID	quantity
2	121
1	118
6	49
3	38
5	10
4	7

```
SELECT supplierID, COUNT(itemQty) AS quantity
FROM SUPPLEMENT JOIN INVOICE_ITEM USING(supID)
JOIN INVOICE USING (invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29')
GROUP BY supplierID
ORDER BY quantity DESC;
```

supplierID	quantity
2	128
1	117
3	44
6	34
5	12
4	6

```
-- MIS dashboard
-- create a regional breakdown of where the products are being delivered /
sold
-- invoice order used to count number of orders - in future expansion of
system, a customer may have more than one order
-- Northern region
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) BETWEEN 0001 AND 2899;
quantity
1041

-- Eastern region
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) BETWEEN 2900 AND 4730;
quantity
48

-- Southern region
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) BETWEEN 4731 AND 6499;
quantity
56

-- Western region
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) BETWEEN 6500 AND 8299;
quantity
206

-- Central region
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) BETWEEN 8300 AND 9999;
quantity
144

-- No region given
SELECT COUNT(invID) AS quantity
FROM INVOICE JOIN CUS_ADDRESS USING(cusID)
WHERE CONVERT(cusPostCode, INTEGER) = 0000;
quantity
21
```

```
-- MIS dashboard
-- Create a summary of how customers are being referred
SELECT cr.refName, COUNT(c.refID) AS quantity
FROM CUS_REFERENCE cr JOIN CUSTOMER c USING(refID)
GROUP BY cr.refName
ORDER BY quantity DESC;

```

refName	quantity
WORD OF MOUTH	252
OTHER	97
WEBSITE	71
MYSELF	13
MOTHER-IN-LAW	7
QWT	4

```
-- End of commands relating to sales information
-- commands relating to financial data
-----
-- MIS dashboard
-- total value of sales for current month
-- data only included if sale has been approved or is recorded as historic
SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
      invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);

```

val
NULL

```
-- Detailed view
-- total value of sales for the previous financial years
SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
      invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE;

```

val
NULL

```
SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
      invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');

```

val
66,480.92

```
SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
      invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29');

```

val
482,375.31

```
SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
    invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29');
val
311,938.38

SELECT FORMAT(SUM(salePaymentAmt),2) AS val
FROM SALE JOIN INVOICE USING(invID)
WHERE saleStatus IN('APPROVED', 'HISTORIC') AND
    invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29');
val
242,026.21
-----
-- MIS dashboard
-- total profit for the month
SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);
val
NULL

-- Detailed view
-- -- total profit for previous financial years
SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE;
val
NULL

SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');
profit
1,967.00

SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29');
profit
15,797.00
```

```
SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29');
profit
10,073.00
```

```
SELECT FORMAT(SUM(supPercInc), 2) AS profit
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29');
profit
10,850.00
```

```
-- MIS dashboard
-- total tax for the month
SELECT FORMAT(SUM(supCostInc - supCostExc), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);
val
NULL
```

```
-- Detailed view
-- -- total tax for previous financial years
SELECT FORMAT(SUM(supCostInc - supCostExc), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE;
val
NULL
```

```
SELECT FORMAT(SUM(supCostInc - supCostExc), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');
tax
2,716.57
```

```

SELECT FORMAT(SUM(supCostInc - supCostExc ), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29');
tax
21,502.07

```

```

SELECT FORMAT(SUM(supCostInc - supCostExc ), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29');
tax
12,283.26

```

```

SELECT FORMAT(SUM(supCostInc - supCostExc ), 2) AS tax
FROM SALE JOIN INVOICE USING(invID)
    JOIN INVOICE_ITEM USING(invID)
    JOIN SUPPLEMENT_COST USING(supID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29');
tax
10,635.93
-----
```

```

-- MIS dashboard
-- calculate the average price paid for an order this month
SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate >= DATE_SUB(CURRENT_DATE, INTERVAL 1 MONTH);
val
NULL

```

```

-- Detailed view
-- calculate the average price paid for an order for previous financial years
SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(CURRENT_DATE), '-03-01') AND CURRENT_DATE;
val
NULL

```

```

SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-03-01') AND CONCAT(YEAR(CURRENT_DATE), '-02-29');
avgPrice
1,621.49

```

```
SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)), '-02-29');
avgPrice
1,374.29
```

```
SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 2 YEAR)), '-02-29');
avgPrice
1,316.20
```

```
SELECT FORMAT(AVG(invTotalCost), 2) AS avgPrice
FROM SALE JOIN INVOICE USING(invID)
WHERE invDate BETWEEN CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 4 YEAR)), '-03-01') AND CONCAT(YEAR(DATE_SUB(CURRENT_DATE, INTERVAL 3 YEAR)), '-02-29');
avgPrice
1,052.29
```

-- End of commands relating to financial information

-- Start of commands relating to pricing information

```
-- MIS dashboard
-- show the products with the lowest profit markup
SELECT supID, supPercInc AS "Lowest Mark Up"
FROM supplement_cost
WHERE supPercInc = (
    SELECT MIN(supPercInc)
    FROM supplement_cost);
-----
```

supID	Lowest Mark Up
34	0.00
35	0.00
88	0.00
100	0.00
103	0.00
111	0.00
113	0.00
188	0.00
193	0.00
207	0.00
214	0.00
226	0.00
236	0.00
237	0.00
245	0.00

```
-- show the products with the highest profit markup
SELECT supID, supPercInc AS "Highest Mark Up"
FROM supplement_cost
WHERE supPercInc = (
    SELECT MAX(supPercInc)
    FROM supplement_cost);
```

supID	Highest Mark Up
10	60.00
11	60.00
24	60.00
37	60.00
47	60.00
50	60.00
59	60.00
119	60.00
140	60.00
148	60.00

```
-- Detailed view
-- show the products with the lowest profit markup and the qty sold since the
start of records
SELECT supID, supPercInc AS "Lowest Mark Up", COUNT(itemQty) AS "Quantity
Sold"
FROM supplement_cost JOIN INVOICE_ITEM USING(supID)
WHERE supPercInc = (
    SELECT MIN(supPercInc)
    FROM supplement_cost)
GROUP BY supID;
```

supID	Lowest Mark Up	Quantity Sold
34	0.00	4
35	0.00	9
88	0.00	13
100	0.00	13
103	0.00	12
111	0.00	7
113	0.00	7
188	0.00	10
193	0.00	15
207	0.00	12
214	0.00	11
226	0.00	12
236	0.00	91
237	0.00	7
245	0.00	8

-- show the products with the highest profit markup and the qty sold since the start of records

```
SELECT supID, supPercInc AS "Highest Mark Up", COUNT(itemQty) AS "Quantity Sold"
FROM supplement_cost JOIN INVOICE_ITEM USING(supID)
WHERE supPercInc = (
    SELECT MAX(supPercInc)
    FROM supplement_cost)
GROUP BY supID;
```

supID	Highest Mark Up	Quantity Sold
10	60.00	10
11	60.00	11
24	60.00	13
37	60.00	11
47	60.00	11
50	60.00	13
59	60.00	18
119	60.00	17
140	60.00	12
148	60.00	7

-- Calculate the average profit of products sold by supplier and the standard deviation of the profit for each supplier

```
SELECT s.supplierID, COUNT(*) AS "Number of Products", AVG(sc.supPercInc) AS "Average Profit", STDDEV(sc.supPercInc) AS "Standard Deviation"
FROM supplement s JOIN supplement_cost sc USING(supID)
GROUP BY supplierID;
```

supplierID	Number of Products	Average Profit	Standard Deviation
1	95	22.789474	8.294443
2	83	37.831325	6.169922
3	25	45.600000	14.650597
4	5	36.000000	4.898979
5	9	34.444444	17.069213
6	28	17.714286	15.082765

Assignment 6

Analysis and Design Documentation

Screenshots

Screenshot of site are taken for the following devices using the Mozilla Firefox Developer Window:

- Laptop (1280 x 720 @ 1.5dppx)
- Samsung Galaxy S7

Lorem text is currently being used as placeholder text while the site and content is being developed.

Firefox allows full page screen shots to be taken so that the whole screen is available and removes the need for scrolling.

The following two screenshots are for the home page for the Online Store.

The same format has also been used for:

The about us page, services page and the individual treatments pages.

The screenshot shows the homepage of the My Vitality website. At the top, there's a dark header with the logo 'MYVITALITY' and a teal shopping cart icon showing '0 items'. Below the header is a teal navigation bar with links: Home, About, Services, Store, and Contact. The main title 'A NATURAL WAY TO IMPROVE YOUR HEALTH' is centered above a text block. The text block contains placeholder text (Lorem ipsum) and a link to 'Read more'. Below the main title are three sections: 'SHOP ONLINE' with a shopping cart icon, 'SERVICES' with a heart icon, and 'CONTACT' with a phone icon. Each section has a short paragraph of placeholder text. At the bottom, there's a dark footer with sections for 'Get to know us', 'Follow us' (with social media icons for Facebook, Twitter, and YouTube), and 'Let us help you' (with links to FAQ, Terms and conditions, and Return policies). Copyright information and page numbers are also present.

MYVITALITY
ALTERNATIVE HEALTHCARE

Cart (0 items)

Home About Services Store Contact

A NATURAL WAY TO IMPROVE YOUR HEALTH

Read more

SHOP ONLINE

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Saepe ea temporibus, veniam consectetur natus, aspernatur consequatur dicta quaerat quo nihil nisi recusandae harum fugit reiciendis debitis accusamus eaque error quidem corrupti aperiam maiores officiis unde, eius praesentium. Obcaecati quaerat sapiente numquam cum officiis, deleniti labore mollitia nesciunt hic facilis. Perspiciatis?

SERVICES

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Esse saepe, id, numquam quod suscipit sit porro! Rerum esse pariatur, ipsum debitis! Delectus neque illo impedit hic quasi ipsa sapiente laudantium repudiandae quea, ipsum aspernatur consectetur architecto saepe sed commodi maiores.

CONTACT

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Esse saepe, id, numquam quod suscipit sit porro! Rerum esse pariatur, ipsum debitis! Delectus neque illo impedit hic quasi ipsa sapiente laudantium repudiandae quea, ipsum aspernatur consectetur architecto saepe sed commodi maiores.

Get to know us

[Who are we](#)
[Where are we](#)
[Contact us](#)

Follow us

Copyright My Vitality © 2018

Let us help you

[FAQ](#)
[Terms and conditions](#)
[Return policies](#)

Page | 95

 **MYVITALITY**
ALTERNATIVE HEALTH CARE

Cart (0 items)

Home About Services Store Contact

A NATURAL WAY TO IMPROVE YOUR HEALTH

Lore ipsum dolor sit amet, consectetur adipisciing elit. Saepe ea temporibus, veniam consectetur natus, aspernatur consequatur dicta quaerat quo nihil nisi recusandae harum fugit reiciendis debitis accusamus eaque error quidem corrupti aperiam maiores officiis unde, eius praesentium. Obcaecati quaerat sapiente numquam cum officiis, deleniti labore mollitia nesciunt hic facilis. Perspiciatis? [Read more](#)

SHOP ONLINE



Lore ipsum dolor sit amet, consectetur adipisciing elit. Esse saepe, id, numquam quod suscipit sit porro! Rerum esse pariatur, ipsum debitis! Delectus neque illo impedit hic quasi ipsa sapiente laudantium repudiandae que, ipsum aspernatur consectetur architecto saepe sed commodi maiores.

SERVICES



Lore ipsum dolor sit amet, consectetur adipisciing elit. Esse saepe, id, numquam quod suscipit sit porro! Rerum esse pariatur, ipsum debitis! Delectus neque illo impedit hic quasi ipsa sapiente laudantium repudiandae que, ipsum aspernatur consectetur architecto saepe sed commodi maiores.

CONTACT



Lore ipsum dolor sit amet, consectetur adipisciing elit. Esse saepe, id, numquam quod suscipit sit porro! Rerum esse pariatur, ipsum debitis! Delectus neque illo impedit hic quasi ipsa sapiente laudantium repudiandae que, ipsum aspernatur consectetur architecto saepe sed commodi maiores.

Follow us

Get to know us

[Who are we](#)
[Where are we](#)
[Contact us](#)

Let us help you

[FAQ](#)
[Terms and conditions](#)
[Return policies](#)

Copyright My Vitality © 2018

Supplements page

The screenshot shows a grid of supplier logos and product details from the MyVitality website. The grid is organized into several rows:

- Row 1:** Supplier A (circle logo), Supplier B (triangle logo), Supplier B (triangle logo), Supplier F (flower logo).
- Row 2:** R233.70 ID:236 Qty(26) | R234.94 ID:127 Qty(16) | R302.46 ID:157 Qty(28) | R2761.12 ID:112 Qty(34)
- Row 3:** Supplier B (triangle logo), Supplier B (triangle logo), Supplier B (triangle logo), Supplier E (green person logo).
- Row 4:** R249.76 ID:70 Qty(26) | R268.08 ID:4 Qty(13) | R238.36 ID:46 Qty(3) | R219.64 ID:52 Qty(33)
- Row 5:** Supplier C (person logo), Supplier B (triangle logo), Supplier F (flower logo), Supplier A (circle logo).
- Row 6:** R258.36 ID:59 Qty(39) | R244.06 ID:176 Qty(11) | R252.61 ID:225 Qty(31) | R388.92 ID:231 Qty(46)
- Row 7:** Supplier F (flower logo), Supplier F (flower logo), Supplier B (triangle logo), Supplier C (person logo).
- Row 8:** R150.48 ID:226 Qty(36) | R266.76 ID:193 Qty(2) | R214.42 ID:18 Qty(38) | R263.42 ID:119 Qty(50)
- Row 9:** Supplier E (green person logo), Supplier F (flower logo), Supplier A (circle logo), Supplier C (person logo).
- Row 10:** R207.08 ID:55 Qty(48) | R348.87 ID:88 Qty(17) | R278.08 ID:145 Qty(40) | R311.36 ID:93 Qty(36)

Footer:

- Get to know us:** Who are we? Where are we? Contact us.
- Follow us:** Facebook, Twitter, YouTube.
- Let us help you:** FAQ, Terms and conditions, Return policies.

Copyright My Vitality © 2018

Note: Sort by code change to 3 items for display purposes only. Screenshot shows 20 items.

The screenshot shows the MyVitality website's supplier search page. At the top, there is a header with the logo 'MYVITALITY ALTERNATIVE HEALTH CARE' and a 'Cart (0 items)' link. Below the header is a navigation menu with links to 'Home', 'About', 'Services', 'Store', and 'Contact'. The main title 'SUPPLIER' is displayed in large, bold, black capital letters. Below the title are two dropdown menus: 'Sort by' set to 'Popular' and 'Show' set to '20 items'. A prominent orange 'SEARCH' button is located below the dropdowns. The background of the page is light grey.



A product card for 'Supplier A' item 60. The card features a small circular icon with the number '60' in the center. Below the icon, the price 'R233.70', ID 'ID:236', and quantity 'Qty(26)' are listed. The background of the card is light grey.



SUPPLIER B

A product card for 'Supplier B' item 90. The card features a small circular icon with the number '90' in the center. Below the icon, the price 'R234.94', ID 'ID:127', and quantity 'Qty(16)' are listed. The background of the card is light grey.



SUPPLIER B

The footer section of the MyVitality website includes social media links for Facebook, Twitter, and YouTube under the heading 'Follow us'. It also features sections for 'Get to know us' with links to 'Who are we', 'Where are we', and 'Contact us', and for 'Let us help you' with links to 'FAQ', 'Terms and conditions', and 'Return policies'. At the bottom, a copyright notice reads 'Copyright My Vitality © 2018'.

Supplement page

The screenshot shows a product page for 'Supplier E' on the MyVitality website. The header features the MyVitality logo with a blue cross icon and the text 'MYVITALITY ALTERNATIVE HEALTHCARE'. A navigation bar below includes links for Home, About, Services, Store, and Contact. The main title 'PRODUCT' is centered above the product details. The product information is organized into three columns: 'SUPPLIER E' (image of Supplier E's logo), 'DESCRIPTION' (cost: R219.64, supplement ID: 52, description: 500ml glass, NAPPI code:), and 'BUY NOW' (quantity selector set to 2, subtotal: R439.28, and an orange 'ADD TO CART' button). The footer contains sections for 'Get to know us' (links to Who are we, Where are we, and Contact us), 'Follow us' (social media icons for Facebook, Twitter, and Google+), and 'Let us help you' (links to FAQ, Terms and conditions, and Return policies). Copyright information at the bottom states 'Copyright My Vitality © 2018'.

SUPPLIER E

DESCRIPTION

Cost: **R219.64**
Supplement ID: **52**
Description: 500ml glass
NAPPI code:

BUY NOW

Quantity
(**33**)

Subtotal
R439.28

ADD TO CART

Get to know us

[Who are we](#)
[Where are we](#)
[Contact us](#)

Follow us

Let us help you

[FAQ](#)
[Terms and conditions](#)
[Return policies](#)

Copyright My Vitality © 2018

The screenshot shows a product page on the My Vitality website. At the top, there's a header with the My Vitality logo, a cart icon showing '0 items', and navigation links for Home, About, Services, Store, and Contact. The main title 'PRODUCT' is displayed prominently. Below it, the section 'SUPPLIER E' is shown, featuring a logo with the text 'Supplier E' and stylized green and yellow shapes. The 'DESCRIPTION' section includes the cost 'R219.64', supplement ID '52', and a description '500ml glass'. There's also a field for 'NAPPI code:'. The 'BUY NOW' section allows selecting a quantity of '(33)' with a dropdown menu set to '2', showing a subtotal of '439.28' and a large orange 'ADD TO CART' button. At the bottom, there's a dark sidebar with social media links for Facebook, Twitter, and YouTube, followed by sections for 'Get to know us' (links to Who are we, Where are we, and Contact us) and 'Let us help you' (links to FAQ, Terms and conditions, and Return policies). The copyright notice 'Copyright My Vitality © 2018' is at the very bottom.

The Shopping Cart

The screenshot shows a shopping cart page for 'MYVITALITY ALTERNATIVE HEALTHCARE'. The cart contains three items:

Supplier	Supplement	Price	Quantity	Change	Subtotal
	ID: 52 Description 500ml glass	R219.64	<input type="text" value="2"/>	UPDATE	R439.28
	ID: 157 Description 90	R302.46	<input type="text" value="1"/>	UPDATE	R302.46
	ID: 112 Description NONE	R2761.12	<input type="text" value="1"/>	UPDATE	R2761.12

At the bottom, there are 'CLEAR CART' and 'CHECKOUT' buttons.

Get to know us
Who are we
Where are we
Contact us

Follow us
  

Copyright My Vitality © 2018

Let us help you
FAQ
Terms and conditions
Return policies

Note: Mobile version of cart is still under development.

The Checkout Page

The screenshot shows the checkout process for My Vitality Alternative Healthcare. At the top, there's a header with the My Vitality logo (a green bag with a white cross) and the text "MYVITALITY ALTERNATIVE HEALTHCARE". A shopping cart icon indicates 4 items. Below the header is a navigation bar with links for Home, About, Services, Store, and Contact. The main title "CHECKOUT" is centered above the form fields.

PERSONAL DETAILS

Name	First name
Surname	Surname
Email	email
Home Phone	(###)-(###)-####)
Work Phone	(###)-(###)-####)
Cell Phone	(###)-(###)-####)
Who referred you to us	ADVERT

ADDRESS

Line One	House number and street
Line Two	Town
Line Three	City
Line Four	Province
Postal Code	0000 for outside SA

ORDER SUMMARY

Total R3502.86
(4 Items)
All items are in stock
BUY NOW
payment procedure
description

Get to know us

Who are we
Where are we
Contact us

Follow us

f t g

Copyright My Vitality © 2018

Let us help you

FAQ
Terms and conditions
Return policies

The screenshot shows the checkout process for My Vitality. At the top, there's a header with the My Vitality logo (a blue cross inside a circle), the text 'MYVITALITY ALTERNATIVE HEALTH-CARE', and a 'Cart (4 items)' link. Below the header is a navigation bar with links for Home, About, Services, Store, and Contact.

CHECKOUT

PERSONAL DETAILS

Form fields for personal details:

- Name: First name
- Surname: Surname
- Email: email
- Home Phone: (###)-(###)-####
- Work Phone: (###)-(###)-####
- Cell Phone: (###)-(###)-####
- Who referred you to us: ADVERT

ADDRESS

Form fields for address:

- Line One: House number and street
- Line Two: Town
- Line Three: City
- Line Four: Province
- Postal Code: 0000 for outside SA

ORDER SUMMARY

Total R3502.86
(4 Items)

All items are in stock

BUY NOW

payment procedure description

Follow us

Social media icons for Facebook, Twitter, and YouTube.

Get to know us

[Who are we](#)
[Where are we](#)
[Contact us](#)

Let us help you

[FAQ](#)
[Terms and conditions](#)
[Return policies](#)

Copyright My Vitality © 2018

Dataflow diagrams

Context Diagram

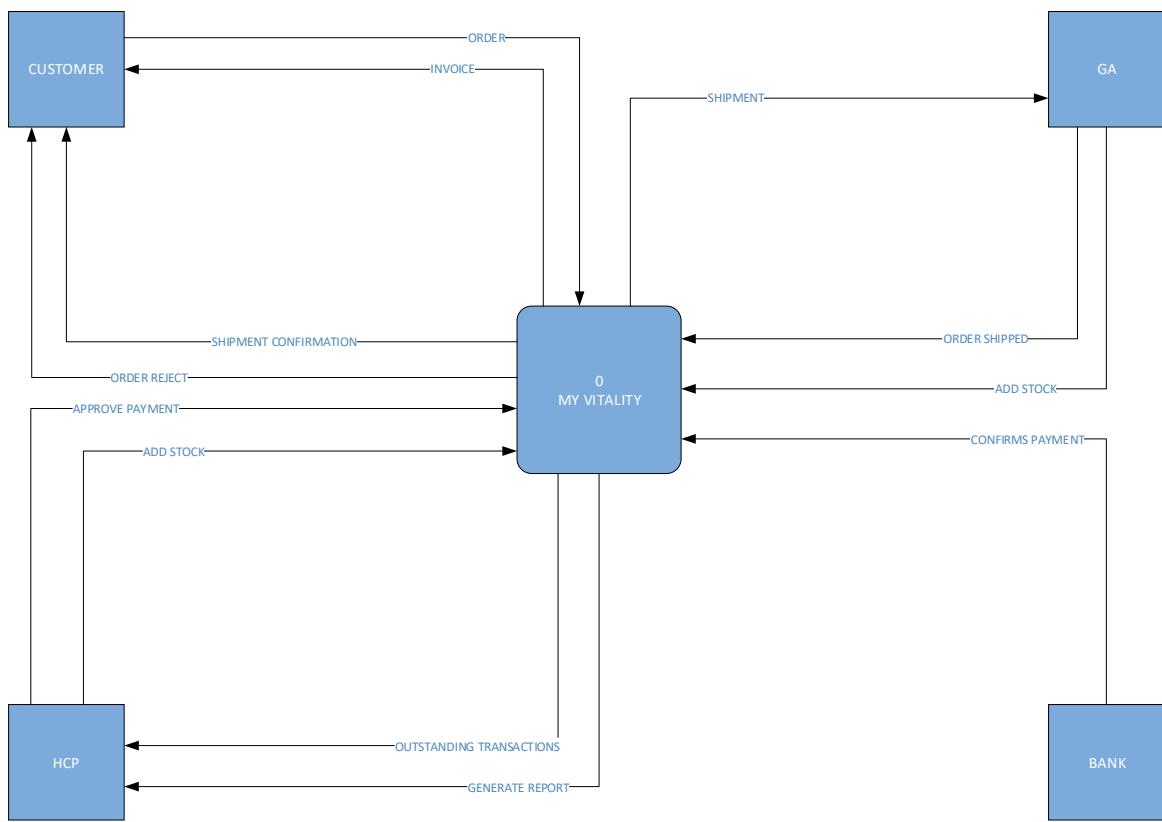


Diagram 0

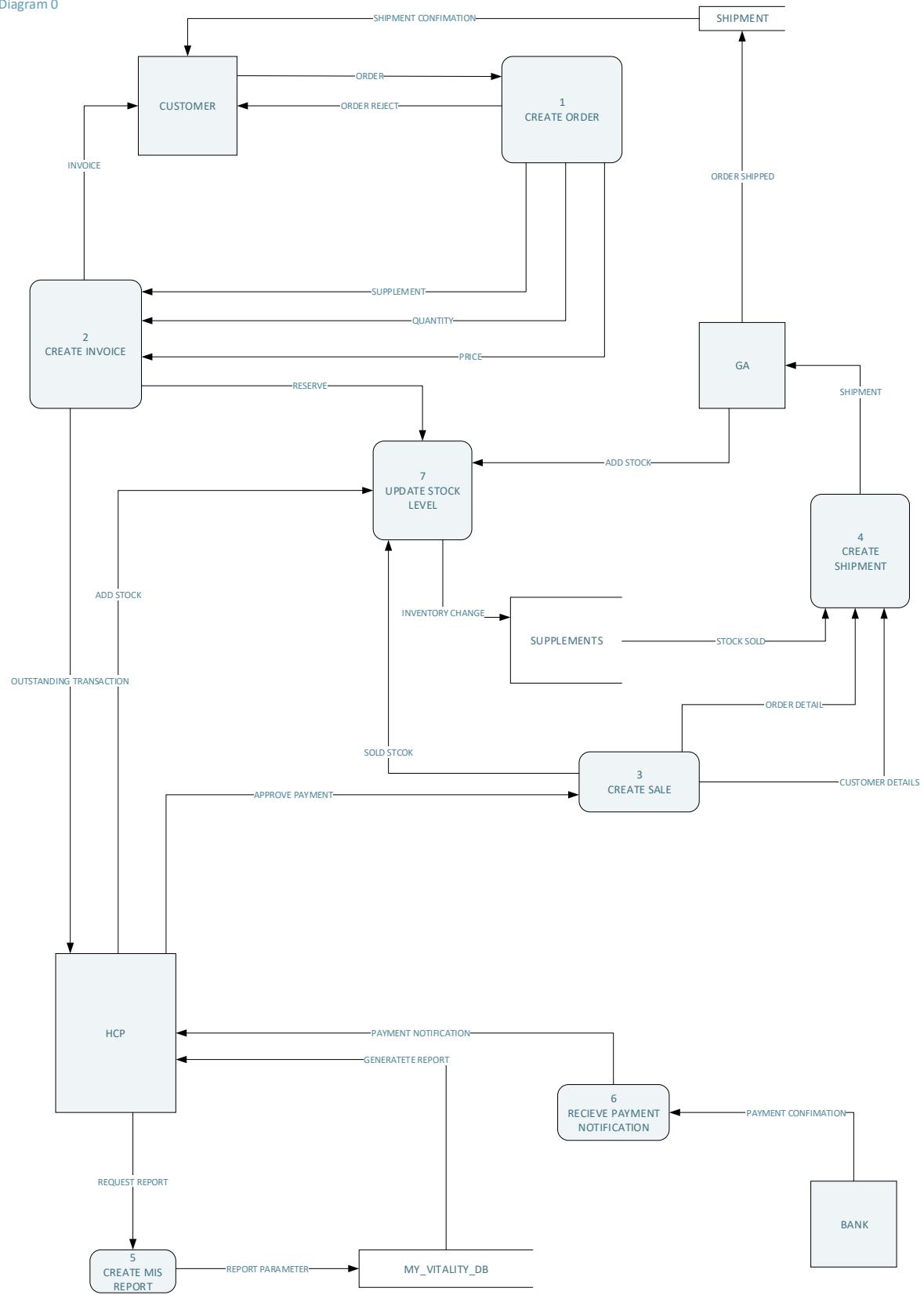
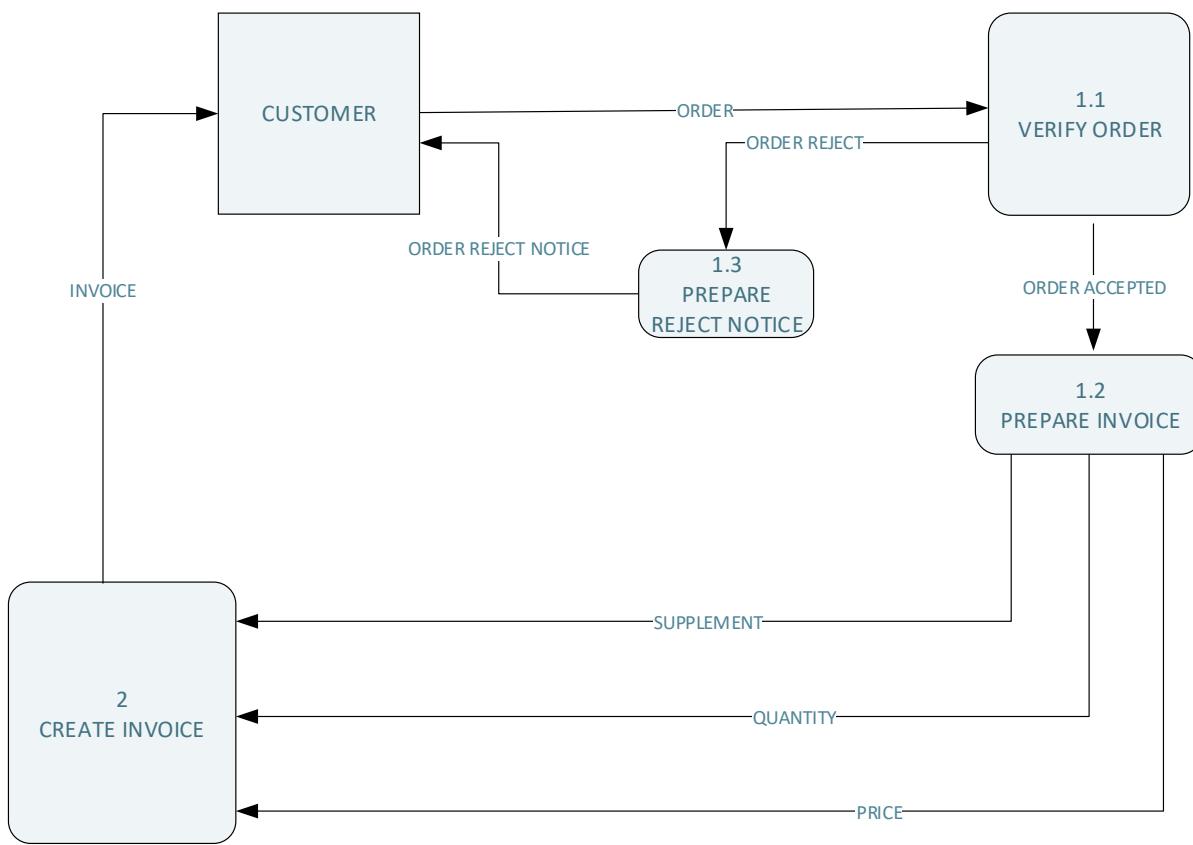
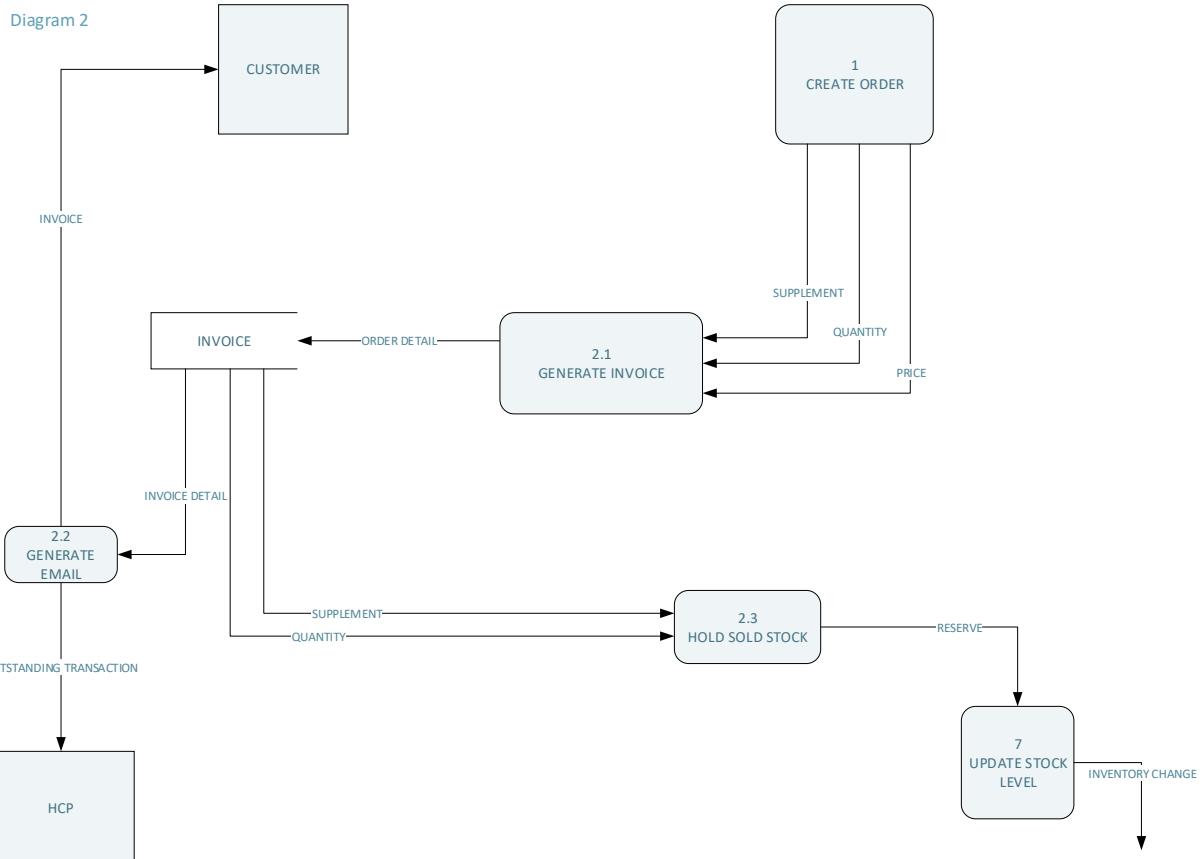


Diagram 1





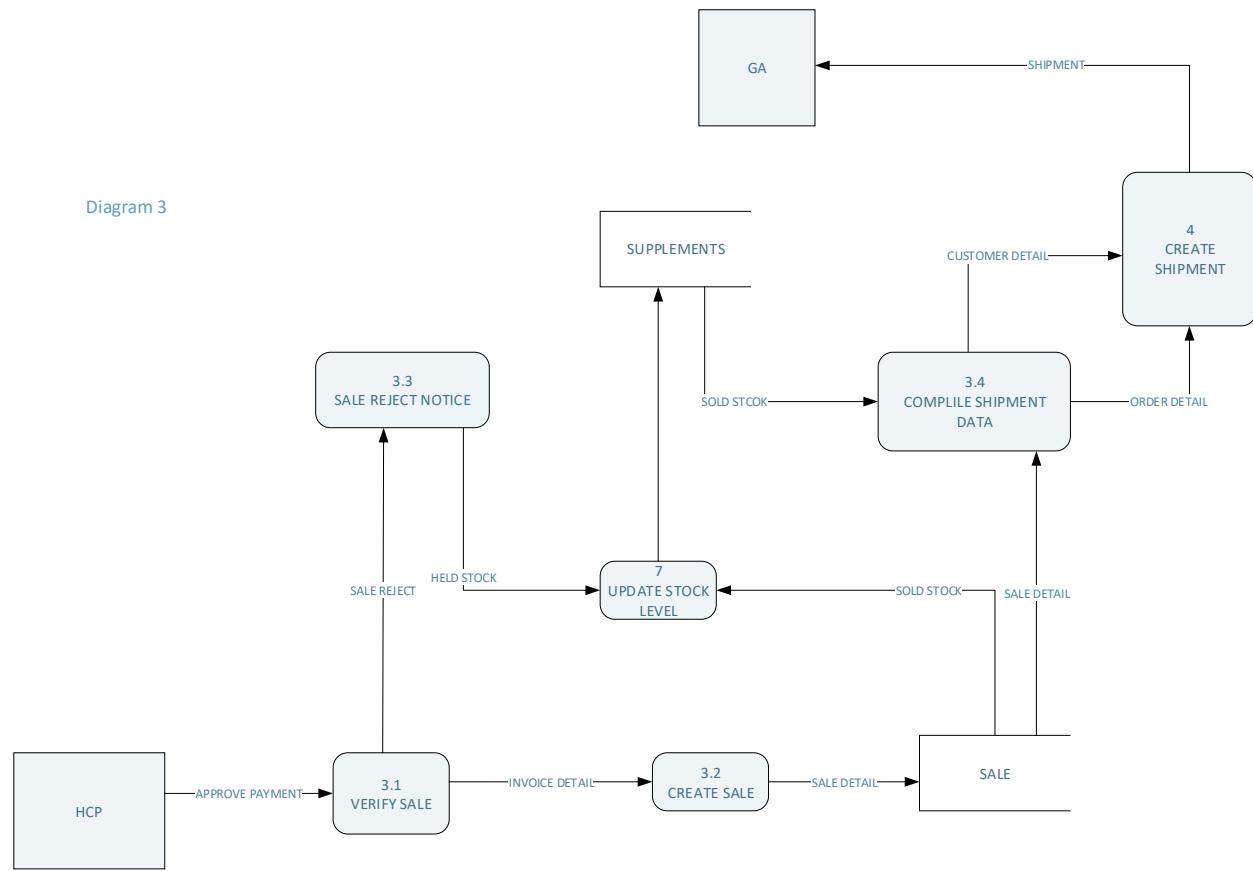
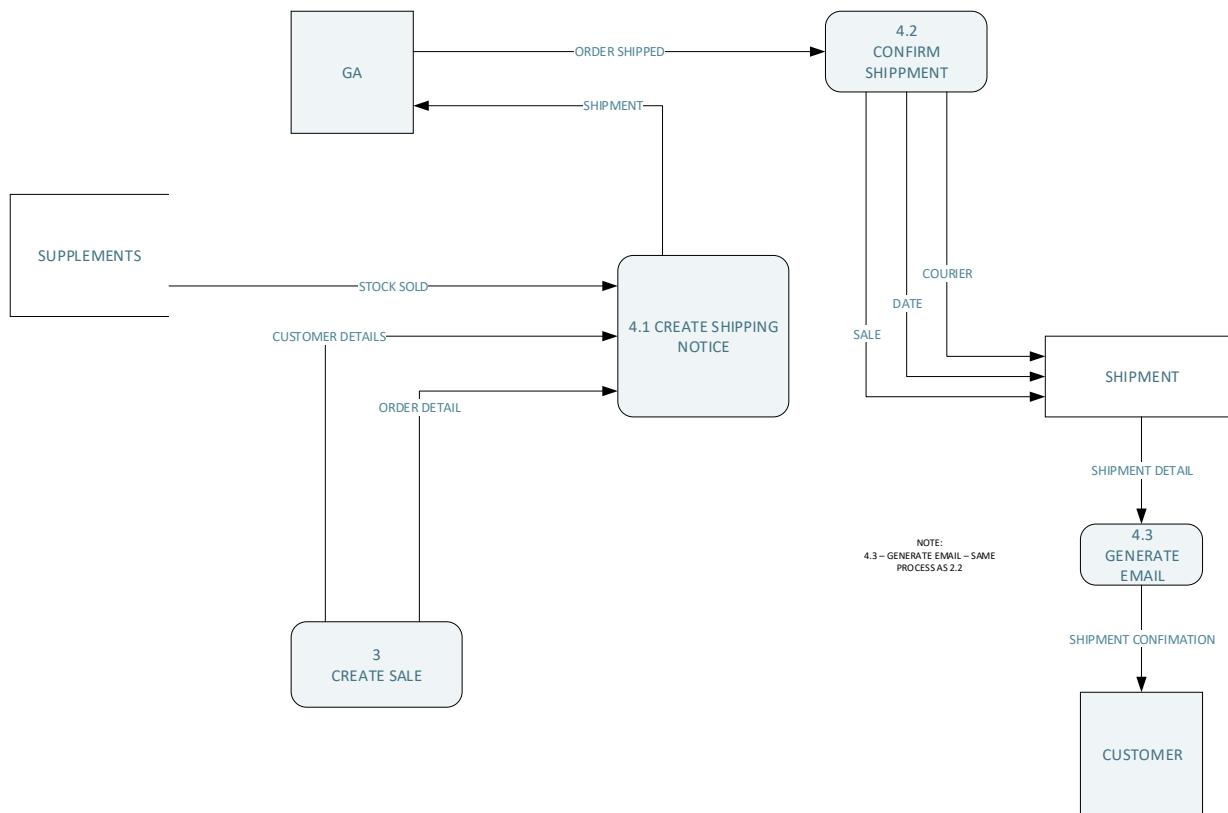


Diagram 4



Processes 5, 6, and 7 do not need to be broken further.

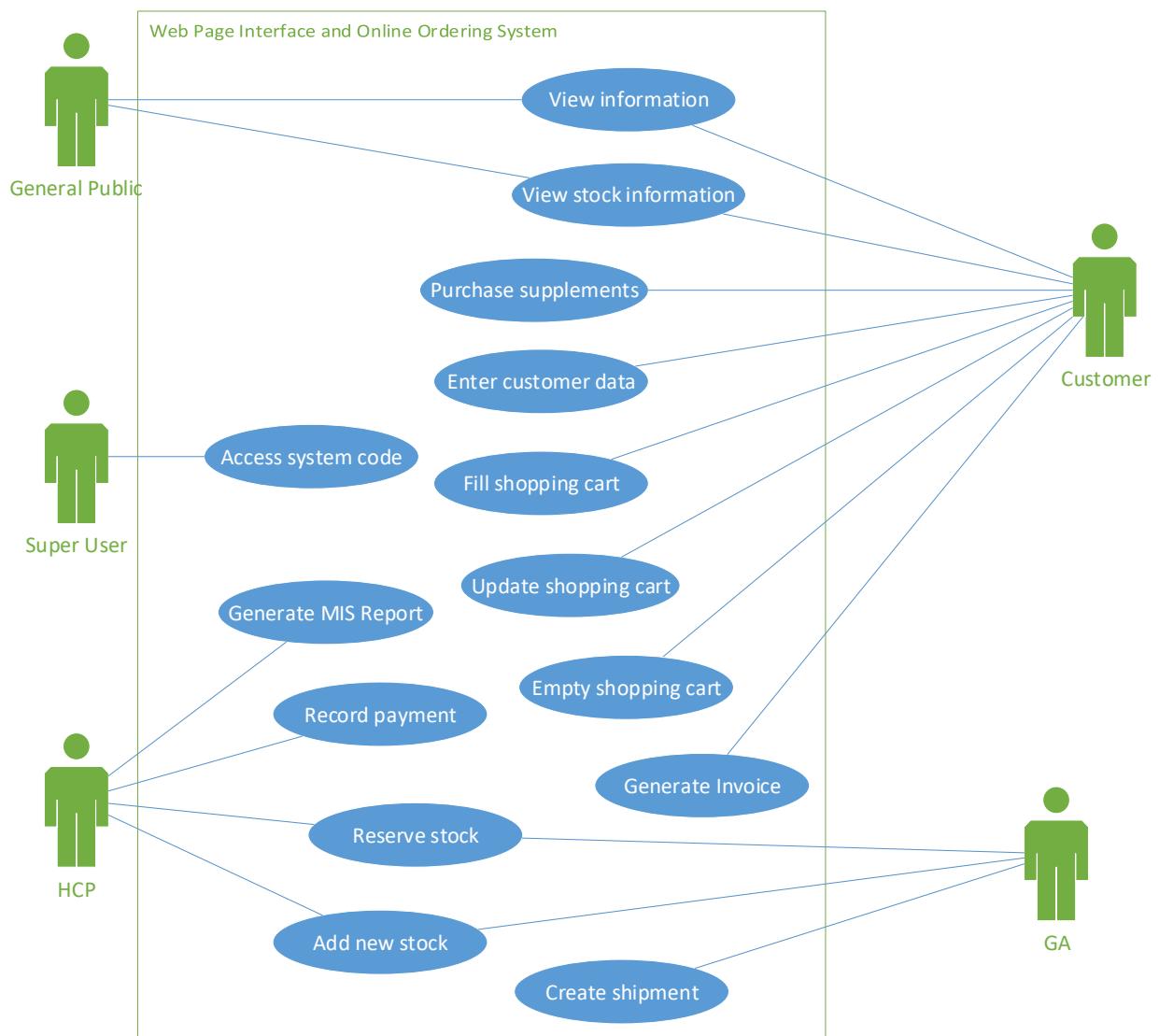
Process symbol 5 represents the process of generating MIS reports. To break this down deeper would only show the queries for each report.

Process symbol 6 represents the process for receiving payment notification. For his system that entails ensure the HCP has a valid email account where the payment company can notify them of payments made.

Process symbol 7 represents the process of updating stock levels. This process is similar to process 6, where the internal details are based around SQL queries.

UML's diagrams

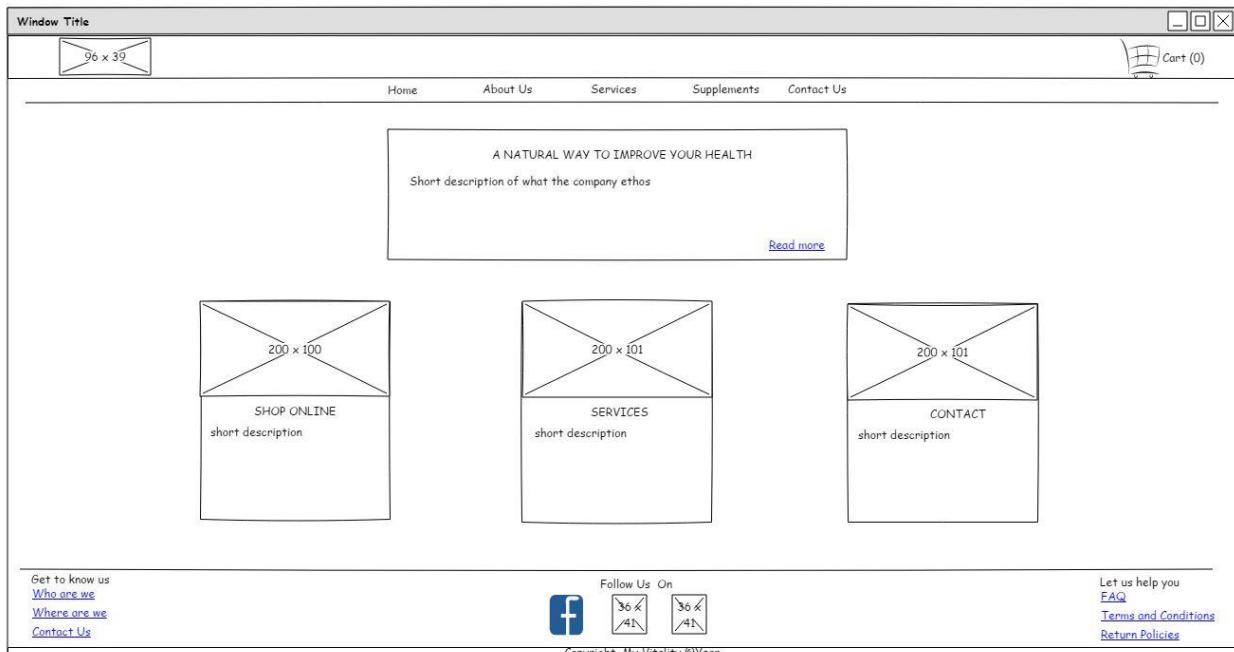
Use Case Diagram



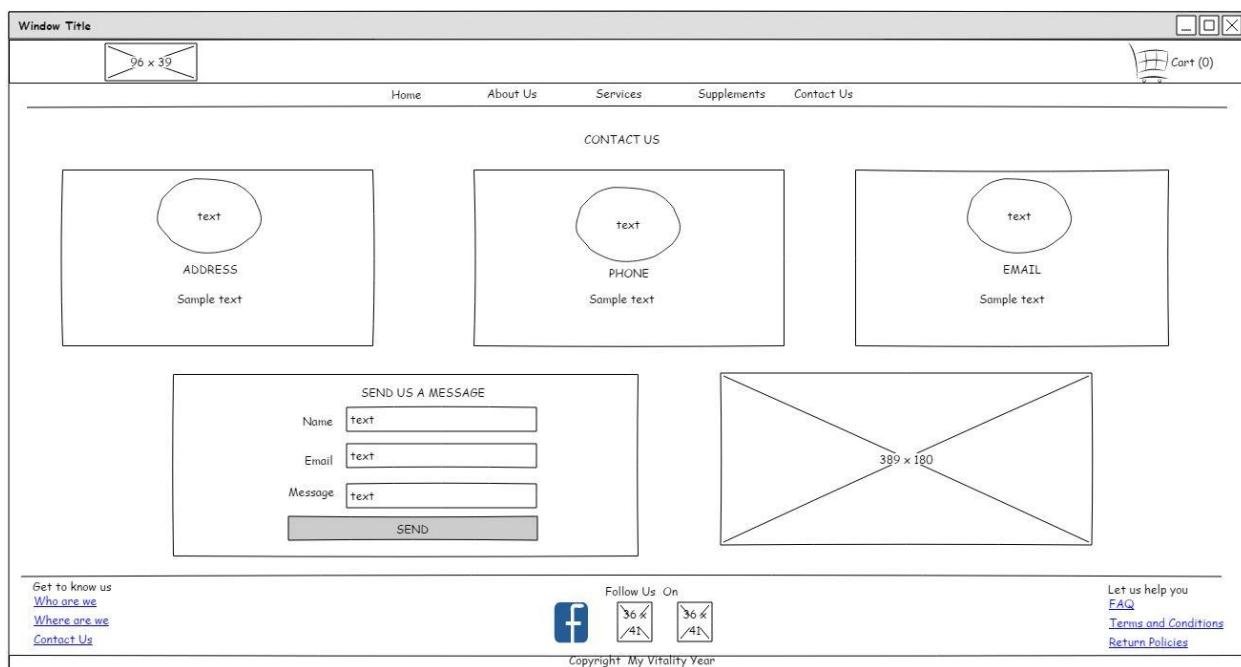
Wireframes

Sample of wireframe diagrams.

Home page



Contact us page



Supplements page

Window Title

Supplier

Sort By: text goes here ▾ Show: text goes here ▾ SEARCH

Description	Price	ID	Quantity
241 x 123			

Get to know us
[Who are we](#)
[Where are we](#)
[Contact Us](#)

Follow Us On

Let us help you
[FAQ](#)
[Terms and Conditions](#)
[Return Policies](#)

Copyright My Vitality Year

Cart page

Window Title

HEADING

Supplier	Supplement	Price	Quantity	Subtotal
	Supplement ID: 100 Description: 100x100 Nappi Code: delete	R199	+ <input type="button" value="2"/> - Available (111)	R398
	Supplement ID: 200 Description: 200x200 Nappi Code: delete	R199	+ <input type="button" value="2"/> - Available (111)	R199

plus minus button adds and removes quantity, then the totals are re calculated. CSS tricks has a article on how to do this. See bookmark -final project - css tricks

Total R796

CLEAR CART CHECKOUT

Get to know us
[Who are we](#)
[Where are we](#)
[Contact Us](#)

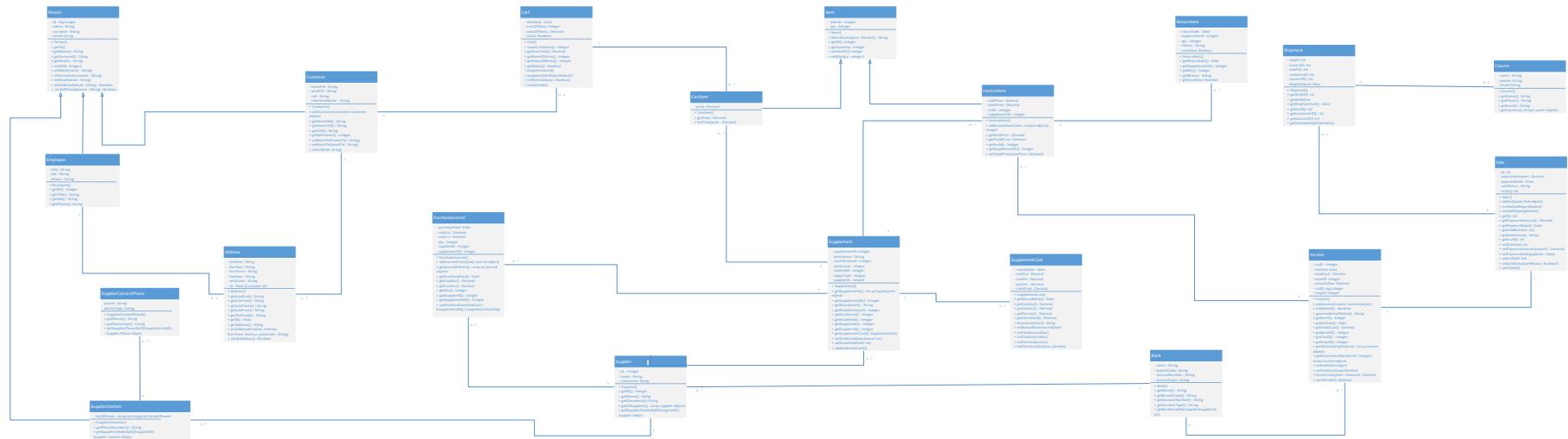
Follow Us On

Let us help you
[FAQ](#)
[Terms and Conditions](#)
[Return Policies](#)

Copyright My Vitality Year

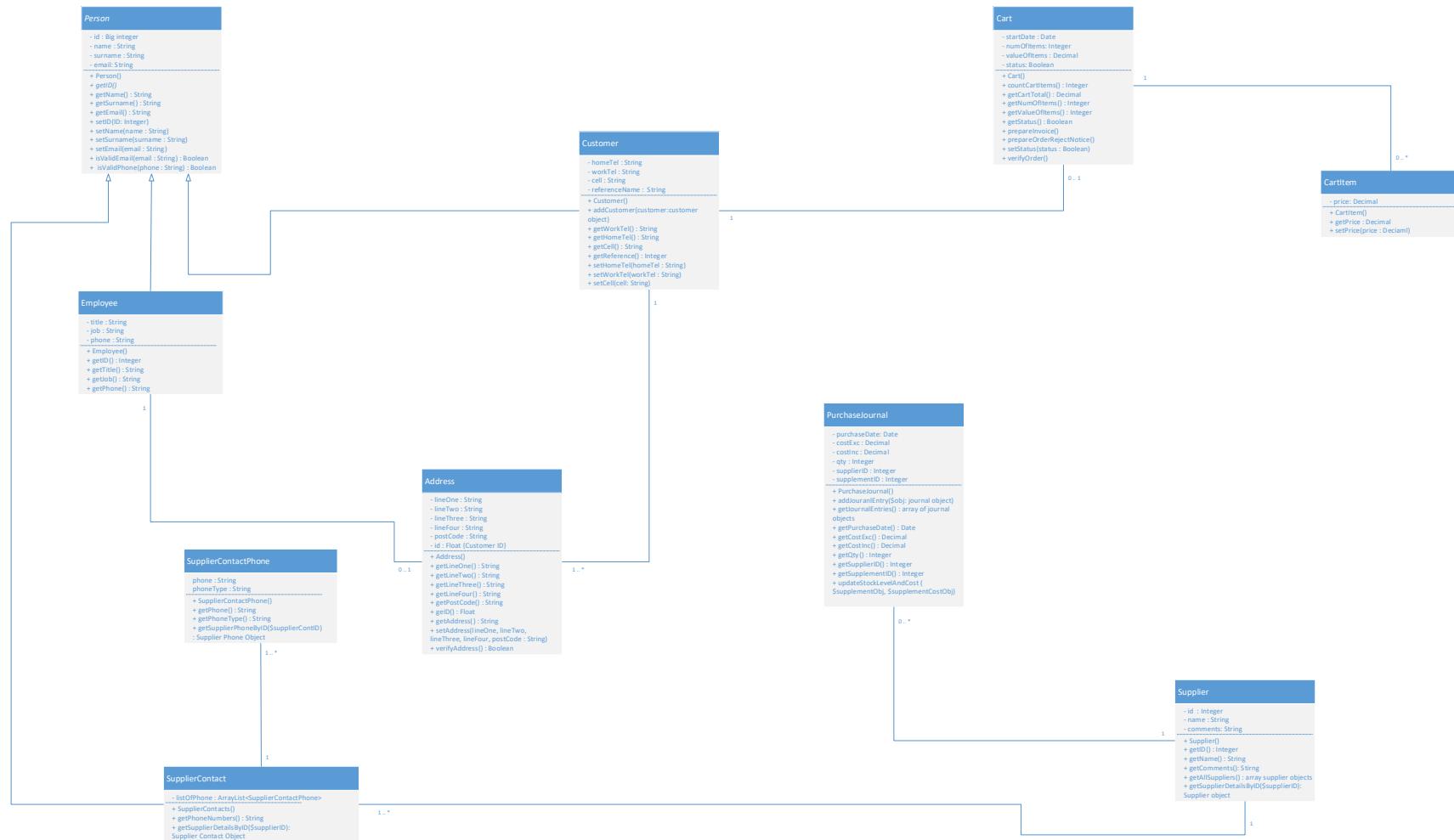
Domain Class Diagram

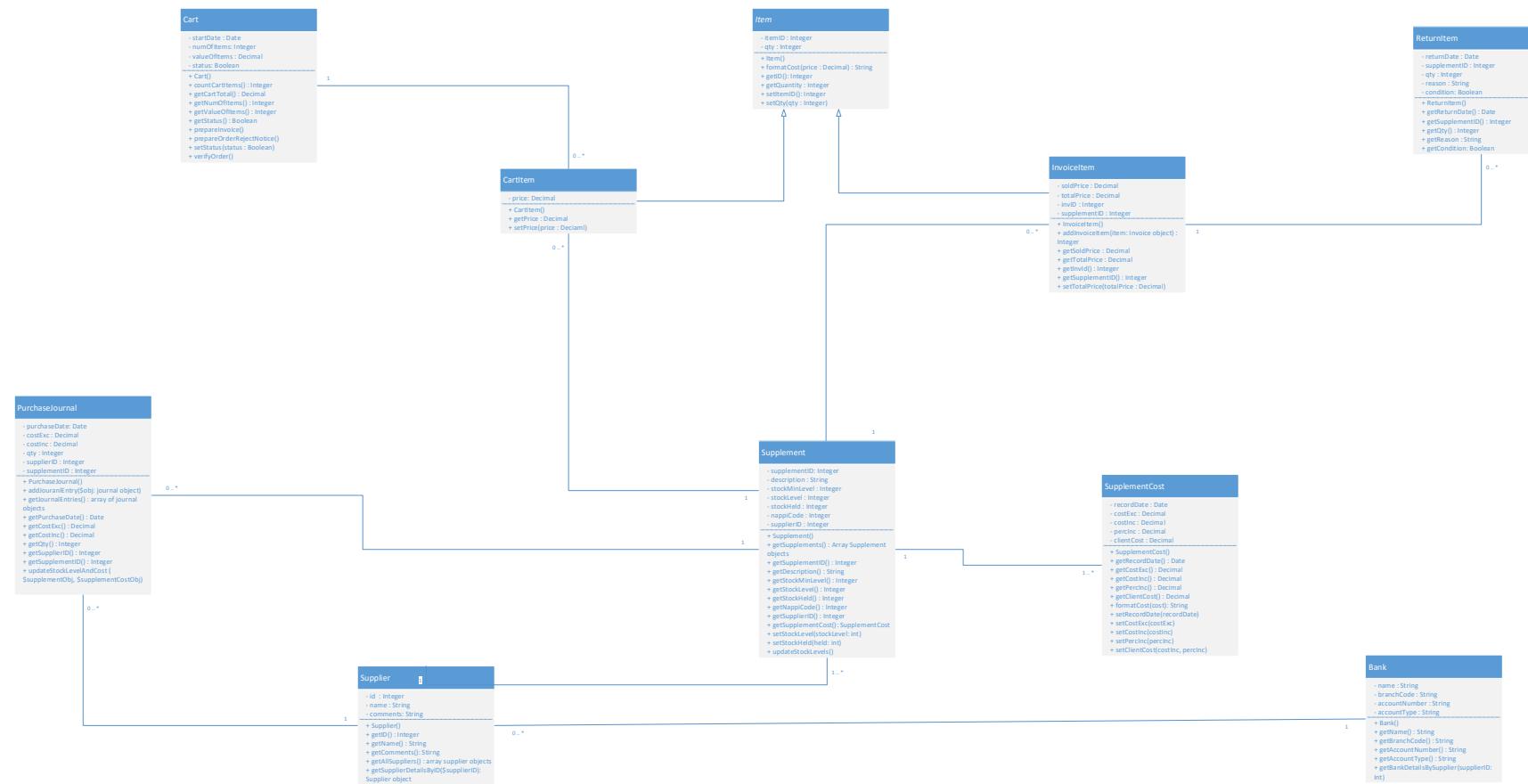
Overview of Domain Class Diagram

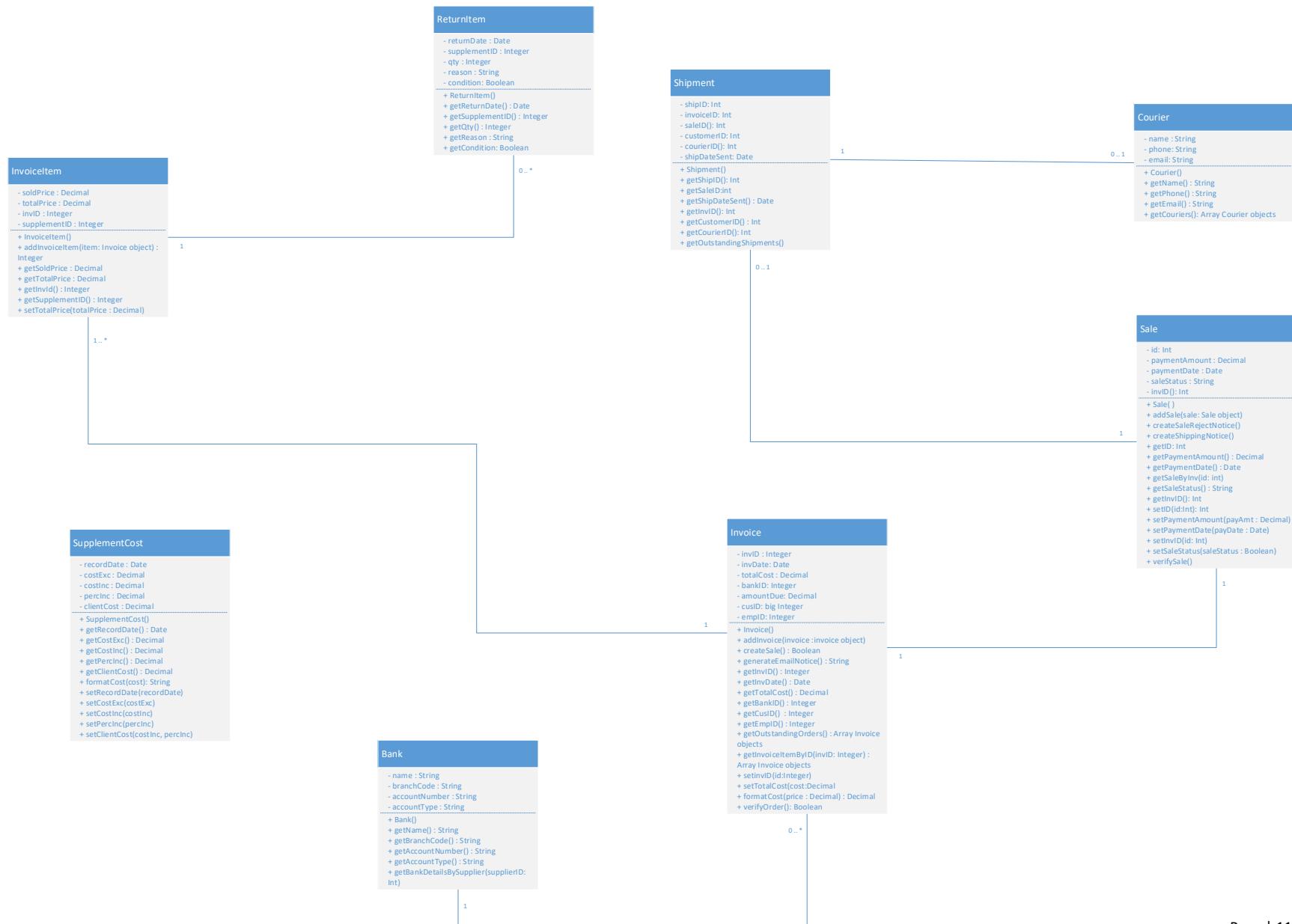


The following are sections from the above diagram, inserted from left to right.

Due to copy and paste issues between Microsoft Visio and Word, connecting lines to tables that are not in view are cut off. Thus some tables appear in multiple shots for continuity.







Implementation Methodology

Implementation and Requirements document

Requirements

Hardware:

- Company Computer
- Network attached Storage (Synology 2 bay NAS DiskStation DS218j - Diskless)
- Router: Asus RT-AC88U Wireless AC3100

Software:

- Operating system: 32-bit Windows 7 based PC
- Windows Defender (comes with Windows 7 OS)
- Licensed version of Microsoft Office 2010
- XAMPP: local web server for testing and deployment purposes
- ATOM: text editor - used for HTML, CSS, JavaScript and PHP
- Notepad++: text editor - Used for SQL queries
- Mozilla Firefox and Google Chrome: Web browsers - used for testing and deployment purposes
- Pencil: An open-source GUI prototyping tool
- Microsoft Visio: Diagramming software

Software Interfaces:

- 000webhost (<https://www.000webhost.com/>)
- phpMyAdmin - Database management tool using Maria DB
 - o The Interface will need a username and a password to access to phpMyAdmin

Communication Interfaces:

- The application will interface with the router to enable a connection to the Internet

Technical performance

Reliability is the probability that the system will be able to process all work correctly and completely without being aborted. Reliability is evaluated as follows:

What damage can result in failure of this system?

- Complete loss of ability for customers to purchase online
- Partial loss of MIS reporting
- Partial loss in processing orders for shipment
- Partial loss of stock reporting functions. Such as low stock level / re-order functionality
- Loss of revenue

Required operational requirements:

- Administration subsystem shall operate in both online and offline modes
- Customer facing system shall operate fully in online mode
- Hosting provider shall guarantee at least 99% uptime
- System shall accurately record all transactions
- System shall maintain historical accuracy of transactions
- Shall use a reliable DBMS

Required functional requirements:

- Shall collect and store information about products
- Shall collect and store information contained in invoices
- Shall be able to provide the capabilities to generate MIS reports
- Shall connect to an online relational database
- Shall support different levels of user access
- Shall support a graphical user interface
- Shall allow online customers to view stock levels
- Shall allow online customers to purchase stock

- Shall allow automation of email messages

In serves support

Classes and Categories of user.

Classes of User:

- Superuser. This user has root access to the system, access to system code and hosting site
- Employee. This user has access to the administration subsystem
- Customer. This user has access to the online site and ordering subsystem.

Categories of user:

The system will recognize two types of employee:

- HCP. Functions available to user:
 - o Generate and view MIS reports
 - o Record payment
 - o Reserve Stock
 - o Add new stock
- GA. Functions available to user:
 - o Reserve stock
 - o Add new stock
 - o Create shipment

The system will recognize two types of customer:

- General Public. Functions available to user:
 - o View general information
 - o View stock information
- Purchaser. Functions available to user:
 - o View general information
 - o View stock information

- Purchase products
- Fill, update, empty a shopping cart
- Generate an invoice

Training for users

The superuser will not be provided with any specific training. Any new superuser will have access to all system documentation and may request the installation and user manuals.

The employee will receive group and individual training, as well as access to the user manual.

The GA and HCP will receive a demonstration of how the customer facing application works and will be able to test out the application and ask any questions they have about its operations.

The HCP and GA will then receive individual training specific to the functions of the system that are applicable to their roles in the company.

The customer will have a link on the site where they can get answers to commonly asked questions about the operation of the system. Since the customer facing application carries a common theme among web sites and online shopping portals, this should be enough for most users. The contact us page allows for users to contact the company for any additional detail.

Security

Consequences of security and data breach

Erasure or contamination of application data.

The consequences of erasure or contamination of application data:

- Incorrect product prices / data displayed to customer
- Stock level inaccuracy
- Transaction history compromised

Disclosure of privileged information about individuals.

The consequences of the disclosure of privileged information about individuals:

- Customer details would be compromised

- Supplier and my vitality banking details would be compromised
- Supplier contact information would be compromised

Type of security required

- Ability to restrict access to the company work computer. Members of the public should not have access to the work computer.
- Ability to restrict access by type of user.
- Ability to restrict access by data type. Employees are grouped into two roles, HCP and GA. These two roles require access to different data attributes. For example, the HCP has access to the MIS reporting system, where as the GA does not.
- Ability to control access based on system function. The superuser will be the only user who is granted access to system code, root access to the Database Management System, and hosting site.

Backup procedure

The ability to restore function in the event of a failure.

- A copy of system code shall be stored on and offsite. Offsite locations:
 - o Cloud
 - o Superuser
- System data shall be stored on and offsite: Offsite location:
 - o Cloud

Onsite backup procedure

Procedure for Synology DiskStation NAS device and Windows operating system

Before the first backup commences the following must be completed:

- Hardware and software installation for DiskStation
- Create volumes and shared folders
- Create permissions for users who are granted permission to access shared folders

After the previous steps were completed, follow the following instructions:

1. Launch Backup and Restore and configure automatic backup from Windows 7 to your DiskStation
2. Select a location to save your backup to. Save on a network to locate your DiskStation on the local area network
3. In the window that appears, enter \\Synology_Server_Name\Shared_Folder_Name or \\Synology_Server_IP\Shared_Folder_Na me, and then authorize using the credentials of a DSM user who has the "Read/Write" privilege to the specified shared folder.
4. Click OK and follow the on-screen instructions to complete the setup

After the following procedure has been completed the system will automatically be backed up at the stated intervals.

Testing and evaluation of system

The system will be tested according to a test plan. The plan will specify how, when the testing will be performed, who will conduct it, and what data will be used. The plan will consist of four phases.

Phase One

How: The first phase involves the programmer compiling the program during coding to remove any syntax errors and to desk check the program code to spot and remove logic errors.

When: As the programmer writes code, they will check for syntax errors as an ongoing process.

Who: Programmer / Superuser

Phase Two

How: Unit testing. The second phase involves testing an individual unit or module of code before it is integrated into the software. This will include testing that methods produce the correct results and that classes model the data accurately and function as stand-alone units.

When: As each module or unit of code is completed.

Who: Programmer / Superuser

Test Data Used: Each class will be given data specific to its operation to test that it functions correctly.

- Each class will be tested that it can be instantiated
- That properties can only be retrieved via their get methods

- That properties can only be set via their set methods
- That each function is given the correct values to return / or accomplish the task it is given
- That it handles any errors that occur during operation

Phase Three

How: Integration testing. The third phase involves testing that the programs created can work together. This includes testing that the individual units tested in phase two can work in unison with each other.

When: As units that work in unison are both completed. For example, when the supplement class is complete, and the supplement database class are both complete, they will be tested to see that they work together.

Who: Programmer / Superuser

Test Data Used: Each class will be given data specific to its operation to test that it functions correctly when in use with other classes.

Phase Four

How: The fourth phase involves testing the entire system. This is to ensure that the system has delivered the required level of functionality and has satisfied the agreed upon business needs. Once system testing has been completed in-house, the system will be demonstrated to the clients for the final acceptance tests.

When: When the system is complete and all three of the previous phases have successfully completed testing.

Who: Programmer / Superuser, independent third party, client (HCP), and GA.

Test Data Used:

Each user will be given a separate task list to complete. The task list will be geared towards the specific role the user has with the system. Except for the superuser who will test the complete system.

Independent third party in the role of a customer.

- 1) Navigate to the home page of the online site
- 2) Find the following information:
 - What the company does and who works for them
 - What in house services do they offer

- How do you contact them?
 - What do they sell?
- 3) Add two different supplements to your cart
 - 4) Clear the cart
 - 5) Add a variety of products in varying quantities to the cart. Try to add quantities that are not available
 - 6) Update the quantities in the cart
 - 7) Purchase the items in the cart
 - 8) View your invoice and contact the company with a question about it
 - 9) Browse the site and try to enter information that would not normally be entered in forms and see the results

GA

- 1) Log into administration subsystem
- 2) Browse the site to see what you have access to
- 3) View shipments that are ready to be processed
- 4) Update shipments that are processed
- 5) View stock levels
- 6) Add new stock to the system

HCP

- 1) Log into administration subsystem
- 2) Browse the site to see what you have access to
- 3) View the MIS reporting system
- 4) View stock levels

- 5) Click supplement ID to view data for re-ordering that item
- 6) Add new stock to the system
- 7) Check the outstanding orders and approve or reject the orders depending on payment approval
- 8) Click on the Invoice ID to view the invoice items for that invoice

Superuser

Complete all three of the to-lists. The third party, GA, and HCP list listed in the above section.

Implementation

Operational Concepts

The hardware and software requirements will be initially managed by the superuser. Software will be:

- installed to work with the client's hardware
- configured to interface with any external software interfaces
- where possible, automated to remove the need for the client to interact with it (i.e. backup and security)

Technical Performance

Operational and functional requirements evaluated through research, testing and performance analysis.

Research findings:

- Hosting provider used instead of in-house hosting. Reasons: Cost, security, lack of IT skills in company
- Open source or free to use software used where possible. Reasons: Cost and not being tied into a single vendor

The system will be tested to ensure that all functional and operational requirements are fulfilled when the system becomes operational. Once the system goes live, it must be periodically evaluated for performance. 000webhost and myPhpAdmin both come with statistical analysis tools. From here it is possible to view data on how well the site is performing.

In-Serves Support

The training for users will be performed depending on the class or category that user falls into. Each user performs different tasks with the system and thus needed training specific to their goals.

Training will be provided in a variety of forms:

- In-person
 - o Group and individual training
- Manuals. A user manual and installation guide will be provided
- Section on the Web site where the customer can find out procedures for purchasing products and other frequently asked questions

Security

The data stored in the system is vital to its operation. The following controls were implemented to protect it.

To prevent invalid data from entering the system, the following input controls were used:

- Value limit controls – check numeric data inputs to ensure that the amount entered is reasonable
i.e. When a user attempts to purchase a product, they should be limited to selecting a quantity that is less than or equal to the amount for sale. They should also not be able to purchase a negative number of products.
- Completeness controls – check that all required values are present
i.e. When a user checks out and completes the transaction, the user must enter some personal information and an address before the transaction can be processed. To ensure that the details are present and valid the following completeness controls were implemented:

Browser based controls:

- o HTML5 require attribute to ensure that the user has entered a value in the required controls
- o HTML5 email input control was used to ensure that a user entered a valid email format

Server-side controls:

- o Ensure that the required controls are not empty

- Escape input (remove leading and trailing whitespace, remove slashes and escape input that could contain malicious content)
- Use regular expressions to ensure that the input entered is valid. For example:
 - Names only contain upper and lowercase letters, spaces, and a single quote (')
 - Phone numbers are entered in the desired format - (###)-(####)-(####)
 - Emails are entered in the correct format
 - Postal codes only contain 4 digits
- Field combination controls – review combinations of data inputs to ensure that the correct data is entered

i.e. When a user adds new stock into the system, the date should not be greater than the current date.

To restrict who can add, modify or view the data stored in the database, the following access controls were implemented.

- Only the super user can log into the database management system (using a username and password). The administration staff can only access the database through the user interface of the administration application.
- Each staff member has been assigned to specific roles and groups within the database management system, which restricts further restricts the data they can view and access.

These two controls will limit who can access the data and what can be accessed once inside the system.

To keep a track of transactions that occurred in the system the following was implemented:

The database has been designed to record the historical accuracy of transactions.

- Supplement costs have their own table to record the prices of the supplements over time
- A purchase journal table has been included to record when a stock was added to the system, in what quantity and at what price.
- Invoices and invoice item entities were created to record what a user ordered, and a separate entity was created to record the sale data associated with that invoice. i.e. When and what a customer paid, and if the sale was approved, rejected or cancelled during processing.

To prevent the same data being updated multiple times or when recording a single transaction requires multiple related database updates the following update control was implemented.

- Once an invoice item is added to the invoice, the stock levels are updated so that another user cannot order the same stock that is involved in the current order. The same is done in reverse when an invoice is rejected for non-payment.

To limit access to the system, the following output control was used:

- A valid username and password are required to access the administration system so that only valid employees can access, modify and view the data in the system

- The administration system is not accessed via the customer facing site

Each employee has been granted specific roles so that they are limited in what they can access and modify as per the Alt health document, in which each employee's job duties were detailed.

Each transaction is logged to keep a record of transactions.

Ensure that only valid data can be entered into the database.

Each staff computer running Windows is set to run updates automatically and that Windows defender is active and set to run periodically. Windows defender firewall is set to be active.

Backup

The ability to restore function to the system in the event of a failure is a vital aspect of the system. It is not recommended to keep all backups in a single location or to leave the backup procedure to be carried out manually.

- Backups to be stored in a variety of devices, not in a single location or by a single person
- Backup procedure to be automated to remove the possibility of human error

Testing and Evaluation

The system will be tested according to a test plan. The plan will specify how and when the testing will be performed, who will conduct it, and what data will be used. The plan will consist of four phases.

The first phase involves the programmer removing syntax errors and checking for logic errors. The second phase, called unit testing, involves testing an individual unit or module of code before it is integrated into the software.

The third phase, called integration testing, involves testing that the individual units and modules can work together as a unit.

The fourth phase, called system testing, involves testing the entire system.

To ensure that the system works correctly for each type of user, each user type will be provided with its own test plan to test the functionality of specific parts of the system, and the superuser will test the entire system.

References

- Canva.com. (2018). [online] Available at: <https://www.canva.com/> [Accessed 18 Aug. 2018].
- Inc., S. (2018). *DiskStation Manager - Knowledge Base / Synology Inc.*. [online] Synology.com. Available at: https://www.synology.com/en-global/knowledgebase/DSM/tutorial/Backup_Restore/How_to_back_up_data_from_Windows_computer_to_Synology_NAS_using_Windows_or_3rd_party_applications [Accessed 17 Aug. 2018].
- Tilley, S. and Rosenblatt, H. (2016). *Systems analysis and design*. 11th ed. Cengage Learning.
- Unc.edu. (2018). [online] Available at: <https://www.unc.edu/~stotts/comp523/USDA-funcSpecs.pdf> [Accessed 17 Aug. 2018].

Assignment 7

User manual



MY VITALITY

Alternative Healthcare Solutions

System User Manual

MY VITALITY: ALTERNATIVE HEALTHCARE SOLUTIONS

System User Manual: Customer

© My Vitality
01 Diagonal Street • Johannesburg
2001, South Africa
Phone 0112345678 • Fax 0112345679
Email support@myvitality.com

Table of Content

How do I navigate the site?	2
How do I search for your products?	4
How do I see more information about a product?	6
How do I add an item to my shopping cart?	8
How do I alter my shopping cart?	9
How do I purchase items in my shopping cart?	10
How do I make a payment?.....	12

How do I navigate the site?

To navigate My Vitality, there are several options:

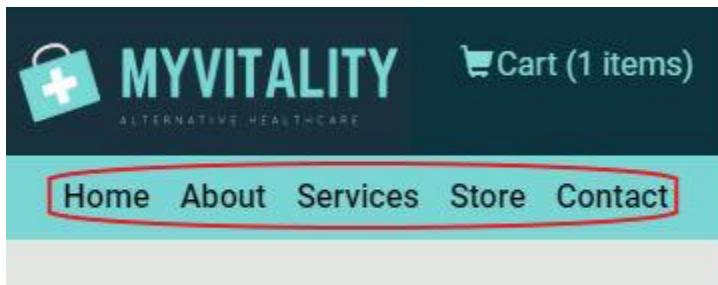


Figure 1

- 1) Navigate to the menu at the top of the page and click one of the menu options.

Hover over one of the menu options circled in red (see Figure 1). Once the option you require changes from black to orange, you may click it, to take you to your desired location.



Figure 2

- 2) Navigate to the cart icon in the top right-hand corner of the page, to quickly take you to your shopping cart.
- Hover over the cart icon circled in red (see Figure 2). Once the icon changes to orange, you can click it to take you to your cart.

- 3) A variety of pages throughout the site contain icons that may be clicked to take you to each selected section. If the icon changes color when you hover over it, it will take you to that section.

Hover over an icon circled in red (see Figure 3), to see if it changes from light blue to orange. If it does, you may click the icon to be taken to that part of the site. If it does not change color, you are at the end of the navigation tree and cannot go any deeper into that section.



Figure 3

4) Navigate to the bottom of the web page and you will find the sections titled, ‘Get to know us’ and ‘Let us help you’. You may click on the links in these sections to take you to different sections of the site.

Hover over any of the light blue icon options circled in red (see Figure 4) and when the text changes to



Figure 4

orange it may be clicked.

5) Navigate to the bottom of the web page and you will find the section titled ‘Follow us’. From here you can click on one of the icons and it will take you from our site to one of our social media platforms.

Hover over any of the icons in this section, circled in red (see Figure 5) and when the center of the icon changes from white to orange, it can be clicked to take you to the social media site.

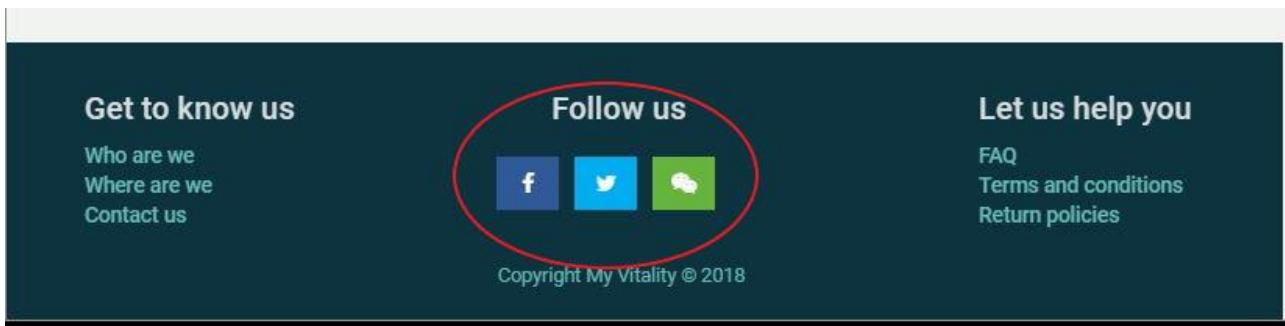


Figure 5

How do I search for your products?

To search our products, you must navigate to our online store. To get there, click on the Store menu option at the top of the page. Alternatively, if you already know the supplier of the product you desire, you may choose one of the drop-down menu options, to see only their products.

Hover over any of the items circled in red (see Figure 6) and when they change to orange, you may click it to take you to that part of our online store.

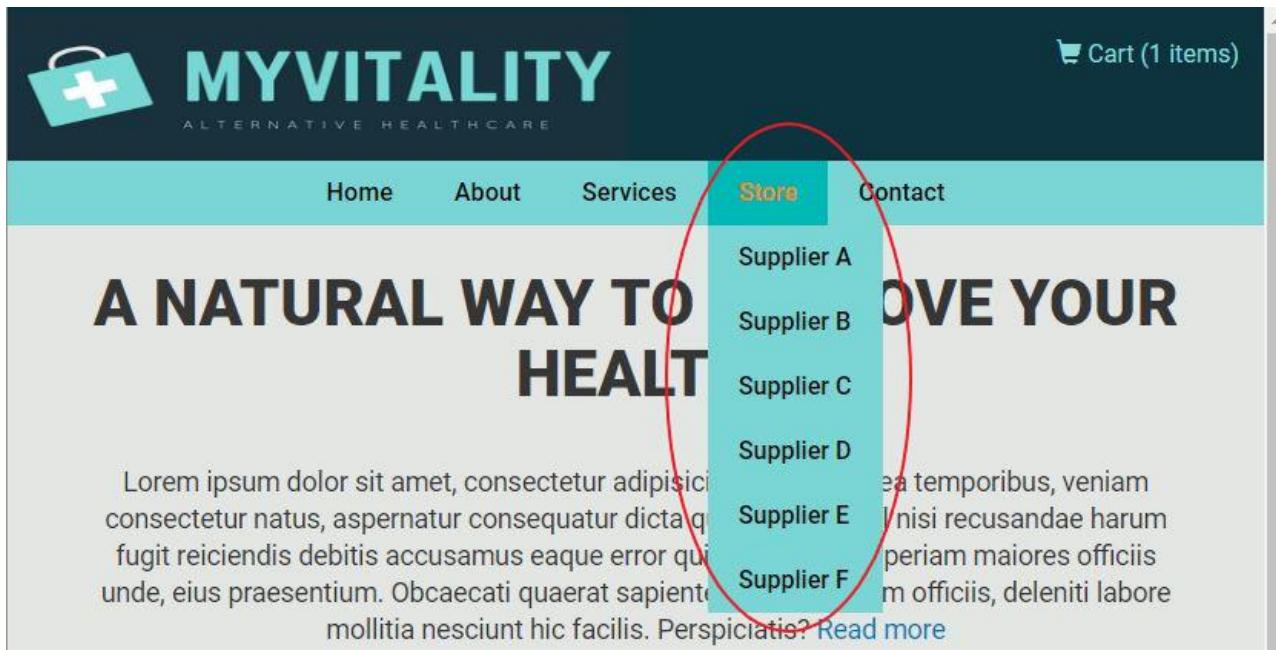


Figure 6

When you have arrived in the online store, you may use our search functions to show the products in a different order, or you may limit the amount of results in your search. The search functions are in the upper left corner of the page (see Figure 7 below).

To sort the products by different parameters, click on the drop-down menu shown in green, choose your parameter and the click on the Search button circled in red (see Figure 8 below).

To limit the number of products, click on the drop-down menu shown in purple, choose your parameter and the click on the Search button circled in red (see Figure 8 below).

You may select both drop-down menus before you press the search button.

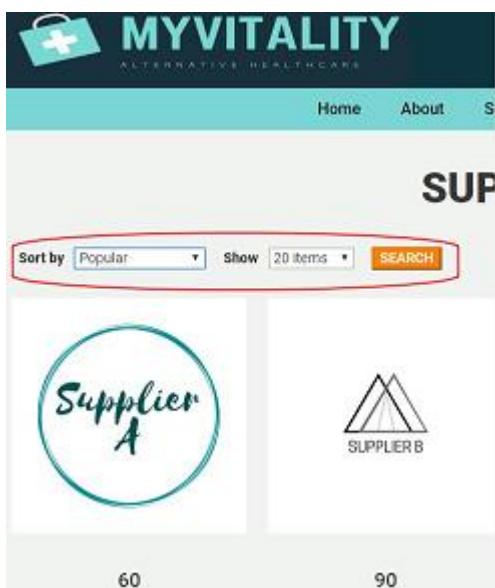


Figure 7: Search function location

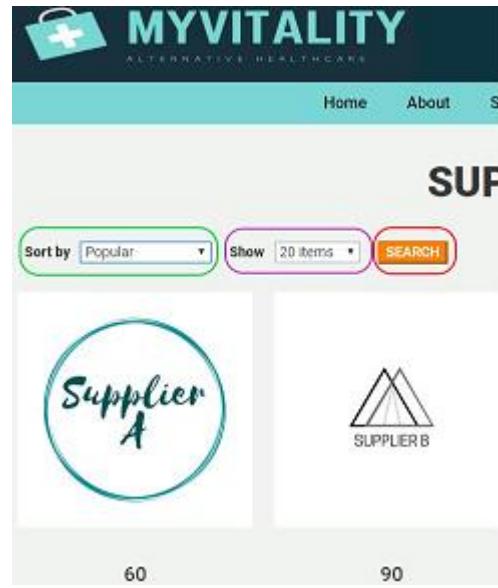


Figure 8: Search function capabilities

How do I see more information about a product?

The online store home pages provide you with basic information about a product. The basic information about each product that is displayed:

- Product image (marked in red, see Figure 9)
- Product description (marked in green, see Figure 9)
- Product ID (marked in yellow, see Figure 9)
- Quantity available to purchase (marked in blue, see Figure 9)



Figure 9: Basic information about a product

To see the specific product page, you click on the product image. From here you can see more information about the product, choose how many items you would like to purchase and add an item to the shopping cart. The page is split into 3 sections. The two sections marked in purple (see Figure 10) describe the product and the section marked in green (see Figure 10) is where you will add the item to your shopping cart.

PRODUCT

SUPPLIER E

DESCRIPTION

Cost: **R219.64**

Supplement ID: **52**

Description: 500ml glass

NAPPI code:

BUY NOW

Quantity
(31)

Subtotal
R219.64

ADD TO CART

Figure 10: Product page

How do I add an item to my shopping cart?

To add an item to your shopping cart, you must first navigate to the specific page for that product (see the section titled 'How do I see more information about a product?').

Once you have arrived at this page you can:

- view the quantity available to purchase (marked in blue, see Figure 11)
- select the quantity you would like to purchase (marked in red, see Figure 11). To change the quantity:
 - enter a number in the text box or
 - click on the up or down arrows in the text box
- review the subtotal (marked in green, see Figure 11)
- add the items to your cart by pressing the ADD TO CART button (marked in purple, see Figure 11)



Figure 11: Add to cart

How do I alter my shopping cart?

First navigate to the shopping cart page. The shopping cart is accessed by clicking on the shopping cart icon. The icon is in the top-right corner of the page. For further instruction see the section titled 'How do I navigate the site', subsection 2.

Once you have navigated to the page you can view your shopping cart. At the top of the cart are headings which describe to content of the table below (marked in red, see Figure 12).

Each row in the table represents one shopping cart item (marked in blue, see Figure 12).

To update the quantity in the shopping cart you alter the numeric text box (marked in pink, see Figure 12) and press the update button (marked in green, see Figure 12).

To remove an item from the shopping cart, set the numeric text box quantity (marked in pink, see Figure 12) to 0 (zero) (marked in yellow, see Figure 12) and press the update button (marked in green, see Figure 12). Note: the line item will remain in the cart but will not appear on the invoice. See subtotal is R0.00 (marked in purple, see Figure 12).

To view the individual price of each line item, see the column titled 'Price' (marked in black, see Figure 12).

To view the subtotal of a line item, see the column titled 'Subtotal' (marked in purple, see Figure 12).

To clear the contents of the shopping cart, press the CLEAR CART button (marked in orange, see Figure 12).

Supplier	Supplement	Price	Quantity	Change	Subtotal
SUPPLIER B	ID: 127 Description 90	R234.94	<input type="text" value="2"/>	UPDATE	R469.88
Supplier E	ID: 52 Description 500ml glass	R219.64	<input type="text" value="0"/>	UPDATE	R0.00

CLEAR CART **CHECKOUT**

Figure 12: Update Cart

How do I purchase items in my shopping cart?

First navigate to the shopping cart page. The shopping cart is accessed by clicking on the shopping cart icon. The icon is in the top-right corner of the page. For further instruction see the section titled 'How do I navigate the site', subsection 2.

Supplier	Supplement	Price	Quantity	Change	Subtotal
	ID: 127 Description 90	R234.94	<input type="text" value="2"/>	UPDATE	R469.88
	ID: 52 Description 500ml glass	R219.64	<input type="text" value="0"/>	UPDATE	R0.00

Figure 13: Checkout

Once you have reviewed your shopping cart (for further information see the section titled 'How do I alter my shopping cart') and are happy with your purchase, press the CHECKOUT button, in the lower right side of the page (marked in red, see Figure 13)

You will now be directed to the checkout page. Here you are required to fill in your personal information (marked in red, see Figure 14) and your delivery address information (marked in green, see Figure 14).

Take care to review your order (marked in purple, see Figure 14) before you submit your order by pressing the BUY NOW button (marked in blue, see Figure 14).

CHECKOUT

PERSONAL DETAILS

Name

Surname

Email

Home Phone

Work Phone

Cell Phone

Who referred you to us

ADDRESS

Line One

Line Two

Line Three

Line Four

Postal Code

ORDER SUMMARY

Total R469.88
(2 Items)

All items are in stock

BUY NOW

Payment via EFT

Figure 14

PERSONAL DETAILS		ADDRESS	
Name	\$	Only letter, space, and apostrophe allowed	
Surname	99	Only letter, space, and apostrophe allowed	
Email	lee@email.com		
Home Phone	721733939	Invalid Format, use (###)-(###)-(###)	
Work Phone	(000)-(000)-(0000)		
Cell Phone	(000)-(000)-(0000)		
Who referred you to us	ADVERT		
Line One	6 WOODLEA*****		Only letters, numbers and white space allowed
Line Two	FIRWOOD PARK		
Line Three	OLDHAM		
Line Four	ENGLAND		
Postal Code	000000		4 digits. 0000 for outside SA

Figure 15: Checkout errors

Once you press the BUY NOW button, if you are not directed to the ‘Order Complete’ page, you must first fix any errors in the forms that were completed. The errors for each form item will be displayed to the right of the item, in red text. It will contain a short description of what is required (marked in blue, see Figure 15).

Once all errors are rectified, your order will be processed and directed to the ‘Order Complete’ page. From here you will be able to see your online invoice. The invoice sections display:

- Your personal details (marked in blue, see Figure 16)
- Invoice overview (marked in red, see Figure 16)
- Payment details (marked in green, see Figure 16)
- A summary of the items that you ordered (marked in orange, see Figure 16)

Your details	Order details	Payment										
01 MIDDLE ROAD LEWISHAM LUTON BUCKS 0000	An overview of your order was sent to leeondet@yahoo.co.uk Client ID: 9911090794131 Invoice Number: INV1556 Date: 2018-09-12 15:39:19 Amount Payable: 2761.12	Make payment to Mrs Casey Milan Bank name: ABSA Account number: 45124561254 SMS proof of payment to: 0824712929 (use the invoice number as reference) : INV1556										
<h3>Item Summary</h3> <table border="1"> <thead> <tr> <th>Supplier</th> <th>Supplement ID</th> <th>Price</th> <th>Quantity</th> <th>Total</th> </tr> </thead> <tbody> <tr> <td>Supplier F</td> <td>112</td> <td>R2761.12</td> <td>Qty: 1</td> <td>R2761.12</td> </tr> </tbody> </table>			Supplier	Supplement ID	Price	Quantity	Total	Supplier F	112	R2761.12	Qty: 1	R2761.12
Supplier	Supplement ID	Price	Quantity	Total								
Supplier F	112	R2761.12	Qty: 1	R2761.12								

Figure 16: Online Invoice

How do I make a payment?

Payments are made via EFT to the person and bank that are listed on the 'Payment' section of your online invoice (marked in green, see Figure 17). Here you will find the:

- person's name to make payment to (marked in red, see Figure 17)
- bank's name (marked in purple, see Figure 17)
- account number (marked in yellow, see Figure 17)
- phone number to SMS the proof of payment, and the invoice number to use as a reference (marked in blue, see Figure 17)

The screenshot shows an online invoice interface with three main sections: 'Your details', 'Order details', and 'Payment'.

Your details (grey background):

- 01 MIDDLE ROAD
- LEWISHAM
- LUTON
- BUCKS
- 0000

Order details (white background):

- An overview of your order was sent to leondet@yahoo.co.uk
- Client ID: 9911090794131
- Invoice Number: INV1556
- Date: 2018-09-12 15:39:19
- Amount Payable: 2761.12

Payment (green rounded rectangle):

- Make payment to Mrs Casey Milan (red box)
- Bank name: ABSA (purple box)
- Account number: 45124561254 (yellow box)
- SMS proof of payment to: 0824712929 (use the invoice number as reference) : INV1556 (blue box)

Item Summary (grey background):

Supplier	Supplement ID	Price	Quantity	Total
	112	R2761.12	Qty: 1	R2761.12

Figure 17: How to make a payment



MY VITALITY

Alternative Healthcare Solutions

System User Manual

MY VITALITY: ALTERNATIVE HEALTHCARE SOLUTIONS

System User Manual: Administration

© My Vitality
01 Diagonal Street • Johannesburg
2001, South Africa
Phone 0112345678 • Fax 0112345679
Email support@myvitality.com

Table of Contents

How do I navigate the site?	3
How do I view management information reports?	4
What management information reports are available?	4
How do I view pending transactions?	11
What do the status indicators mean?.....	12
How do I confirm a customer's payment?.....	13
How do I reject an order?.....	14
How do I view the status of an order?.....	15
How do I view orders that are ready to be shipped?	16
How do I record that an order has been shipped to the customer?	17
How do I view orders that have been shipped?	19
How do I view stock levels?	20
How do I add new stock to the system?	22
How do I view what stock has been added to the system?.....	24
How do I log into the system?	25
How do I logout of the system?	26
What do I do if I forget my password?.....	27
How do I start the system?	28

How do I navigate the site?

To navigate MyVitality, there are two options:



- 1) Navigate to the menu at the top of the page and click one of the menu options.

Figure 8

- 2) The administration site home page contains three icons that may be clicked to take you to a different section of the site.

Hover over any of the light blue icons circled in red (see Figure 2) and when the icon changes to orange it may be clicked.



Figure 9

How do I view management information reports?

To view management information reports, you must navigate to the Management Information System reporting section of the system. To do that, you use the navigation bar at the top of the web page and click on the menu option 'MIS' (marked in red, see Figure 3).

Once you are directed to the MIS dashboard you can view a variety of reports. Each report has an option to click on a link so that you can view historical data. For more information on the type of reports or the structure of the MIS dashboard, refer to the section titled 'How do I view management information reports?'.



Figure 10

What management information reports are available?

The MIS reports cover the following categories of information:

- Finance (marked in red, see Figure 4)
 - Total Profit (marked in red, see Figure 5)
 - In last month
 - Link to historical data
 - Total Tax (marked in green, see Figure 5)
 - In last month
 - Link to historical data
 - Total Sales (marked in blue, see Figure 5)
 - In last month
 - Link to historical data
 - Average Order Amount (marked in orange, see Figure 5)
 - In last month
 - Link to historical data
- Orders (marked in green, see Figure 4)

- Number of orders (marked in red, see Figure 6)
 - In last month
 - Link to historical data
 - Number of cancelled orders (marked in green, see Figure 6)
 - In last month
 - Link to historical data
 - Number of pending orders (marked in blue, see Figure 6)
 - Current orders that are pending approval
 - Link to Invoice numbers pending approval
 - Number of orders waiting to be shipped (marked in orange, see Figure 6)
 - Current orders that are waiting to be shipped
 - Link to Invoice numbers waiting to be shipped
- Inventory (marked in orange, see Figure 4)
 - Number of items that need to be re-ordered (marked in red, see Figure 7)
 - Link to Supplement IDs
 - Number of items that are within 10% of the re-order level (marked in green, see Figure 7)
 - Link to Supplement IDs
 - Number of items that were not ordered in the last month (marked in blue, see Figure 7)
 - Link to Supplement IDs
 - Number of items that were not ordered in the last year (marked in orange, see Figure 7)
 - Link to Supplement IDs
- Pricing (marked in purple, see Figure 4)
 - The lowest profit margin placed on a Supplement (marked in red, see Figure 8)
 - Link to Supplement IDs
 - The highest profit margin placed on a supplement (marked in green, see Figure 8)

- Link to Supplement IDs
- Sales (marked in yellow, see Figure 4)
 - Breakdown of sales made by supplier in the last month (marked in red, see Figure 9)
 - Link to supplements sold by supplier, all time
 - Breakdown of orders per region in the last month (marked in green, see Figure 9)
 - Link to supplements sold by region, all time
 - A list of the top ten supplements sold in the last month (marked in blue, see Figure 10)
 - Link to the top ten supplements sold, all time
 - Breakdown of how customers were referred to the company in the last month (marked in orange, see Figure 11)
 - Link to how customers were referred to the company, all time



Figure 11

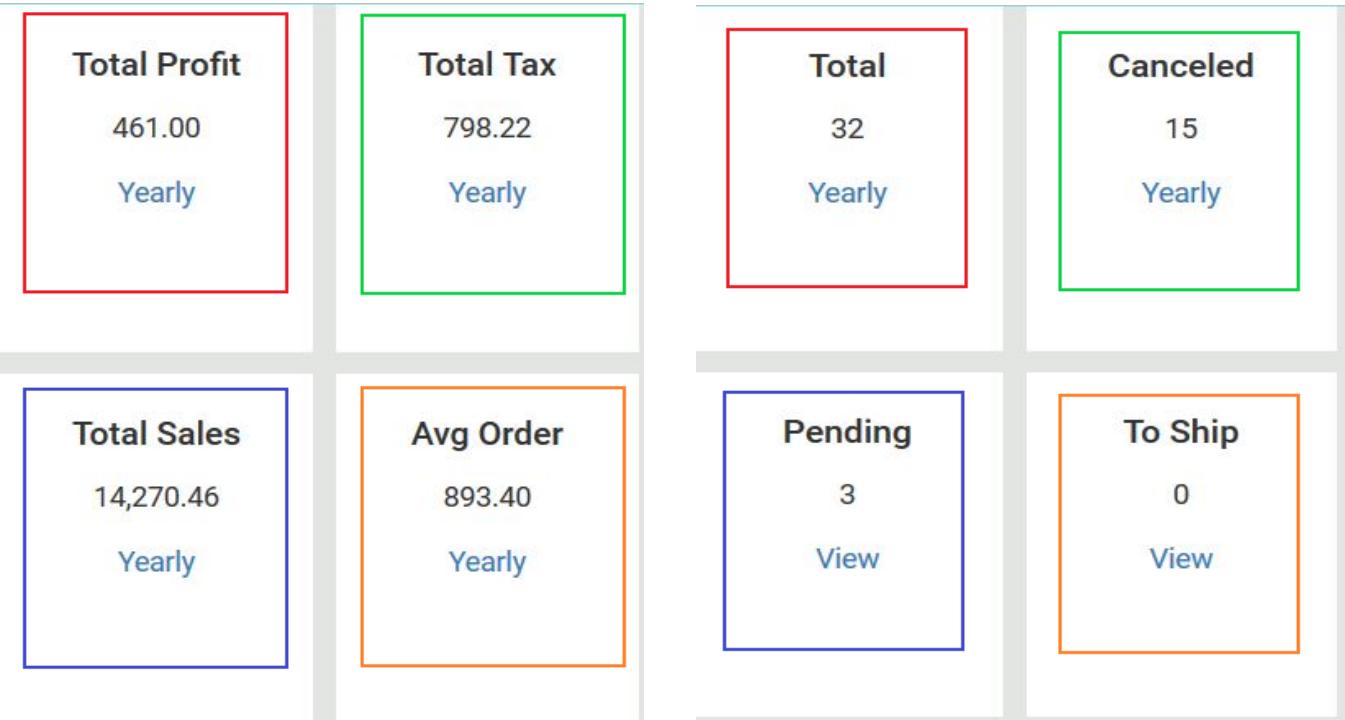


Figure 5

Figure 6



Figure 7

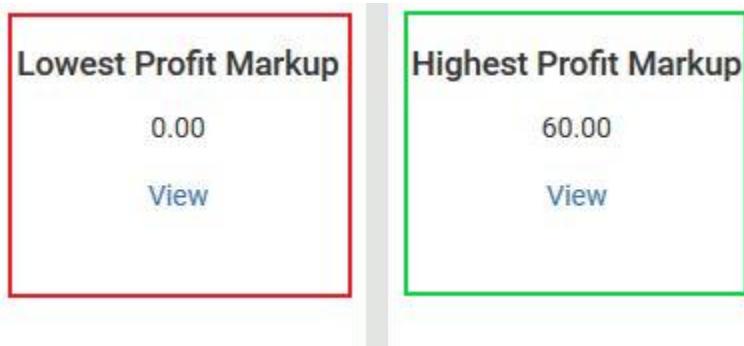


Figure 8



Figure 9

Top Ten Sold Month	
Supplement ID	Quantity
236	6
50	6
127	6
112	5
1	3
93	2
157	2
52	2
226	2
134	2

[All Top Ten](#)

Figure 10

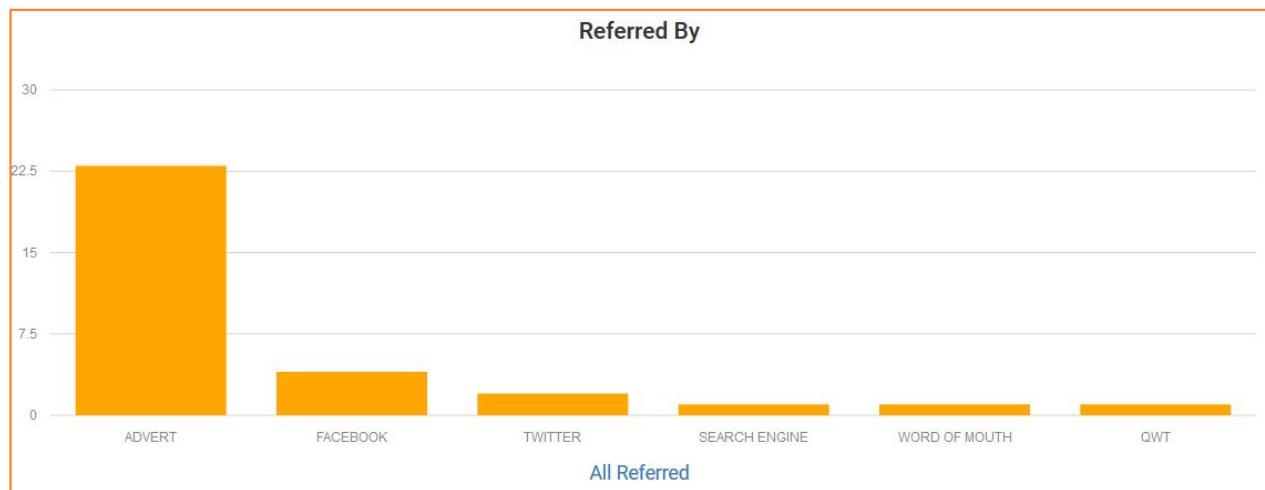


Figure 11

How do I view pending transactions?

To view pending transactions, navigate to the Orders section of the system. To do that, use the navigation bar at the top of the web page and click on the menu option ‘Order’ (marked in red, see Figure 12).



Figure 12

You will be directed to the ‘Pending Orders’ page. A variety of information about pending transactions will be displayed:

- Table headings (marked in brown, see Figure 13)
- Invoice ID (marked in orange, see Figure 13). The invoice ID is clickable. When clicked, the contents of the invoice will appear in a pop-up box (marked in red, see Figure 14)
- Customer ID associated with a specific invoice (marked in dark red, see Figure 13)
- Date the invoice was generated (marked in green, see Figure 13)
- Total amount due by the customer (marked in light blue, see Figure 13)
- Amount that has been paid by the customer (marked in purple, see Figure 13)
- ‘ENTER PAYMENT’ button. The button can be clicked to open a pop-up box where the user can enter a customer’s payment. See the section titled ‘How do I confirm a customer’s payment?’ for more information.
- The status of an order (marked in red, see Figure 13). For more information on status indicators, see the section titled ‘What do the status indicators mean?’.
- ‘REJECT’ button. The button can be clicked to reject an order (marked in brown, see Figure 13)

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1545	9911090794120	2018-09-04	R219.64	R0.00	ENTER PAYMENT	PENDING	REJECT
INV1546	9911090794121	2018-09-04	R150.48	R0.00	ENTER PAYMENT	PENDING	REJECT

Figure 13

MYVITALITY ALTERNATIVE HEALTHCARE							
INV1555							
Invoice ID	Customer ID	Supplement ID	Quantity	Item Price	Total Price	Status	Action
INV1555	9911090794130	ID: 134	1	R261.16	R261.16	CLOSE	PENDING REJECT
INV1558	9911090794133		2018-09-19	R4,825.72	R0.00	ENTER PAYMENT PENDING	REJECT

Figure 14

What do the status indicators mean?

A status indicator shows the user what stage the order is in the order fulfillment process. The status indicators used are as follows:

- PENDING: The order has yet to be confirmed or rejected
- APPROVED: The order has been approved for shipment
- REJECTED: The order has been rejected.
- SHIPPED: The order has been shipped to the customer

Status indicators are listed under the heading ‘Status’. An example is shown in Figure 15, circled in red.

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1545	9911090794120	2018-09-04	R219.64	R0.00	ENTER PAYMENT	PENDING	REJECT
INV1546	9911090794121	2018-09-04	R150.48	R0.00	ENTER PAYMENT	PENDING	REJECT

Figure 15

How do I confirm a customer's payment?

To confirm a customer's payment, navigate to the 'Pending Orders' page, located in the Orders section of the system. For more information on how to do this, see the section titled 'How do I view pending transactions?'.

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1545	9911090794120	2018-09-04	R219.64	R0.00	ENTER PAYMENT	PENDING	REJECT
INV1546	9911090794121	2018-09-04	R150.48	R0.00	ENTER PAYMENT	PENDING	REJECT

Figure 16

From here you click the 'ENTER PAYMENT' button (marked in red, see Figure 16 above). A pop-up box will appear (marked in red, see Figure 17 below).

The screenshot shows the IVVITALITY software interface. At the top left is the logo 'IVVITALITY' with the tagline 'ALTERNATIVE HEALTHCARE'. Below the logo is a navigation bar with tabs 'Home', 'Payment', and 'Stock'. The main area displays a table of pending orders:

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1546	9911090794121	2018-09-04	R150.48	R0.00	ENTER PAYMENT	PENDING	REJECT

A modal dialog box is overlaid on the screen, centered over the INV1546 row. The dialog has a light blue header with the text 'Payment For: INV1546'. The body of the dialog contains a 'Payment' input field with the value '0.00'. At the bottom of the dialog are two buttons: 'CLOSE' (grey) and 'ADD PAYMENT' (orange). The 'ENTER PAYMENT' button in the main table row is highlighted with a red oval.

Figure 17

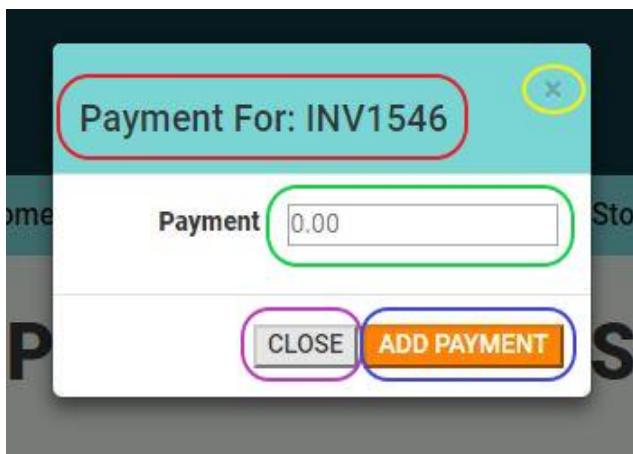


Figure 18

Confirm that it is the correct invoice, by double checking the invoice ID in the heading of the box (marked in red, see figure 18) before entering a payment.

To enter a payment, click on the payment box and enter the required payment (marked in green, see figure 18).

To confirm the payment, press the 'ADD PAYMENT' button (marked in blue, see figure 18).

To exit the payment box without confirming the payment, press either the 'x' in the top right corner of the pop-up box (marked in yellow, see figure 18), or

press the ‘CLOSE’ button in the footer of the pop-up box (marked in purple, see figure 18)

Once the box has been closed, you can check the status of the order (marked in red, see figure 19).

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1546	9911090794121	2018-09-04	R150.48	R150.48	ENTER PAYMENT	APPROVED	REJECT

Figure 19

If you enter a payment by mistake, you can reset the amount by entering 0.00 into the payment box shown in figure 18.

How do I reject an order?

To reject an order, navigate to the ‘Pending Orders’ page, located in the Orders section of the system. For more information on how to do this, see the section titled ‘How do I view pending transactions?’.

Once you are at the ‘Pending Orders’ page, to reject an order, click the ‘REJECT’ button in the action column (marked in green, see figure 20). Make sure that the button you click is in the row associated with

Invoice ID	Customer ID	Date	Total Due	Amount Paid	Payment Amount	Status	Action
INV1545	9911090794120	2018-09-04	R219.64	R0.00	ENTER PAYMENT	PENDING	REJECT
INV1546	9911090794121	2018-09-04	R150.48	R0.00	ENTER PAYMENT	PENDING	REJECT

Figure 20

the Invoice ID (marked in red, see figure 20).

If you reject an order by accident, you can re-enter a payment to override the reject operation. To do this, follow the instructions in the section titled ‘How do I confirm a customer’s payment?’.

How do I view the status of an order?

To view the status of an order, navigate to the Orders section of the system. To do that, choose one of the Order menu options located in the navigation bar at the top of the page (marked in red, see figure 21).

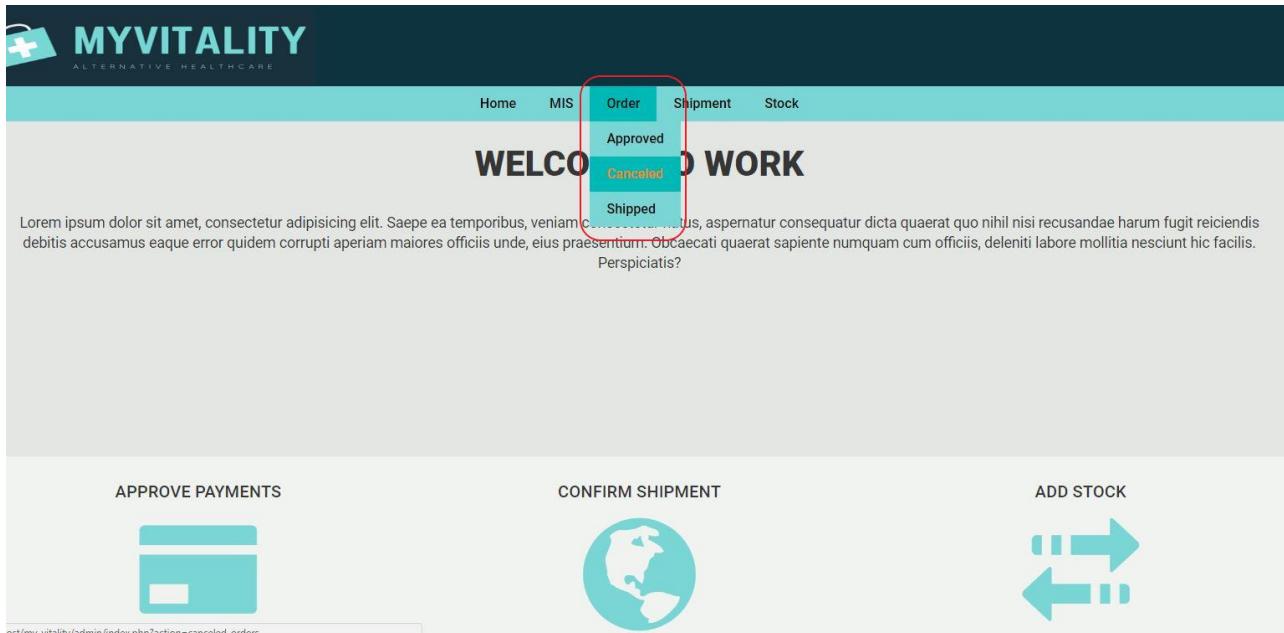


Figure 21

To see pending transactions, chose ‘Order’;

To see approved transactions, choose ‘Approved’;

To see rejected transactions, choose ‘Cancelled’;

To see shipped orders, choose ‘Shipped’;

For more information on what the headings on the pages indicate, see the section titled ‘How do I view pending transactions?’.

For more information on status indicators, see the section titled ‘What do the status indicators mean?’.

How do I view orders that are ready to be shipped?

To view orders that are ready to be shipped, navigate to the ‘Shipment’ page, located in the Shipments section of the system. To do that, use the navigation bar at the top of the page and choose the ‘Shipment’ option (marked in red, see figure 22).



Figure 22

Once you have been directed to the Shipment page, you will be able to view the following information:

- Table headings (marked in brown, see Figure 23)
- Invoice ID (marked in orange, see Figure 23). The invoice ID is clickable. When clicked, the contents of the invoice will appear in a pop-up box.
- Customer ID associated with a specific invoice (marked in dark red, see Figure 23)
- Customer’s delivery address (marked in green, see Figure 23)
- Courier used to ship the order. If the order is yet to be shipped, this section will be blank (marked in purple, see Figure 23). If the order has been shipped, the courier used will be displayed here.
- The status of an order (marked in blue, see Figure 23 and Figure 24). For more information on status indicators, see the section titled ‘What do the status indicators mean?’.
- The ‘SHIP ORDER’ button (marked in yellow, see Figure 23). This button can be clicked to confirm that the order has been shipped.

A screenshot of the 'SHIPMENT' page. The page title is 'SHIPMENT' in large, bold, black capital letters. Below the title is a horizontal table with six columns: 'Invoice ID', 'Customer ID', 'Address', 'Courier', 'Status', and 'Confirm Shipment'.

- The 'Invoice ID' field contains the value '1542' in a blue box.
- The 'Customer ID' field contains the value '9911090794117' in a red box.
- The 'Address' field contains the value '6 WOODLEA FIRWOOD PARK OLDHAM LANCS 0000' in a green box.
- The 'Courier' field is empty and highlighted with a purple box.
- The 'Status' field contains the value 'APPROVED' in a blue box.
- The 'Confirm Shipment' button is a large orange button labeled 'SHIP ORDER'.

Figure 23

Invoice ID	Customer ID	Address	Courier	Status	Confirm Shipment
1542	9911090794117	6 WOODLEA FIRWOOD PARK OLDHAM LANCS 0000	DHL EXPRESS	SHIPPED 2018-09-14	SHIP ORDER

Figure 24

How do I record that an order has been shipped to the customer?

To record that an order has been shipped to a customer, navigate to the ‘Shipment’ page, located in the Shipments section of the system. To do that, use the navigation bar at the top of the page and choose the ‘Shipment’ option (marked in red, see figure 25).



Figure 25

Once you have been directed to the Shipment page, you can view the contents of the order by clicking on the invoice number order (marked in red, see Figure 26). Press the ‘SHIP ORDER’ button (marked in green, see Figure 26) and wait for the pop-up box to load. For more information on the data displayed on this page, see the section titled ‘How do I view orders that are ready to be shipped?’.

SHIPMENT					
Invoice ID	Customer ID	Address	Courier	Status	Confirm Shipment
1542	9911090794117	6 WOODLEA FIRWOOD PARK OLDHAM LANCS 0000		APPROVED	SHIP ORDER

Figure 26

Now that the pop-up box has been loaded, you will be presented with the following:

- a header that contains the invoice number (marked in orange, see Figure 27),
- a drop-down menu to select the courier that was used to ship the order (marked in red, see Figure 27), and
- a ‘CONFIRM’ button (marked in purple, see Figure 27),
- a table that contains the information of the couriers (marked in blue, see Figure 27),
- a ‘CLOSE’ button (marked in green, see Figure 27), and a ‘X’ in the top right corner of the pop-up (marked in yellow, see Figure 27), to exit the pop-up without processing the request.

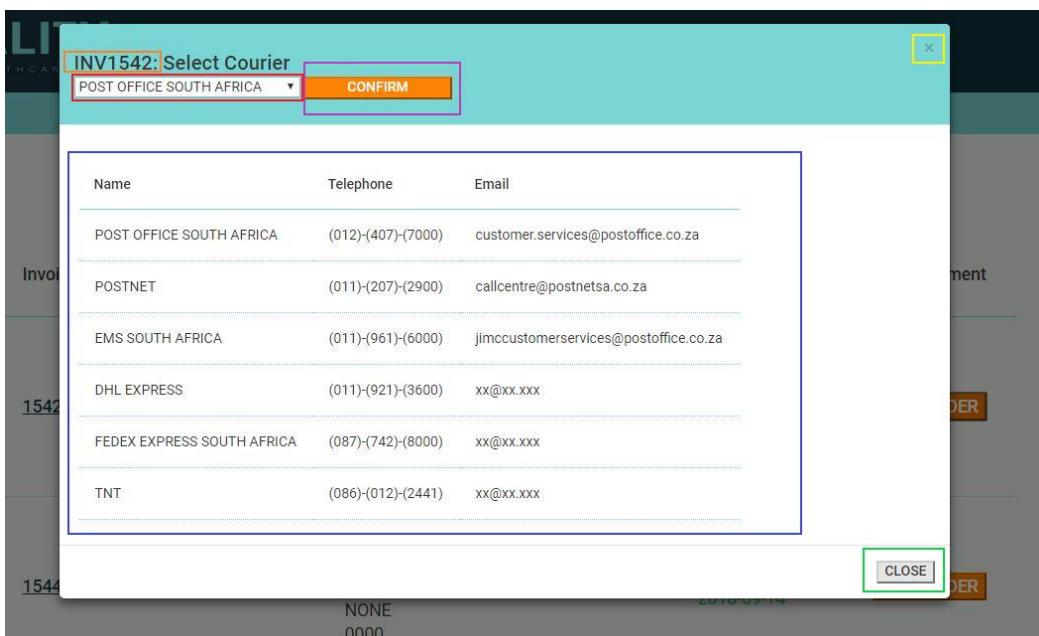


Figure 27

Once you have confirmed the shipment, check the status of the order (marked in blue, see Figure 28), and the courier selected (marked in purple, see Figure 28), is correct.

Invoice ID	Customer ID	Address	Courier	Status	Confirm Shipment
1542	9911090794117	6 WOODLEA FIRWOOD PARK OLDHAM LANCS 0000	DHL EXPRESS	SHIPPED 2018-09-14	SHIP ORDER

Figure 28

How do I view orders that have been shipped?

To record that an order has been shipped to a customer, check that the status of an order has changed when you record that an order has been shipped. For more information on how to do this, see the section titled ‘How do I record that an order has been shipped to the customer?’.

Alternatively, you can navigate to the ‘Sent’ page with a dropdown menu option below ‘Shipment’. To do that, use the navigation bar at the top of the page and hover over the ‘Shipment’ (marked in red, see figure 25) option to reveal the dropdown menu and click ‘Sent’. From this page you can view the following information that is associated with a shipment:

- table header (marked in yellow, see Figure 29),
- invoice number (marked in purple, see Figure 29),
 - the invoice number can be clicked to open a window showing the contents of the shipment
- the customer ID (marked in blue, see Figure 29),
- customers delivery address (marked in green, see Figure 29),
- courier used to ship the order (marked in red, see Figure 29),
- the status of the order and the date it was shipped (marked in orange, see Figure 29)

Invoice ID	Customer ID	Address	Courier	Status
1554	9911090794129	01 MIDDLE ROAD LEWISHAM LUTON BUCKS 0000	DHL EXPRESS	SHIPPED 2018-09-19
1551	9911090794126	6 WOODLEA NONE OLDHAM NONE 0000	POST OFFICE SOUTH AFRICA	SHIPPED 2018-09-19

Figure 29

How do I view stock levels?

To view stock levels, navigate to the 'Stock' page, located in the Inventory section of the system. To do that, use the navigation bar at the top of the page and choose the 'Stock' option (marked in red, see Figure 30).



Figure 30

Once you have been directed to this page, you can view the stock levels of all the supplements that are stored in the system. The following information is displayed:

- table header (marked in black, see Figure 31)
- Supplement ID (marked in light blue, see Figure 31). The supplement ID can be clicked to show the details of how it can be purchased (marked in red, see Figure 32 two figures down).
- Supplier name (marked in red, see Figure 31)
- The total amount of stock in the system (marked in blue, see Figure 31)
- The amount of stock that is on hold, waiting for payment approval (marked in purple, see Figure 31)
- The amount of stock that is available for sale (marked in pink, see Figure 31)
- The amount at which stock needs to be re-ordered (marked in brown, see Figure 31)
- A status indicating the stock levels for each supplement
 - A green 'IN STOCK' indicator means that the total stock level is above the re-order level (marked in green, see Figure 31)
 - An orange 'IN STOCK' indicator means that the total stock level is above the re-order level, but that it is within 10% of the re-order level. This means that the stock is close to needing to be re-ordered (marked in orange, see Figure 31)
 - A red 'REPLENISH' indicator means that the supplement is at or below the re-order level and that it needs to be replenished (marked in dark red, see Figure 31)

When the supplement ID is clicked, a pop-up box will be displayed (marked in red, see Figure 32), showing the following information:

- The supplier name and any comments about that supplier (marked in green, see Figure 32)
- The contact details for a specific person(s) at the supplier (marked in orange, see Figure 32)

- The banking details of the supplier (marked in blue, see Figure 32)
- Buttons to close the pop-up window (marked in yellow, see Figure 32)

Supplement ID	Supplier	Total Stock	On Hold	For Sale	Reorder Level	Status	Add Stock
1	SUPPLIER A	56	0	56	10	IN STOCK	ADD STOCK
2	SUPPLIER B	41	1	40	1	IN STOCK	ADD STOCK
3	SUPPLIER C	22	1	21	1	IN STOCK	ADD STOCK
4	SUPPLIER B	11	0	11	10	IN STOCK	ADD STOCK
5	SUPPLIER B	6	0	6	8	REPLENISH	ADD STOCK

Figure 31

Supplier Contact Details For Supplement ID: 4

Supplier ID	Supplier Name Supplier Comments				
1	SUPPLIER B	NONE			
2	Name	Surname	Email	Phone	Type
3	Mary	Nkosi	mary@webmail.com	(011)-(894)-(9004)	TEL
4	Bank Name	Branch Code	Account Number	Account Type	
5	STANDARD BANK	23460	420315985	CHEQUE	

CLOSE

Figure 32

How do I add new stock to the system?

To add stock to the system, navigate to the ‘Stock’ page, located in the Inventory section of the system. For more information on how to do this, navigate to the section titled ‘How do I view stock levels?’.

Once you are at this page, click on the button titled ‘ADD STOCK’, (marked in red, see Figure 33) and a pop-up window will appear (marked in red, see Figure 34).

Supplement ID	Supplier	Total Stock	On Hold	For Sale	Reorder Level	Status	Add Stock
1	SUPPLIER A	56	0	56	10	IN STOCK	ADD STOCK

Figure 33

On the pop-up window you will find the following information:

- Supplement ID of the item you selected (marked in orange, see Figure 34)
- The date it is logged to the system (marked in purple, see Figure 34)
- A text box where you enter the cost price per item, excluding VAT (marked in green, see Figure 34)
- A text box where you enter the quantity of stock purchased (marked in brown, see Figure 34)
- Buttons that will exit the window without adding a transaction (marked in yellow, see Figure 34)
- A ‘ADD STOCK’ button that will add the details entered to the system. (marked in blue, see Figure 34)

In the pop-up window, enter the following details, and then press the ‘ADD STOCK’ button to complete the transaction, or one of the close buttons to exit without completing the transaction.

- The cost price excluding VAT paid per item
- The quantity purchased

Note:

- The cost including VAT is calculated automatically at a rate of 14%
- If the cost entered is different than the current cost stored in the system, a new cost price will automatically be generated.

- The client cost will automatically be generated by the system, to include VAT and markup costs.

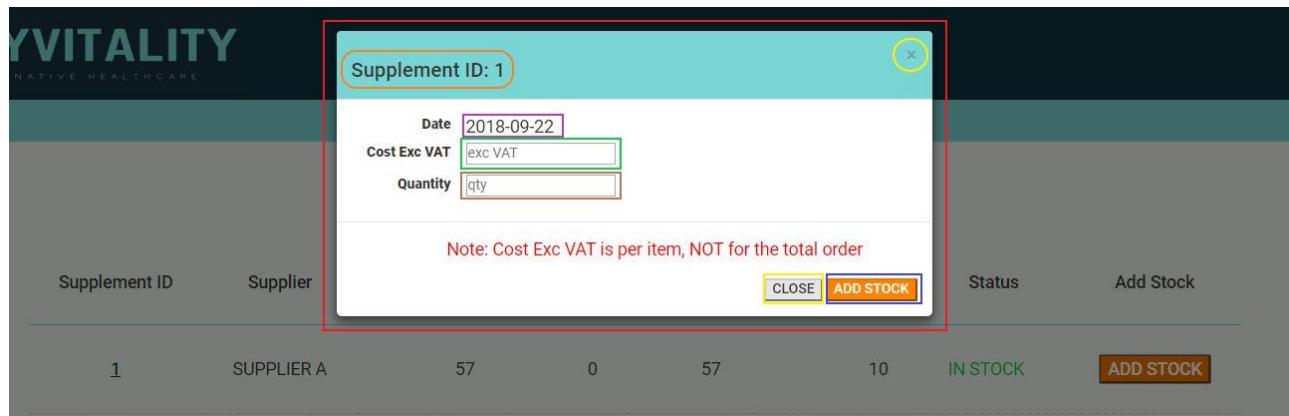


Figure 34

How do I view what stock has been added to the system?

To view the stock that was added to the system, navigate to the 'Journal' page, located in the dropdown menu of the 'Stock' page. For more information where the 'Stock' page menu option is located, navigate to the section titled 'How do I view stock levels?'.

Once you are directed to this page, you can view the following information:

- Table header (marked in black, see Figure 35)
- Date stock was purchased (marked in orange, see Figure 35)
- The cost of each item excluding VAT (marked in blue, see Figure 35)
- The cost of each item including VAT (marked in yellow, see Figure 35)
- The quantity purchased (marked in brown, see Figure 35)
- The supplier ID (marked in red, see Figure 35)
- The supplement ID (marked in green, see Figure 35)

PURCHASE JOURNAL					
Date	Cost Exc VAT	Cost Inc VAT	Quantity	Supplier ID	Supplement ID
2018-09-19	195.86	223.28	2	SUPPLIER C	10
2018-08-25	201.00	229.14	5	SUPPLIER B	5

Figure 35

How do I log into the system?

To start the administration system, enter the following URL into your browser
<http://myvitality.tk/admin/>

When you start the administration system, you will be directed to the login screen (marked in blue, see Figure 36).

Enter your user name and password into the username (marked in yellow, see Figure 36) and password boxes (marked in red, see Figure 36). Then press the 'LOGIN' button (marked in orange, see Figure 36) to complete the login process.

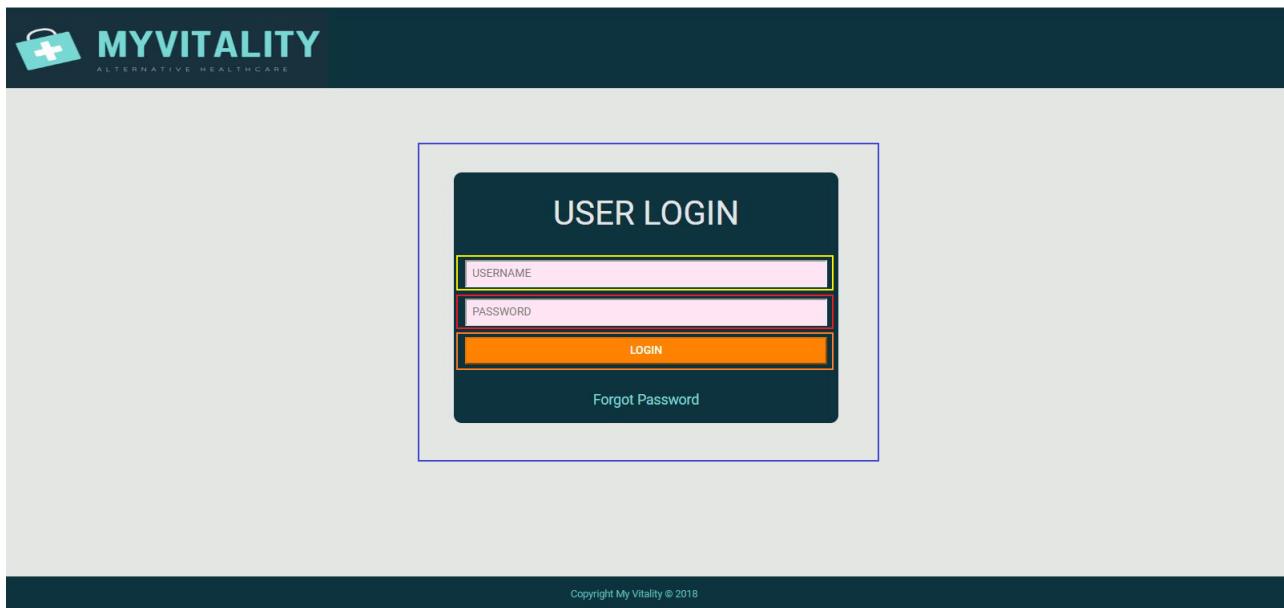


Figure 36

Your username and password will be provided by system administrator.

Usernames and passwords adhere to the following rules:

- Must be at least 8 characters in length
- Must be a maximum of 30 characters in length
- Can contain uppercase letters
- Can lowercase letters
- Can contain numbers 0 through 9

If there is an error logging into the system, see the error message in the login box (marked in red, see Figure 37)



Figure 37

How do I logout of the system?

To logout of the system, click on the icon or 'Logout' text in the upper right-hand corner of the screen (marked in red, see Figure 38).

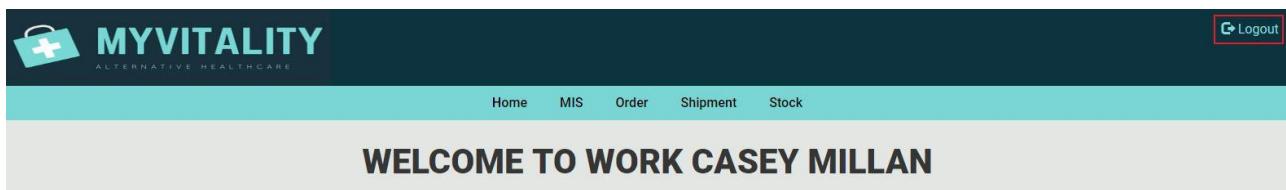


Figure 38

What do I do if I forget my password?

If you forget your password, on the login screen, click the link ‘Forgot Password’ (marked in red, see Figure 39). A popup window will now appear (marked in red, see Figure 40) where you can enter your username (marked in green, see Figure 40) and then press the ‘REQUEST’ button to request your new password (marked in purple, see Figure 40). You may close the window without requesting a new password by clicking on either of the close buttons (marked in yellow, see Figure 40). The system administer will contact you to provide you with a new password.

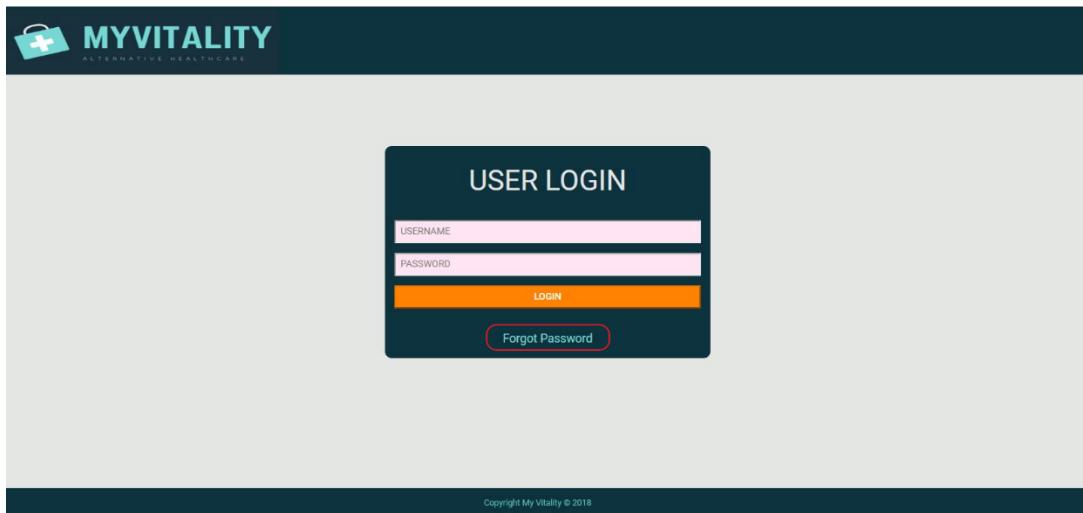


Figure 39



Figure 40

How do I start the system?

To start the system, start your browser and enter the following URL into the search bar.

<http://myvitality.tk/admin/> (marked in red, see Figure 41)

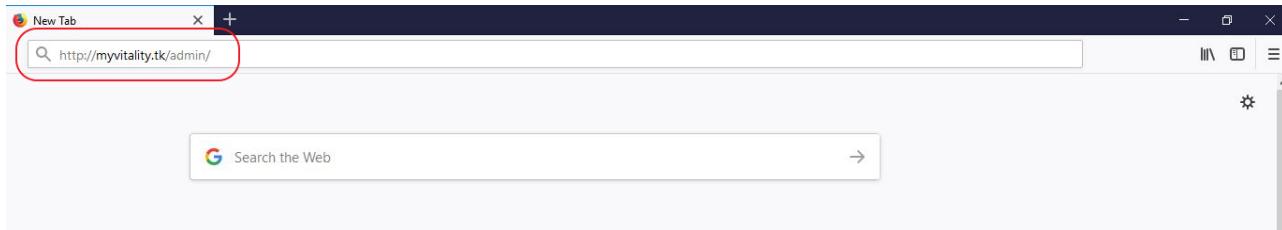


Figure 41

Installation manual



MY VITALITY

Alternative Healthcare Solutions

Installation Manual

MY VITALITY: ALTERNATIVE HEALTHCARE SOLUTIONS

Installation Manual

© My Vitality
01 Diagonal Street • Johannesburg
2001, South Africa
Phone 0112345678 • Fax 0112345679
Email support@myvitality.com

Table of Contents

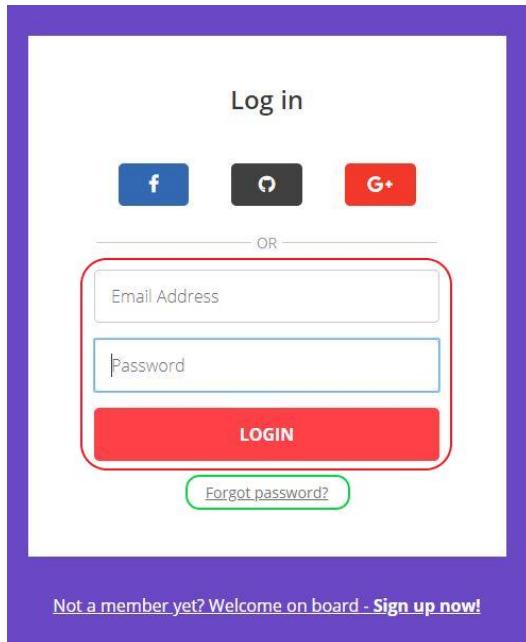
Where is the system hosted?	2
How do I access the hosting site?	2
How do I navigate the hosting site?	3
How do I access the database?	5
What can you tell me about the domain name?	6
What software is needed?	7
How do I install the software?	7

Where is the system hosted?

The site is hosted online at a company called Hostinger.

<https://www.hostinger.com/>

How do I access the hosting site?



To access the hosting site, enter the following URL into the browser:

<https://www.hostinger.com/cpanel-login?r=index/index>

From here, enter the login information to access the site (see figure 1, marked in red) and press the 'LOGIN' button.

If you have forgotten the password, click the 'Forgot password' link (see Figure 1, marked in green) and follow the onscreen instructions to reset the password.

Figure 12

How do I navigate the hosting site?

Once you have logged in, you'll be presented with the screen shown in Figure 2.

You can use the navigation bar at the top of the page to view your profile, account and billing information (see Figure 2, marked in green).

If you require any help, you can click the icon in the bottom right-hand corner of the page to start a chat with one of the site's agents (see Figure 2, marked in orange).

To manage the MyVitality system, click the 'Manage' button in the center of the page (see Figure 2, marked in red).

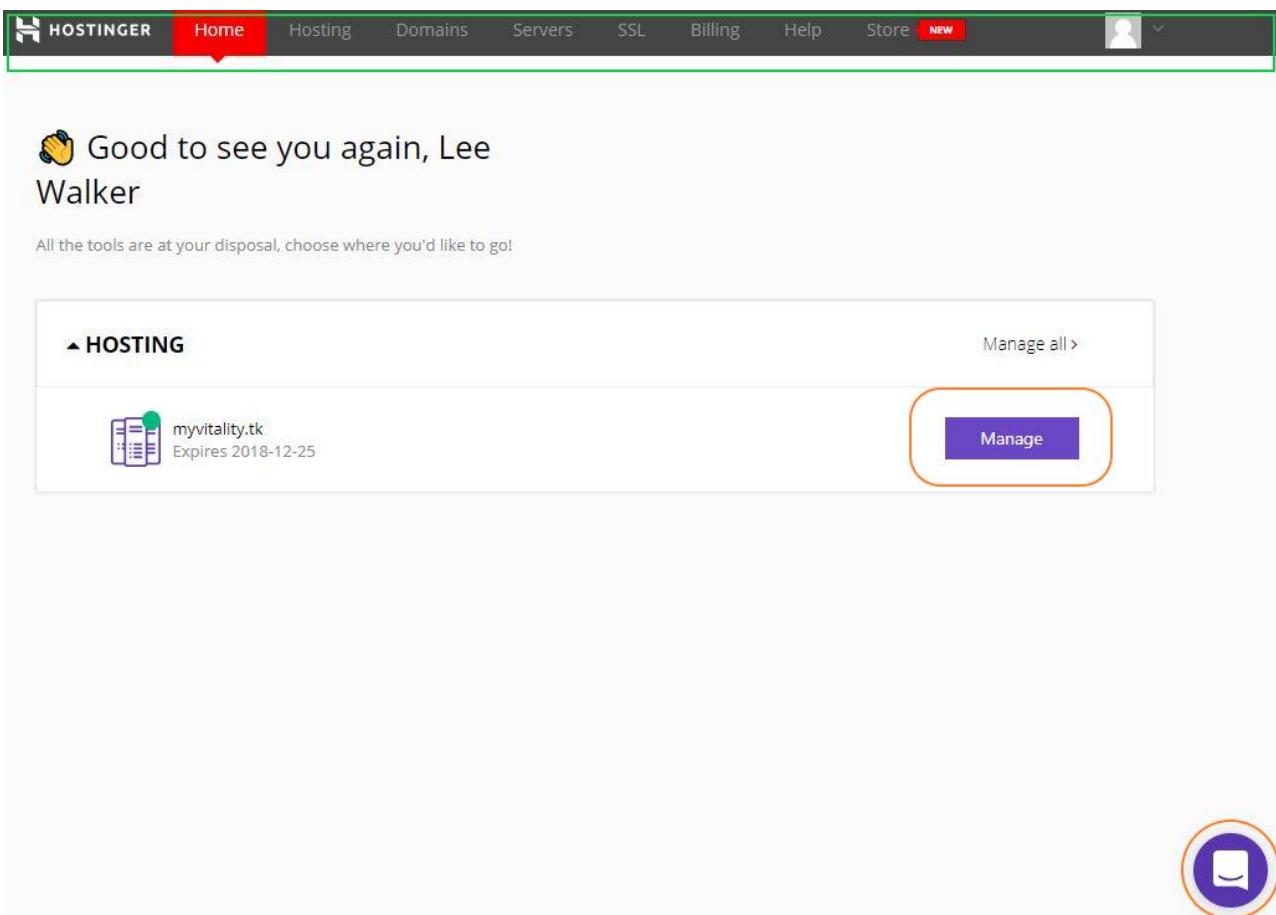


Figure 13

Once you click the 'Manage' button you will be directed to a page where you can manage the online system.

From here you can:

- Manage your account
 - See account details
 - Change account password
 - Renew or upgrade your subscription
- Manage email accounts
- Manage the domain name
- Import web site code
- Manage the web site files and make backups
- Manage the site database (for further information, see the section titled 'How do I access the database?')
- Manage web site advanced settings
 - Add site encryption
 - Alter the PHP configuration

To do any of the above tasks, scroll down the page and click on the icon that represents the task you would like to complete.

How do I access the database?

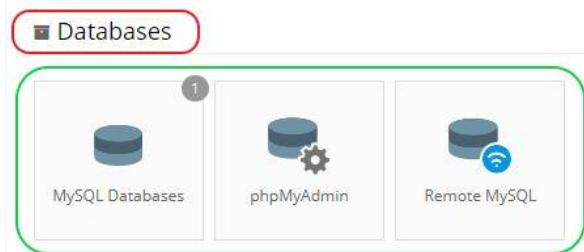


Figure 14

To add a new database to the site, click the icon titled 'MySQL Databases'.

To manage a remote database, click the icon titled 'Remote MySQL'.

To access the existing database, click the icon titled 'phpMyAdmin'. When you click this button, you will be directed to a page that will show the list of current databases associated with the system (see Figure 4, marked in red). From here, click the 'Enter phpMyAdmin' button (see Figure 4, marked in green) to be directed to the phpMyAdmin interface where you can manage the site's existing database.

List of Current Databases		
Mysql Database	Mysql User	Actions
u363377052_myv	u363377052_lee	Enter phpMyAdmin
		← Previous 1 Next →

Figure 15

What can you tell me about the domain name?

The domain name for the system is myvitality.tk

The domain name was obtained from <https://www.freenom.com/en/index.html?lang=en>

The domain name was purchased for free.

To manage the domain, go to the URL above and enter the login information. Once logged in click on ‘Services’ in the navigation bar at the top of the page (see Figure 5, marked in red) and then select either ‘My Domains’ (see Figure 5, marked in green) or ‘Renew Domains’ (see Figure 5, marked in orange) to manage the domain name.

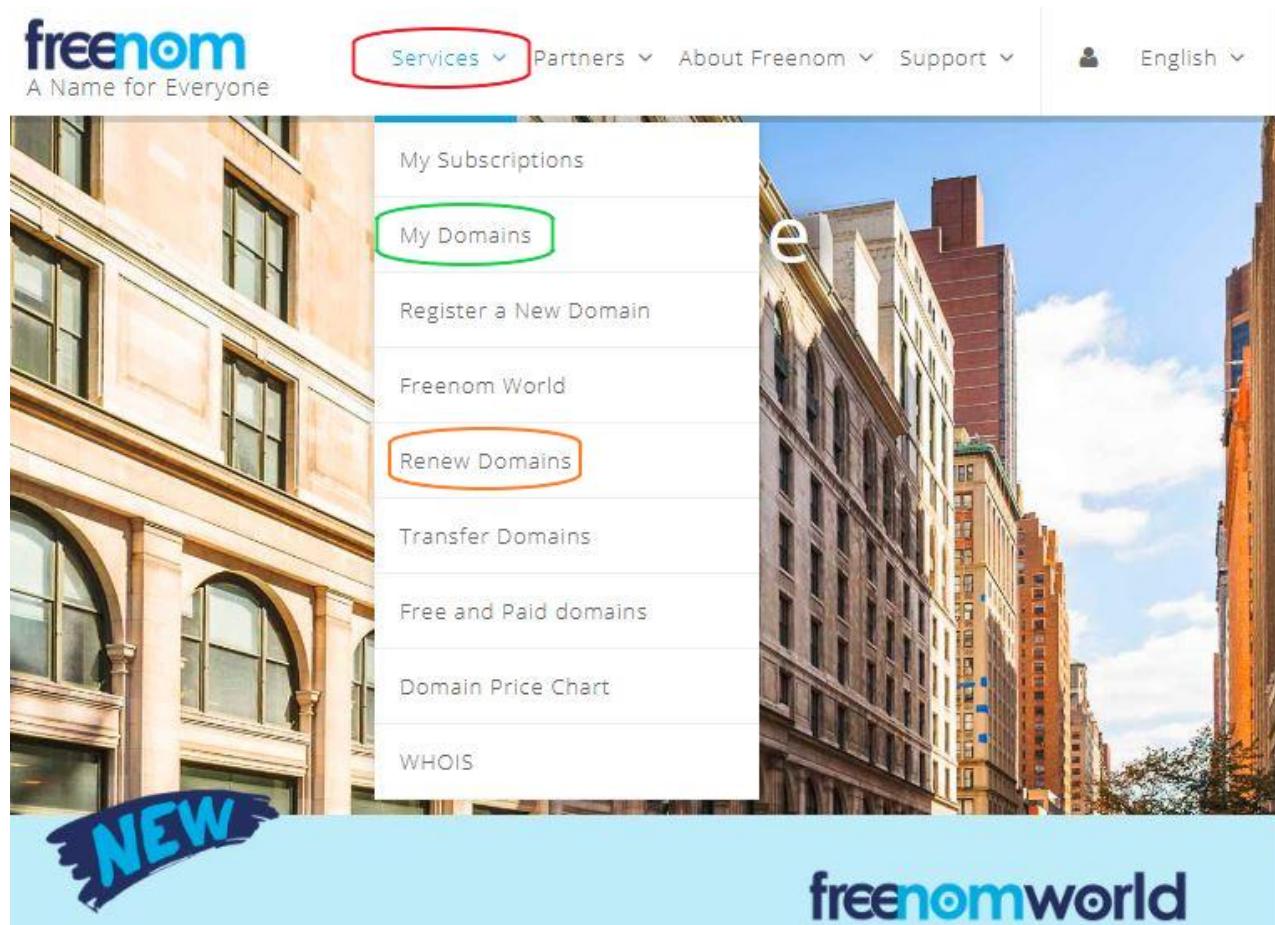


Figure 16

What software is needed?

A browser such as Google Chrome or Mozilla Firefox.

A web server stack such as XAMPP for using the site offline.

phpMyAdmin for database administration.

The software can be obtained from the following locations

- Google Chrome -

<https://support.google.com/chrome/answer/95346?co=GENIE.Platform%3DDesktop&hl=en>

- Mozilla Firefox - <https://www.mozilla.org/en-US/firefox/new/>
- phpMyAdmin - <https://www.phpmyadmin.net/downloads/>

How do I install the software?

For help on installing the software, please visit the vendors documentation manuals, provided below:

- Google Chrome -

<https://support.google.com/chrome/answer/95346?co=GENIE.Platform%3DDesktop&hl=en>

- Mozilla Firefox- <https://support.mozilla.org/en-US/kb/how-download-and-install-firefox-windows>
- phpMyAdmin - <https://www.phpmyadmin.net/docs/>

Module ICT3715

Year 2018

INFORMATION AND COMMUNICATION TECHNOLOGY (PROJECT IMPLEMENTATION)

STUDENT NUMBER (Student completes)									
5	6	6	0		2	8	0		4

IDENTITY NUMBER (Student completes)									
5	1	1	1	6	2	9	8	0	

EVALUATION OPTION (Student completes)										
Face-to-face		X	Regional Office		Florida Campus					
Online			Google Hangouts user name							

INSTRUCTIONS:

Instructions for the student:

Complete this front page (page 1).

Complete the Plagiarism Pledge (page 4), Section A (page 5).

After you've completed the front page with your information, the plagiarism pledge and section A save the document as a PDF and upload it to myUNISA together with Assignment 8 (Portfolio). This document will be used during the evaluation to mark your system. Should you not upload the rubric and your portfolio by the due date; 22 October 2018, then we will not be able to mark your system on the day of your evaluation. Keep a copy of the original should there be problem with the upload.

Instructions for the marker:

Marker indicate the marks obtained by the student for each component of the system on page 2. Refer to the Criteria Document that you've received for the detailed description and requirement for each component.

<i>System component</i> (Marker completes)	MARKS					
	MARKERS					
	1		2		3	
	Max	Stud	Max	Stud	Max	Stud
PREPARATION						
1. SYSTEM DESCRIPTION (Section A)	[5]					
2. Assignment 8 - Portfolio	[10]					
INPUT: GUI AND USAGE ASPECTS						
5. INPUT – USER SUPPORT	[5]					
6. INPUT – ADAPTED FOR	[5]					
SOLUTION TO A PROBLEM						
7. ALGORITHMS	[10]					
8. TECHNICAL FINISHING	[10]					
OUTPUT AND REPORTS						
9. REPORTING – DAY-TO-DAY	[10]					
10. REPORTING – MIS	[15]					
TECHNICAL DEVELOPMENT						
11. LEVEL OF PROGRAMMING APPLIED	[10]					
12. DATABASE	[10]					
13. BACKUP IMPLEMENTATION	[5]					
14. PROFESSIONAL IMPRESSION	[5]					
TOTAL:	[100]					
EXAMINER (NAME)						
EXAMINER (SIGNATURE)						
Date:						

Comments (Moderator):

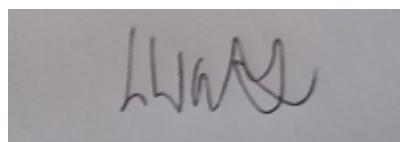
Comments (Marker):

PLAGIARISM PLEDGE BY THE STUDENT

1. I have read Unisa's plagiarism policy.
2. I understand Unisa's plagiarism policy.
3. I agree to abide by Unisa's plagiarism policy.
4. I have read the direct copying, plagiarism, and "patch-writing" document.
5. I understand what direct copying, plagiarism, and "patch-writing" is.
6. I undertake to avoid copying directly, plagiarism and patch writing.
7. All academic work, written or otherwise, that I submit is expected to be the result of my own skill and labour.
8. I understand that, if I am guilty of the infringement or breach of copyright/plagiarism or unethical practice, I will be subject to the applicable disciplinary code as determined by Unisa.
9. The examiner has the right to refuse to assess the system if plagiarism is detected.

Student name and Surname: Lee Russell Walker

Student number: 56602804



Student signature:

12 October 2018

Date:

[We will not be able to assess your final system without this plagiarism page]

SECTION A: System Information (to be completed by the student)

Title of the system:	MyVitality
Is the system a stand-alone system or network based? Explain your answer:	<p>A network-based system.</p> <p>The system needs to be connected to a network to function. The code and database are both stored and accessed via a network. Specifically, the Internet.</p>
Is the system operating system depended? Explain your answer:	<p>Operating System Independent.</p> <p>The system is accessed via a Web browser, which is operating system dependent. Meaning that if the user accesses the site from a Linux, Windows, or MAC operating system, they will have the same experience and functionality.</p>
Is your system an online or web-based system? Explain your answer and if yes, provide the URL.	<p>A Web-based system. The system is accessed via a Web browser and is dependent on Internet Web technologies.</p> <p>http://myvitality.tk – customer portal</p> <p>http://myvitality.tk/admin/ - admin portal</p>
Is your system developed as a Mobile App? Explain your answer and indicate the platforms that it is developed for.	<p>The system was not developed as a mobile application. However, the customer portal was developed with mobile users in mind. The user interface changes depending on the device you are using. This was done to give users on a mobile device the same experience as those using a laptop or PC.</p> <p>The system specification initially specified that the company operated on a Windows platform. This meant that the system must be able to run on Windows. The decision was made to make the system be able to run on multiple platforms, and the system was then designed to run on the Web platform. This gives the owner flexibility when or if they decide to change from a Windows system to a MAC or Linux based platform.</p>
Did you develop the system for a specific client? Explain	Yes. The specific client was the owner of MyVitality, who is also the HCP for the company. This is the person who hired my services for the

your answer.	design and development of the system. Note. The specific client in the first instance was the owner of AltHealth, but the instructions at the start of the project was to give the company its own name, instead of using AltHealth.
Indicate the programming language(s) that you have used in the development of the system.	HTML, CSS, PHP, SQL, JavaScript, Bootstrap (for pop up modal and icons only)
Indicate the database that you are using.	MariaDB

Anything else that we should know about your system.

The model view controller pattern was used to organise and structure the code of the application.

To access the admin side of the site, please use the following information:

Usernames	Password
caseymilan	password1 (HCP)
lienchiang	password2 (GA)
leewalker	password3 (Superuser)

User are granted different levels of access depending on their company role.

Passwords and usernames are stored in the DB using SHA1 encryption. Password were kept simple, for ease of access for system demonstration.

Requesting a new password, sends the superuser an email for them to reset and give a new password. Passwords were not automatically generated, as it could cause problems when I demonstrate the system. (For me to track the new passwords)

To access the user manuals for each system, please click the link in the site footer.

The two pictures that were used of people in the 'About Us' section, have granted their permission for it to be used.

When emails are automatically generated during the system operation, they are sent to my personal email account to show that they work.

Source code functions were documented in the following format:

```
/**  
 * @method – function name  
 * @description – a description of what the function does  
 * @param – description of the parameter  
 * @return – what the function returns  
 */
```