CHRISTIAN SERVICE UNIVERSITY COLLEGE

KUMASI – GHANA

DEPARTMENT OF COMPUTER SCIENCE

A KNOWLEDGE-BASED COSMETIC SHOP MANAGEMENT SYSTEM (DERMALOGIC SYSTEM)

CASE STUDY: KEVY'S COSMETICS PLAZA, ADUM KUMASI

A thesis presented to **The Department Of Computer Science**, Christian service University College in partial fulfillment of the requirement for the **Award of Bsc. (Hons.) Degree in Computer Science.**

Henry Amo-Mensah
Nigel Yao Jesse Mensah

Supervisor: Linda Amoako

May 2013

A KNOWLEDGE-BASED COSMETIC SHOP MANAGEMENT SYSTEM (DERMALOGIC SYSTEM)

CASE STUDY: KEVY'S COSMETICS PLAZA, ADUM KUMASI

A thesis presented to **The Department Of Computer Science,** Christian service University College in partial fulfillment of the requirement for the **Award of Bsc. (Hons.) Degree in Computer Science.**

By:
Henry Amo-Mensah
Nigel Yao Jesse Mensah

Supervisor: Linda Amoako

May 2013

DECLARATION

We hereby declare that this thesis, our original investigation and achievement, submitted for the bachelor degree at Christian service University College has not been submitted before for any degree or examination.

Henry Amo-Mensah	…….……	10th May 2013
Student	(Author's signature)	(Date)
Nigel Yao Jesse Mensah	……	10th May 2013
Student	(Author's signature)	(Date)
Linda Amoako	……..……..	10th May 2013
Supervisor	(signature)	(Date)
Linda Amoako	……..……..	10th June 2013
HoD	(signature)	(Date)

ABSTRACT

The world of technology and innovations are advancing rapidly that stakeholders are doing everything in their capacity to outdo each other whilst little or no recourse is paid to the welfare and safety of those who patronize and use these innovations. Where some efforts are made, the solutions are seemingly generic in nature and problems are not solved according to the peculiar situations of users. Dermalogic Cosmetic Shop Management system was developed to solve the problems of different categories of cosmetic products, users and administers in our society. This was achieved by the process of integration of well researched technologies - knowledge-based management system, information management system, human anatomy & physiology (dermatology), manufacturing, distribution and marketing processes and people to influence others with their knowledge. This Dermalogic system assist both shop administers and clients to eliminate the bottle-neck of making the right selection of product to suit a clients skin type and needs. The research work diligently traversed all the fundamental procedures required namely; feasibility studies, requirement analysis and design, methodologies, tools testing and implementations. The system was developed using the evolutionary prototyping model as the methodology with Microsoft Visual Studio.Net for the front-end and Microsoft SQL Server 2008 for the backend. There is the need for all stakeholders in the cosmetic industry to fully embrace this system in order to help streamline the problems posed by their range of products they often churned out to their user and various experts in the industry as result of convergence.

DEDICATION

We joyfully dedicate this project to our Lord and Saviour Jesus Christ and our ever supportive spouses. For it is said... that by strength shall no man prevail. To God be the Glory.

ACKNOWLEDGEMENT

We are very grateful to Jehovah Almighty God for the ability and grace He gave us to enable undertake this project work successfully.

We also wish to express our profound gratitude to the following people for helping us in diverse ways to accomplish this project.

We are grateful for their resourceful suggestions, directions and moral support.

- Miss Linda Amoako (HOD, Computer Science Department)
- Mr. Phanuel Asense (Lecturer, Computer Science Department)
- Mr. Albert Kweigyah (Lecturer, Computer Science Department)
- Mr. Christopher Ayaaba Abilimi (Lecturer, Computer Science Department)
- Mrs. Patience Mensah (Student, Communications Department)
- Miss. Hagar Mensah (KNUST)

Thank you all and may God bless you.

TABLE OF CONTENT

CHAPTER THREE

METHODOLOGY

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

LIST OF FIGURES

CHAPTER ONE

1.0 INTRODUCTION

Dermalogic System is a knowledge-based management system that complements the work of a Dermatologist. This system will help shop attendants select a range of cosmetic products based on customer's skin type. It has the following feature;

- Customer form
- Derma-assistant form
- Administrator form
- Supplier form
- Sales form
- Staff form
- Report's form

1.1 BACKGROUND

When it comes to creating a fresh, natural appearance, selecting the right cosmetics for your skin type is the key. (Kenda, 2012). Just as important is finding cosmetics that won't trigger allergies and result in rashing, itching or breakouts of skin. Choosing the right cosmetics for one's skin type has been the bane of many. Professional advice is sought most of the time at an extra cost, hence the need for a knowledge-based management system which can be used in the absence of Professional Beauty advisors or Cosmetologists.

The Food and Drug Administration which regulates cosmetics in the United States of America states that cosmetics are intended to be applied to the human body for cleansing, beautifying, promoting attractiveness, or altering the appearance, without affecting the body's structure or functions. (U.S. Department of Health and Human Service, 2011)

Tens of billions of dollars are spent each year by many people around the world on makeup and other cosmetic items. Advertisements promoting hair care items and makeup fill the airwaves and a lot of today's bestselling magazines are beauty related. (Michael Rupkalvis, 2007). In his article, "10 things the Beauty Industry Won't Tell You", Annamaria Andriotis, reveals that Americans spent a whopping $ 33.3billion on cosmetics and other beauty

products in 2010, up about 6% from 2009, according to the Commerce Department. (Andriotis, 2011). This shows the growing nature of the cosmetic industries.

Even though today's cosmetics business is amongst the most dominant markets, the use of makeup has truly been with us for thousands of years. From the copper and lead ore that the ancient Egyptians used to create the world's first cosmetics to the scientifically advanced products of today that can do everything from hide pores, smooth complexions, and turn the pale green of your eyes a vivid shade of emerald, makeup has been an integral part of humankind for thousands of years. Over the centuries, women used burnt matches to darken their eyes, berries to stain their lips and young boys' urine to fade their freckles. They even swallowed ox blood in some misguided attempt to improve their complexions.

Women throughout history put their health at risk with many of their homemade cosmetics. In some cultures, for example, women used arsenic, lead, mercury, and even leeches to give themselves the pale appearance deemed beautiful in the old days. Thankfully, we've come a long way from the days of using toxic and deadly mixtures to enhance our looks.

In recent times cosmetic industries must meet strict government regulations about what it can and cannot include in products and must follow safe manufacturing guidelines. Today, the most serious injury you're likely to receive from your cosmetics is an irritation or a rash from using a product that's too harsh for your skin. Despite decades of safety testing and a safety record unparalleled in many industries, there are many myths circulating about the dangers of cosmetic ingredients. (Wheatley, 2012)

1.2 THE HUMAN ANATOMY

The human body consists of complex organ with several tissues. Anatomically, it is considered according to its various parts or regions. Examples are Head, hand, thorax, pelvis etc but the various structures can also be grouped according to their common function to make up what are commonly called systems (systematic anatomy). These systems include Skeleton and muscular system, cardiovascular system, lymphatic system, respiratory system, digestive system, urinary system, reproductive system, endocrine system and nervous system and the skin. The skin, which forms part of our research, is the largest organ of the body. In cell physiology, the skin system consists of the skin, hair, nails, and various glands and it is the most visible organ of the body. The skin helps define a person's look. Any unsightly skin

appearance, such as scarring, can affect both mental and physical well-being. (Hutchings & Logan, 2003)

Cosmetics are used to enhance or protect the appearance or odor of the human body, but that is not always the case. This brings to fore the many challenges posed when it comes to taking good care of our skin in relation to using the right product. We most of the time apply certain cosmetic products on our skin out of sheer ignorance. Some also come about due to hyped media advertisement and peer influence.

Most clients do purchase cosmetic products that are not suitable for their skin type and they end up causing more harm to their skin. Today, consumers are confused as to what to believe or not, as far as cosmetic products are concerned.

The obvious solution in correcting these anomalies is the hiring up of professionals to advice clients in choosing the right cosmetics for their skin type. From our research, we found out that most of these shops cannot afford to hire the services of these Beauticians and Cosmetologists, especially for a developing country like Ghana, where the income level of most people are low.(Professor Stephen Adei, 2013).

A good solution will therefore be the setting up of a Client Knowledge-Based Management System which can be used in the absence of Professional Beauty Advisors or Cosmetologists. It aims of addressing issues relating to the choice of cosmetics for the right skin types and making recommendations when necessary.

1.3 TYPES OF MANAGEMENT INFORMATION SYSTEM

A management information system (MIS) is a computer-based system that provides the information necessary to manage an organization effectively. An MIS should be designed to enhance communication among employees, provide an objective system for recording information and support the organization's strategic goals and direction. These are;

- **Transaction-processing systems** are designed to handle a large volume of routine, recurring transactions. They were first introduced in the 1960s with the advent of mainframe computers. Transaction-processing systems are used widely today. Banks use them to record deposits and payments into accounts. Supermarkets use them to record sales and track inventory. Managers often use these systems to deal with such tasks as payroll, customer billing and payments to suppliers.

- **Operations information systems** were introduced after transaction-processing systems. An operations information system gathers comprehensive data, organizes it and summarizes it in a form that is useful for managers. These types of systems access data from a transaction-processing system and organize it into a usable form. Managers use operations information systems to obtain sales, inventory, accounting and other performance-related information.
- **Decision Support Systems (DSS)** is an interactive computer system that can be used by managers without help from computer specialists. A DSS provides managers with the necessary information to make informed decisions. A DSS has three fundamental components:
  database management system (DBMS), which stores large amounts of data relevant to problems the DSS has been designed to tackle; model-based management system (MBMS), which transforms data from the DBMS into information that is useful in decision-making; and dialog generation and management system (DGMS), which provides a user-friendly interface between the system and the managers who do not have extensive computer training.
- **Expert Systems and Artificial Intelligence** use human knowledge captured in a computer to solve problems that ordinarily need human expertise. Mimicking human expertise and intelligence requires the computer to do the following: recognize, formulate and solve a problem; explain solutions; and learn from experience. These systems explain the logic of their advice to the user; hence, in addition to solving problems they also can serve as a teacher. They use flexible thinking processes and can accommodate new knowledge.(Alfred Sarkissian, 2011).

From the above descriptions on the different types of Management Information System, Knowledge Management System can be classified as a Decision Support System. This is because it provides the necessary information to make informed decisions.

A computer cannot create knowledge, only a human being can do so through cognitive process to information. Even though Vernor Vinge, a world-renowned pioneer in Artificial Intelligence has predicted that the year computer will gain real intelligence and take over is 2020 (McKinney, January 2010). A knowledge base is a centralized repository for information: a public library, a database of related information about a particular subject, like cosmetics and skin type's database could all be considered to be examples of knowledge bases.

In summary, a Knowledge Management System is one that provides the user with explicit information required in exactly the form required, at precisely the time the user needs it. It connects to all sources of knowledge.

1.4 TYPES OF KNOWLEDGE MANAGEMENT SYSTEM

Enterprise-wide knowledge management system-they are firm-wide efforts to collect, store, distribute, and apply digital content and knowledge. They use an array of technologies for storing structured and unstructured content, locating employee expertise, searching for information, disseminating knowledge, and using data from key corporate systems. There are three major categories of enterprise-wide Knowledge Management System: Structured knowledge systems, Semi-structured knowledge systems and Knowledge networks.

The development of powerful networked workstations and software for assisting engineers and scientists in the discovery of new knowledge has led to the creation of knowledge work systems such as computer-aided design (CAD), visualization, simulation, and virtual reality systems. **Knowledge work systems** (KWS) are specialized systems built for engineers, scientists, and other knowledge workers charged with discovering and creating new knowledge for a company.

Knowledge management also includes a diverse group of **intelligent techniques**, such as data mining, expert systems, neural networks, fuzzy logic, genetic algorithms, and intelligent agents. These techniques have different objectives, from a focus on discovering knowledge (data mining and neural networks), to distilling knowledge in the form of rules for a computer program (expert systems and fuzzy logic), to discovering optimal solutions for problems-genetic algorithms. (Foster, chapter 11)

1.5 PROBLEM STATEMENT

The bane of consumers of skin products is what the professional Cosmetologist and Dermatologist sought to rectify in the various medical center, skin care center, Spa, Hospitals, retail store, service salon and many more by recommending and prescribing the various skin care products to their client from the pool of thousands of brands and makes.

One noticeable feature associated with these range of products is their lateral categorizations into various groups for treating diverse kinds of dermatitis (skin diseases).

Figure 1.0 Skinceutical Eye Complex

A mature skin remedy to dramatically diminish dark circles and puffiness.

Figure 1.1 SkinMedica Dermal Repair Cream

Advanced formula combining the highest levels of Antioxidants

Figure 1.2 StriVectin-AR Advanced Retinol Night Treatment

This midnight multi-tasker refines and smoothes skin texture without irritation.

Figures 1.1, 1.2, and 1.3 are examples of modern and complex cosmetic products, hitherto, were simple to users and in terms of ingredients combinations and functions on the human skin. These modern products apart from their basic functions now stealthily function as curative and pharmaceutical product.

These skin products irrespective of their brand-names and country of origin are widely manufactured as Cleansers, Make-up removers, Exfoliates & Mask, Tonners, Anti-Blemish, Serums, Moisturizers, Eye Care, Lip care, De-aging , Tanning etc ultimately to prevent or treat several skin conditions such as Acne, Eczema, hair disorder, birthmarks, shingles, melisma (pigment disorder), Lentigenes, psoriasis, vitiligo, sun damages and many more.

Then again, lack of adequate regulation, entrepreneurship, and commercialization has brought about a thriving feel good look industry known as Cosmeceutical (also known Skinceutical).

Thus, these consumers become very vulnerable to the adverse effect of these products instead of deriving the optimum benefit from them.

These Cosmeceutical industries focus on churning out variety of cosmetic and beauty products into the open-market. Thus, these products end up with consumers who have little or no knowledge about their skin types, and the appropriate product to use in relation to the chemical constituents.

Thus, these consumers become very vulnerable to the adverse effect of these products instead of deriving the optimum benefit from them.

It can be deduced from the above that:

- consumers of cosmetics are vulnerable when it comes to making the right choice of product based on their individual skin types.
- Most consumers find it very difficult to identify optimized usage of cosmetic product.
- consumers have difficulty accessing the services of these Cosmetologist or Dermatologist, as there are a few available.
- owners of cosmetic shops need the services of Cosmetologists to get more clients. Once these consumers are aware that a particular shop has a cosmetologist, he/she is assured of getting his/her needs met and will even inform friends.

1.6 AIM AND OBJECTIVES

The purpose of this project is to develop a Clients Knowledge-Based System.

The project aim is to develop a knowledge-based system that will assist both client and administrator to select the best skin product.

The objectives are to:

- help clients choose the right cosmetics for their skin types.
- allow information to flow to the right people at the right time.
- foster the re-use of intellectual capital.
- be a useful tool for management of cosmetic shops in their decision making process.
- eliminate the heavy reliance on the few professionals.

1.7 RESEARCH QUESTIONS

Cosmetics as defined earlier, are applied to improve the body's appearance but the questions that arise are;

- what makes a particular cosmetic product suitable for one client but not the other?
- do cosmetics have a direct relation with skin types?
- are there enough Beautician, Aestheticians, Advisors or Cosmetologists to address Client's issues?
- What is the impact of cosmetics on consumers.

- how are the various cosmetic products manufactured.
- what are the alternatives of getting professional advice to these Beauticians, Advisors and Cosmetologist?
- how can a Management Information System be incorporated into the cosmetic industry?
- what kind of Management Information System can be used?
- what technologies can be integrated in future to enhance service delivery.

Product knowledge also plays a part in influencing consumers' purchase intention. Consumers in the 21st century are well-educated in terms of their requirements of a product they intend to buy. They would search for product information to check whether the product fits their needs. Thus, firms must be aware of the importance of delivering product information efficiently, either by advertising or through adequate labeling on the products, which may require effective integrated marketing campaign. It is crucial for firms to remain competitive by keeping up with the current market trends and by conducting market research on consumers' current needs.

1.8 SIGNIFICANCE OF THE STUDY

The choice of this system is basically based on the:

- need to have a robust information location and retrieval channel to enhance individual decision making.
- embrace effective knowledge creation process.
- ensure that created knowledge is shared with and integrated among staff members

The completed software will have a repository of the various skin types and a whole range of cosmetics that will match a particular skin type based on the constituents of the cosmetics.

It will make recommendations based on the acquired knowledge. This will ultimately address arbitrarily picking and using products without recourse to their functional utility objectives as result of emerging technological convergence in the industry.

This means that information sourced will be impeccable and reliable. With this system in place, the user only needs to ask questions from the client and based on the information given, the system will process and make recommendation.

1.9 SOURCE OF DATA

Armed with these descriptions of the various stages for the Prototype model in designing a very good system, we gathered enough information from the management and clients of Kevy's Cosmetics, Angel Cream Company, Chocho Company, FC Beauty College and Komfo Anokye Dermatology Department.

The keeping of reliable data of the various skin types and their relationships to the various cosmetic ingredients will be observed. Again, a comprehensive and reliable approach will be used to ensure the sourcing of accurate information for effective processing and data storage in the database.

The management and clients of Kevy's cosmetics have been of extreme help as far as accessing information is concern. Out of the 70 clients interviewed 50 of them, representing 71% were cooperative.

We also got useful information from the internet as well as books like Database Management Systems by Patricia Ward and George Dafoulas (2006) and The Concise Handbook of Human Anatomy - R.M.H, R.T HUTCHINGS, B.M LOGAN (2003).

Another major source of our data is from several brands, make and manufacturers of cosmetics products. Labels and inscriptions on some of these products have various pieces information that guided us to comprehend their makes, brands, function, expiry dates, descriptions, products types, and the skin types manufacturer for.

1.10 SCOPE OF THE PROJECT

The project is being undertaken to design a software system that can used in the absence of Cosmetologist and Beauty Advisors. The software will be used by Kevy's Cosmetics, and retail cosmetic shop located at Adum-Pampaso.

- With staff strength of six.
- Window Shopper's per day is 27.
- Shopper per day is 49
- Shopper retainability is 15. This is a very low rate.

The low retainability is due to the fact that customers mostly do their purchases on a try and error basis. Since the clients cannot keep on with this pattern of try and error, about 60% of them never come back.

It was observed that if client's needs are met, the likelihood of retaining the client is high. Management of Kevy's cosmetics is finding it difficult employing the services of Cosmetologist as they are few in the system. This information and many other factors form the basis for the design of a Knowledge Based Management System for Kevy's Cosmetics. The project is to commerce on January 14, 2013 and end on February 14 2013.

1.11 TOOLS FOR PROJECT

- VB.NET - it is an object oriented programming language designed by Microsoft. It creates a common platform that every language could use without having to wrap it in another layer of programming interface.
- MSSQL Server- it is a Relational Database Management System (RDBMS) that runs as a server providing multi-user access to a number of databases.
- Microsoft Windows XP- it is an operating system produced by Microsoft for us on personal computers. It is user-friendly, very fast and quick response generative in nature.
- Hardware requirements needed to run the system. Pentium processor - Gigahertz and above. Hard disk - 250 Gigabytes and above Memory of 4 gigabytes or above.

1.12 SUMMARY

After a careful study of the problems under investigation and clarifying our aim and objectives, we were able to map out definite approach and strategies to develop the proposed project.

CHAPTER TWO

LITERATURE REVIEW

2.0 INRODUCTION

Dermalogic Knowledge-based Cosmetic shop Management System is a system intended bridge the seemingly gap between man, processes and technologies. It is intended to manage how cosmetic shop attendant (both trained or untrained) and clients (informed & unformed) access modern cosmetic products that are becoming increasingly complex in term of their chemical and ingredients content as against the total wellbeing, health and safety of the consumers. Therefore we set out to find out about an existing system, if any to improve upon it or develop from the scratch a new automated system to fulfill this purpose. The latter become the obvious choice when we found out that there was no such existing system as cosmetic client shop management system. We then proceeded to focus study on three thematic areas namely: people, processes and technology.

2.1 KNOWLEDGE BASED MANAGEMENT SYSTEM

The focus of this project is to use Knowledge-Based Management Information System in prescribing or recommending the right cosmetics for the right skin type. This project is a novelty; hence literature reviews on them are scanty. With regards to the literature to be reviewed, most of the pieces of information obtained were from the Internet. Notwithstanding, literature reviews on knowledge-based management system on other products are considered here. A review of cosmetics and how we can apply knowledge based management system in recommending cosmetic products for a specific skin type will also be considered.

Knowledge Management is needed by most businesses but unfortunately, there's no universal definition of knowledge management, just as there's no agreement as to what constitutes knowledge in the first place. We chose the following definition for knowledge management for its simplicity and broad context.

2.2 VARIOUS DEFINITIONS OF KNOWLEDGE MANAGEMENT SYSTEM

- Knowledge Management is a Distinct but interdependent processes of knowledge creation, knowledge storage and retrieval, knowledge transfer and knowledge application.(Alavi and Leidner, 2001)
- In the words of Madelyn Blair, Knowledge Management is the process that attempts to facilitate the full generation and flow of knowledge within an organization(Madelyn Blair, March 2008)

- "Focuses on defining the knowledge employees or systems use to perform activities and saving it in some format so that others can access it." –(Business Process Trends, volume 5, Number 4.
- "A system or framework for managing the organizational processes that create, store and distribute knowledge, as defined by its collective data, information and body of experience." (Bridgefield Group ERP/SUPPLY Chain Glossary).
- "…knowledge management is the process by which we manage human centered assets." "…the function of knowledge management is to guard and grow knowledge owned by individuals, and where possible, transfer the asset into a form where it can be more readily shared by other employees in the company." – (Annie Brooking, 1999)
- "A business process that formalizes management and leverage of a firm's intellectual assets. Knowledge Management is an enterprise discipline that promotes a collaborative and integrative approach to the creation, capture, organization, access and use of information assets, including the tacit, un-captured knowledge of people." (Business Resources Online, 2009)
- "Knowledge Management is the process through which organizations generate value from their intellectual and knowledge-based assets. Most often, generating value from such assets involves codifying what employees, partners and customers know, and sharing that information among employees, departments and even with other companies in an effort to devise best practices." (Meridith Levinaon ,2008)
- "The practice of selectively applying knowledge from previous experiences of decision making to current and future decision-making activities with the express purpose of improving the organization's effectiveness." (Jennex, 2005)
- "The distribution, access and retrieval of unstructured information about "human experiences" between interdependent individuals or among members of a workgroup. Knowledge management involves identifying a group of people who have a need to share knowledge, developing technological support that enables knowledge sharing, and creating a process for transferring and disseminating knowledge." (eLiteral, Decision Support System Glossary, March 2008).
- Knowledge Management is the explicit and systematic management of vital knowledge – and its associated processes of creation, organization, diffusion, use and exploitation – in pursuit of business objectives." (David, March 2008)

From the above definitions it can be observed that knowledge Management is a multi-disciplined approach to achieving organizational objectives by making the best use of knowledge. It focuses on processes such as acquiring, creating and sharing knowledge and the cultural and technical foundations that support them. These are:

- **People** – this deals with how to increase the ability of an individual in the organization to influence others with their knowledge
- **Processes** – Its approach varies from organization to organization. There is no limit on the number of processes
- **Technology** – It needs to be chosen only after all the requirements of a knowledge management initiative have been established.

2.3 STRUCTURE AND FUNCTIONS OF THE SKIN

Skin structure

The skin is the largest organ of the body. It has three main layers, the epidermis, the dermis and the subcutaneous layer.

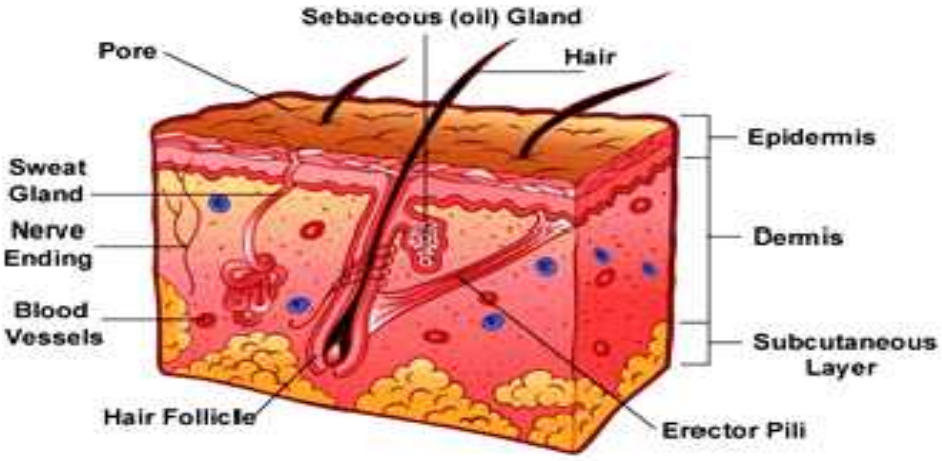
The **epidermis** is an elastic layer on the outside that is continually being regenerated. It includes the following:

- Keratinocytes - the main cells of the epidermis formed by cell division at its base. New cells continually move towards the surface. As they move they gradually die and become flattened.
- Corneocytes - the flattened dead Keratinocytes that together make up the very outer layer of the epidermis is called the stratum corneum or horny layer. This protective layer is continually worn away or shed.
- Melanocytes – produce the pigment melanin that protects against UV radiation and gives skin its colour.

The **dermis** is the inner layer that includes the following:

- Sweat glands – produce sweat that travels via sweat ducts to openings in the epidermis called pores. They play a role in temperature regulation.
- Hair follicles – are pits in which hairs grow. Hairs also play a role in temperature regulation.
- Sebaceous glands – produce sebum (oil) to keep hairs free from dust and bacteria. Sebum and sweat make up the ‘surface film’.

The **subcutaneous layer** under the dermis is made up of connective tissue and fat (a good insulator).



SOURCE-MAL'AK NATURAL (www.malaknatural.com)

Fig 2.0: Structure of the skin and its parts

The Function of the skin

The skin provides a protective barrier against mechanical, thermal and physical injury and hazardous substances. The also performs the following functions;

- Prevents loss of moisture.
- Reduces harmful effects of UV radiation.
- Acts as a sensory organ (touch, detects temperature).
- Helps regulate temperature.
- An immune organ to detect infections etc.
- Production of vitamin D

(Health and Safety Executive, 2011)

With a broad knowledge on cosmetic products and their constituents, a very good database repository will be designed to keep those data to good use.

Dermalogic Systems will be built to offer the needed platform of acquiring information on various cosmetics and their relationships with the different skin types. In this way the right cosmetic product can be recommended for all clients according to their skin types.

2.4 PROJECT EVALUATION

The project will be evaluated by answering the following questions:

- Does the system function as intended?
- Does it meet user's requirements?
- Does it meet functional and non-functional requirements?
- Does it fulfill its aims and objectives?

2.5 OVERVIEW OF THE EXISTING SYSTEM

Our visits to ten Beauty and Cosmetic shops in Kumasi revealed that:

- None of these shops make use of Knowledge Management System.
- Only a few of these even make use of Information Technology of any form.
- Only a few have a Cosmetologist or a Beautician. Mostly those Professionals are basically owners of those shops.
- Clients make purchases based on the advice of peers or hyped media advertisements.

2.6 THE PROPOSED SYSTEM

Objective is to eliminate all drawbacks of the existing system to enable effective management of the products, customer details, and staff to improve the performance and productivity by

- Designing a repository database of the various skin types and established the right relationship between these skin types and the ingredients that make up cosmetic product.
- Serving as an alternative to professional advisors like Cosmetologist.
- Designing a reliable knowledge based management system that will be incorporated into Kevy's cosmetics business setup.

2.7 SUMMARY

After extensive review of available documentations on knowledge based systems and it applications, human activities and interactions with systems in our area of study and technologies, we were able to conclude that we could actually integrate the impact of generic concept of knowledge –based technologies to bear on another systems, field of study or industry, for this matter Cosmetic industry.

Our understanding of how all the key players in the industry of our area of study interact with each other and function from the smallest process to the highest helped us to identify their individual roles, scope and limitations and to further find the best ways of optimizing their contributions through our proposed system.

CHAPTER THREE
METHODOLOGY

3.0 INTRODUCTION

To better meet our aim and objectives of developing a Client's Knowledge-Based Management System as explained in our case study, there was the need to carefully gather the needed requirements in the development of the Knowledge Management Software.

For the system to work perfectly, it is expedient to allow the users and other stakeholders to bring in their inputs for the development of the Software. Requirement gathering involves the determination of the exact requirements of the system. Elaborations of how the system is supposed to function or work as requested by the client or proposed by the designer for client's acceptance.

3.1 RESEARCH STRATEGY

The objective of our Research Strategy is to get the functional and non functional requirements for the project. We also need to better;

- Understand the perspectives of participants(Clients and shop attendants)
- Explore the meaning they give to phenomena
- Observe in-depth the processes that take place

According to Bent Flyvbjerg, (2011) "Qualitative researchers aim to gather an in-depth understanding of human behavior and the reasons that govern such behavior."

The above factors influenced our decision to use Qualitative Research Methods which allowed us to observe the working environment of Kevy's Cosmetics, site visit t had a one-on-one interview with both attendants and clients.

3.2 DATA COLLECTION METHOD

The Observations provided;

- direct information about behavior of individuals and groups
- evaluator to enter into and understand situations in context
- good opportunities for identifying unanticipated outcomes

The One-On-One Interviews

- Permitted face-to-face contact with respondents
- Provided opportunity to explore topics in depth
- Afforded us the ability to experience the effective as well as cognitive aspects of responses
- Allowed us to explain or help clarify questions, increasing the likelihood of useful responses.

Even though they afforded us the opportunity to gather our data, it was time consuming and a little bit expensive.

3.3 THE PROTOTYPING MODEL

Prototyping Model is based on the idea of developing an initial implementation, exposing this to user comment and defining this through many until an adequate system has been developed.

3.4 BENEFITS OF PROTOTYPING MODEL

The prototyping paradigm begins with requirements gathering. Developers and customers meet and define the overall objective for the software, identify the requirements and outline the areas where further definitions are necessary.

The prototype design, is often, quite different from that of the final system. The benefits of developing a prototype early in the software process are:

- Misunderstanding between software developers and users may be identified, as the functions are demonstrated.
- Missing user services may be detected.
- Difficult to use or confusing user services may be indentified and refined.
- Software development staff may find incompleteness and inconsistency in requirement as the prototype is developed.

- A working albeit limited systems is available quickly to demonstrate the feasibility and usefulness of the application to the management.

- The prototype serves as a basis for writing the specification for a production quality system. Though the principle purpose of prototyping is to validate software requirements, software prototype also has other uses.

- A prototype system can be used for training users before the formal system has been delivered.

- Prototype can be run back-to-back tests. This reduces the need for tedious manual checking of test run. The same test is given to both the prototype and the system under test to look for differences in the final results and thereby making necessary changes. Thus prototype serves as a technique of risk reduction.

3.5 SELECTING THE PROTOTYPE APPROACH

The prototype paradigm can be either close ended (throwaway prototyping) or open ended (evolutionary prototyping). Before selecting closed or open-ended approach, it is necessary to determine whether the system to be built is suitable for prototyping or not. This is decided depending on application area, complexity, and customer characteristics and projects characteristics. The throwaway is developed to understand the system requirements while the evolutionary prototype evolves through a number of versions to the final system.

Often, a customer defines a set of general objectives for software but does not identify detailed input, processing, or output requirements. In our case, we were unsure of the efficiency of an algorithm, the adaptability of an operating system, or the form that human/machine interaction should take. These situations, informed our decision to opt for a prototyping model.

It began with requirements gathering. We met with the client and defined the overall objectives for the software, identified whatever requirements are known, and outlined areas where further definition is mandatory.

We designed a representation of those aspects of the software that will be visible to the client (like input approaches and output formats). This led to the construction of a prototype.

The prototype was evaluated by the client and used to refine requirements for the software to be developed. Iteration occurs as the prototype is tuned to satisfy the needs of the customer while at the same time enabling us to better understand what needs to be done.

Ideally, the prototype serves as a mechanism for identifying software requirements.

3.6 CONTEXT LEVEL DIAGRAM

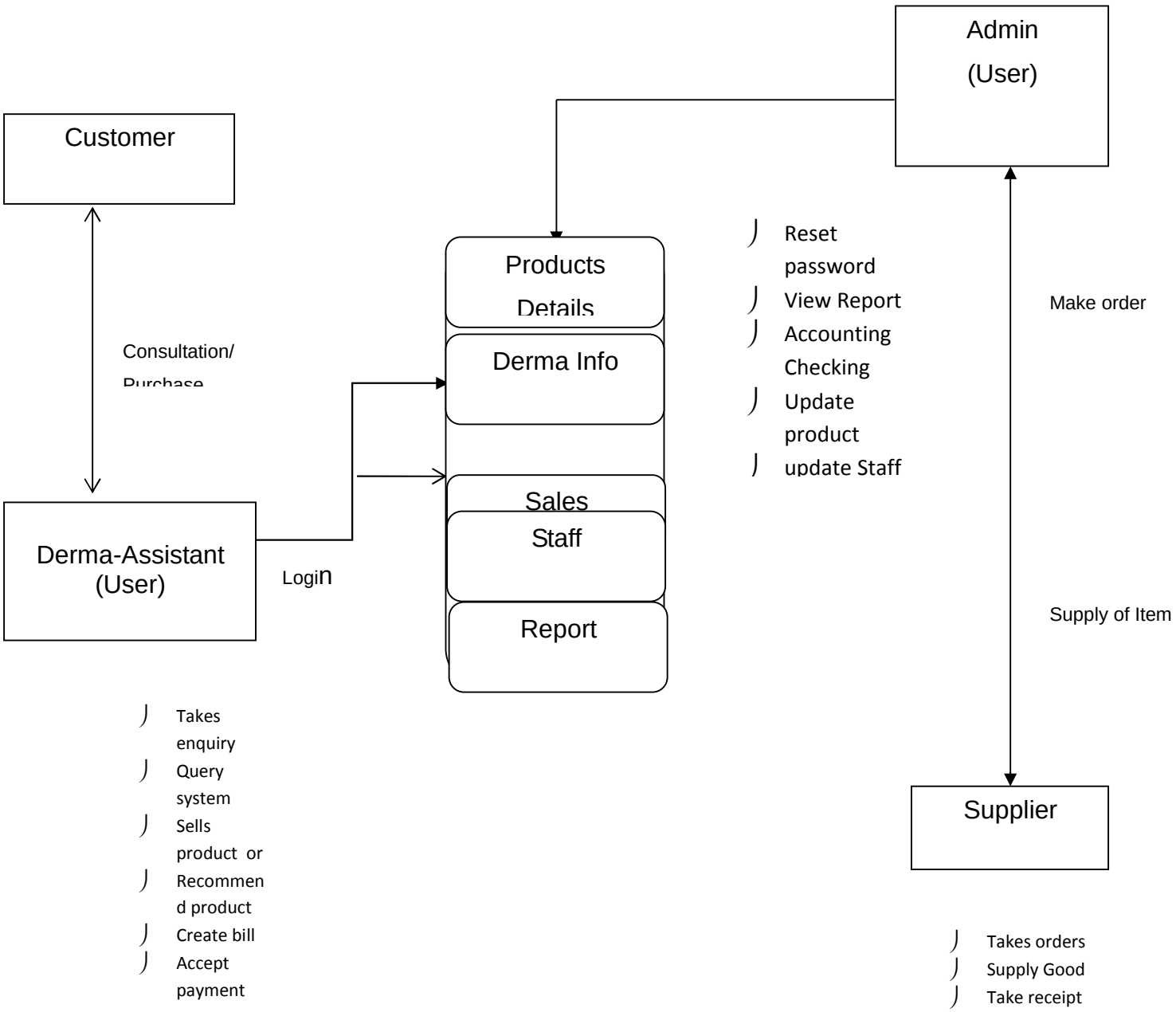


Figure 3.0 Context Level Diagram

3.7 CONTEXT LEVEL DIAGRAM (EXPLAINED)

Customer

There are two categories of customers.

- Those who purchase based on the products their skin is adapted to. They will do their normal purchases straightforward.
- Those who have no or very little knowledge about the kind of product they that will suit their skin type. They will need assistance from the Derma-Assistant.

It must be emphasized this second category of customers is our focal group the system sought to assist in choosing the right product for their skin type.

Derma-Assistant

A Derma-Assistant is the main contact person between the client and the system. The Derma-Assistant will login into the systems' DERMA INFO, and obtain some consulting-queries from the client to obtain a feedback. This Derma Info feedback from the system recommends wide ranges of product according to the client's needs to make a choice from.

The client makes a selection, collects his bill and makes payment for the product purchased. Derma-Assistant accepts payment and prints a receipt for cream collection.

The Derma-Assistant has Access right to only the Derma Info and Sales.

On the other hand, where a client's need is not met, the system allows the Derma-Assistant to collect basic skin information and contact details of the client for processing and delivery within some few week(s).

Administrator

The System Administrator has the highest Access right to the system to;

- Reset password
- View Report
- Accounting Checking
- Update product
- update Staff info

In effect he/she is administers and has full control over the entire system.

3.8 REQUIREMENTS SPECIFICATIONS

Functional requirements

The functional requirements of a system specify what the system should do or its functionalities. The functional requirements of the stock application are:

- The Dermalogic System must be able search through entire system to pick best product for the client based on his/ her need.
- The system must authenticate users before they can Login
- Dermalogic System must be able to pick the best brand of product for clients.
- Dermalogic System must alert user when a product is not in stock.
- Dermalogic System must be able to generate, view transaction reports.
- Dermalogic System must be able to validate and manage products and customers order against payments.
- The Dermalogic System must be user friendly
- The Dermalogic System must be linked to the internet..
- The Dermalogic System must classify products in categories, function, brands and types.
- Dermalogic System must be able to add/modify customer.
- Dermalogic System must store information related to products, skin types and category.
- The Dermalogic System must guarantee secure access to the stored data, managing the permissions according to the user profile.
- The Dermalogic System must support easy addition of functionalities and enhancements
- The Dermalogic System must support device upgrading or changing of devices.

Non-functional requirements

Non- functional requirements are the system's requirements that specify the properties or qualities the system must have and the constraints.

- The Dermalogic System is going to be used commercially, so the project in not open-source and the source code cannot be published
- The safety of the system information shall be insured by means of antivirus software and firewalls.
- The Dermalogic System is the first project to be developed, therefore there is no such existing system which needs to be integrated.

3.9 DATABASE DESIGN

A file format that is compliant with the combination of SQL Server 2005 Compact Edition, SQL Server 2005 and Visual Studio 2005 to build, test, deploy, and manage applications for users. Databases of this type can be accessed directly as a data source, e.g., from Visual Studio 2005, or from SQL Server Management Studio. As such, they can be passed around with programs or projects, although they are mainly used for embedded applications. The file includes all necessary code points to access the data, and so does not require an instance of SQL Server to be installed.

3.10 CONCLUSION

The above enumerated methodologies, approaches and procedures were adopted in view of the limited time at our disposal, available resources and the scope of work to be done to enable us develop the Dermalogic System that would perform all the required functions

CHAPTER FOUR

IMPLEMENTATION AND TESTING

4.0 INTRODUCTION

The detailed presentation and discussions of our project and the objectives of the study. Taking into account the thorough requirement gathering made during the system analysis and also the fact that all stakeholders were involved during the gathering of requirements, almost all the major information needed to design the software were captured, enabling a smooth transition to the other phase of the development cycle.

4.1 TESTING

Due to the Software development life cycle model used as against the various requirements gathered for developing the system, testing the system was significantly less difficult due to the fact that, the prototype model development cycle requires. Both users and developers were actively involved at every stage of the system development hence the requirements, design, and writing of code were all done repetitively until the desire software is obtained – Dermalogic system. The various tests we conducted checked the validations of the system to ascertain an assured level of security, reliability, integrity and functionality to ensure that the system met the required functionality**.** Thus, our proposed system has been meticulously tested by the various users and stakeholders.

The proposed system had been tested by the users on a standalone system and met the security level of user authentication gathered at the requirement phase of the project. This is as a result of the dynamic nature the user requirement gathering and the inclusion of all stakeholders in the project (i.e. Users and management members).

4.2 USER INTERFACE

Figure 4.0 Administrator Login Form

This form enables the System Administrator to login. It also enables the System Administrator Create user account for Users.

Figure 4.1 User Login Form

This Interface is used by users to login with their user name and password.

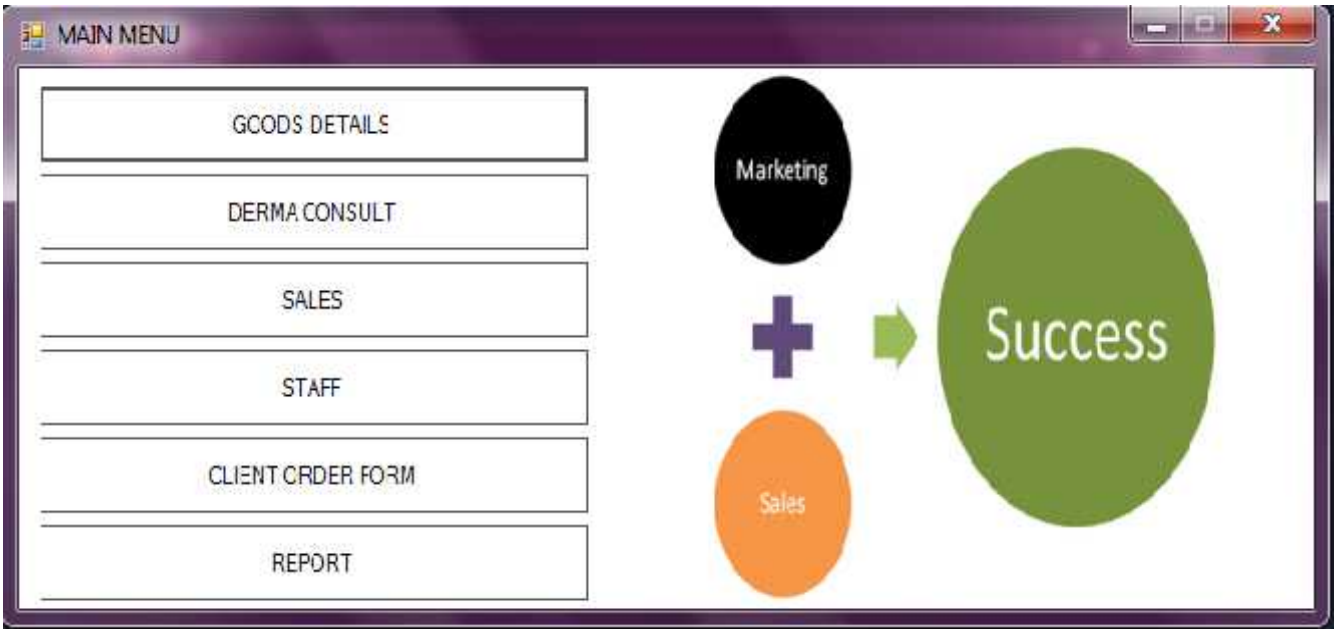


***Figure 4.2** Main MENU Form*

This is the main Menu Form that enables a user to shuffle between the Six major form. Here a user select where he/ she want to navigate to and enter with a click of a button.

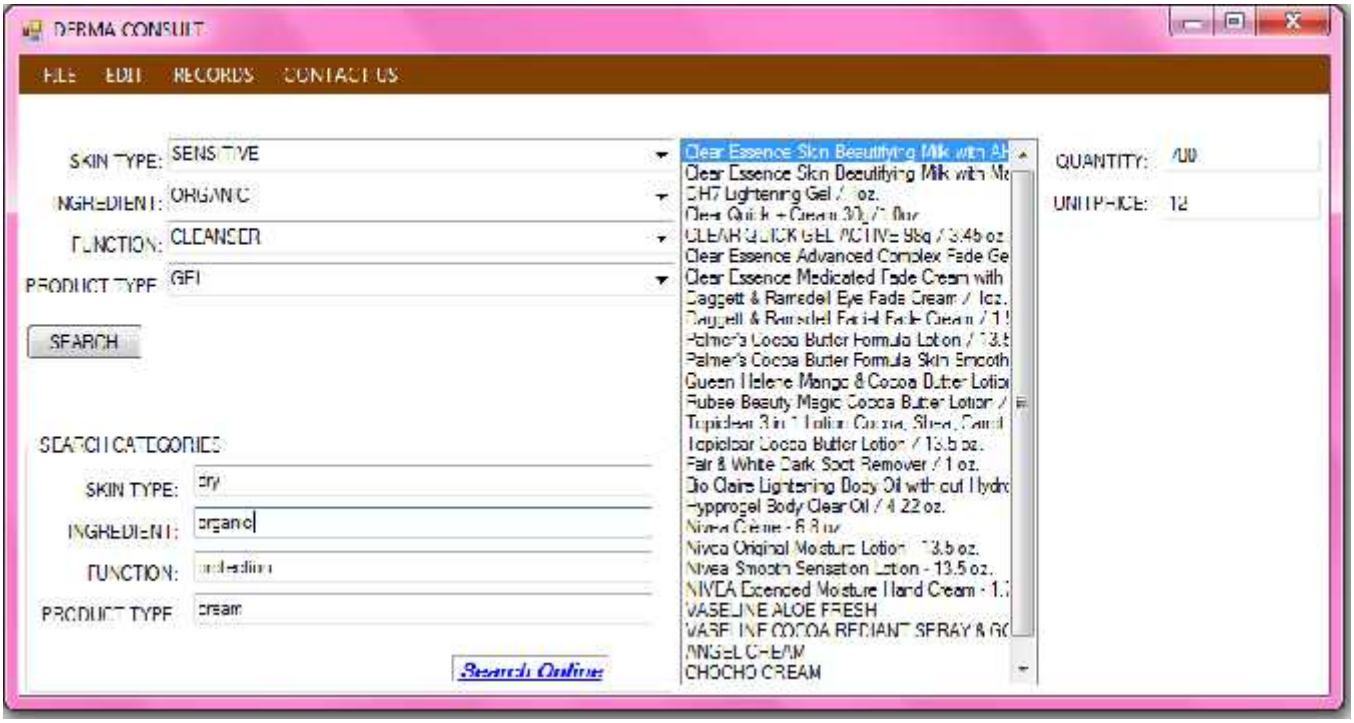


Figure 4.3 Dermal Shop Assistant Form.

Dermal shop assistant enables clients to be taking through the process of identifying the right product for the skin type.

It has two sections, one requires that the user **'select'** the corresponding items from "drop down buttons" for all the corresponding items to be displayed.

The second section enables the user to manual enter the necessary information in the skin **type, ingredient, function** and **product type** category. A simple click of a button would then recommend all the possible products for the client to purchase.

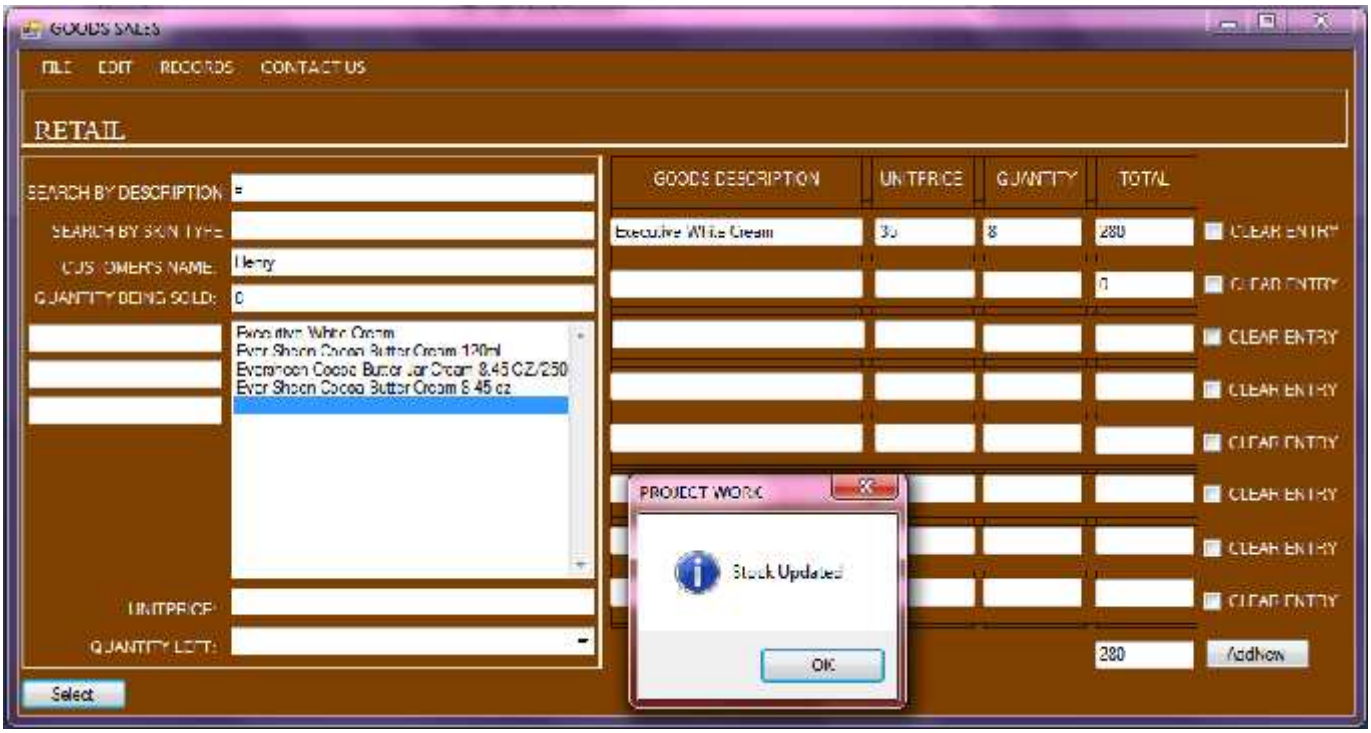


Figure 4.4 Search & Sales Form

This forms enable the system user administer the "sale" functionality of the system to the client. All product selected by client to purchase are displayed here with their corresponding prices, quantity and total cost. The system enables the user to delete an entry if the clients wishes no to proceed with an order.

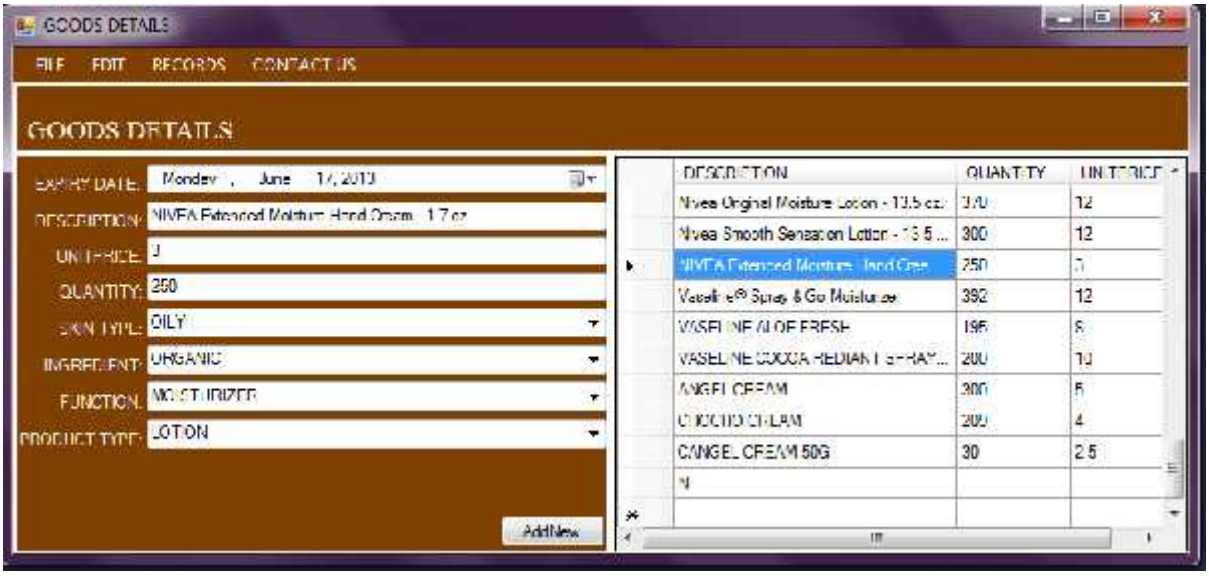


Figure 4.5 Good Details Form

Figure 4.6 Staff Data Form

Staff data form is used to capture all the necessary information about staff.

Figure 4.7 Report Form

Report menu Form is a sub-parent form that contains the four (4) types of Reports from the system.

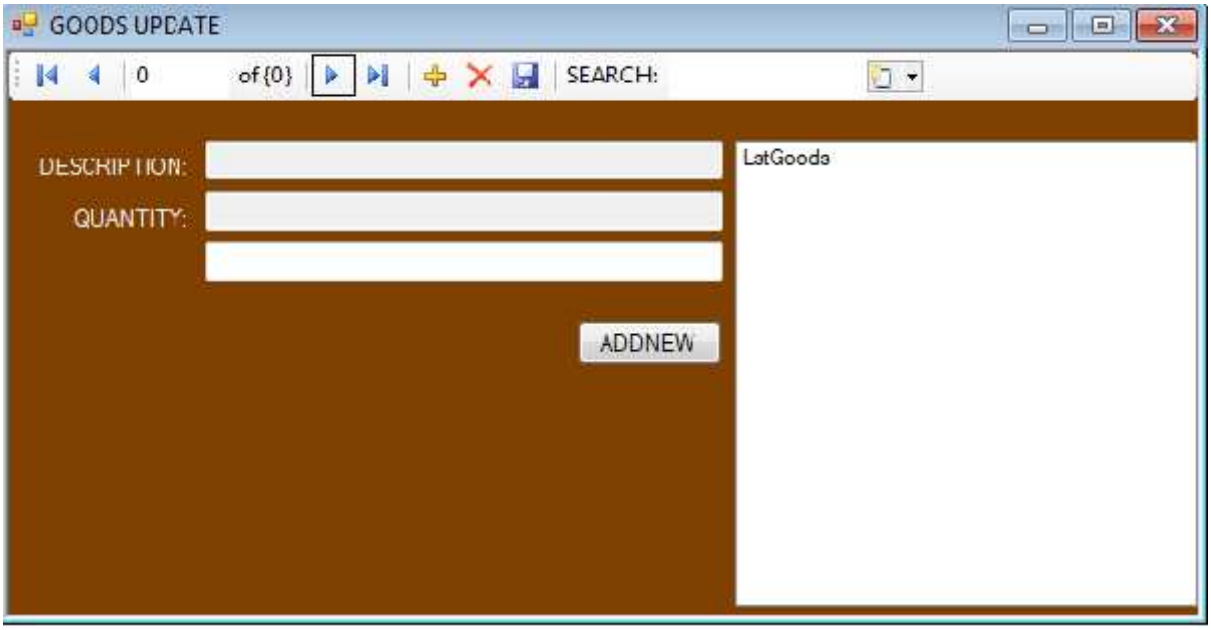


Figure 4.8 Good Update Form

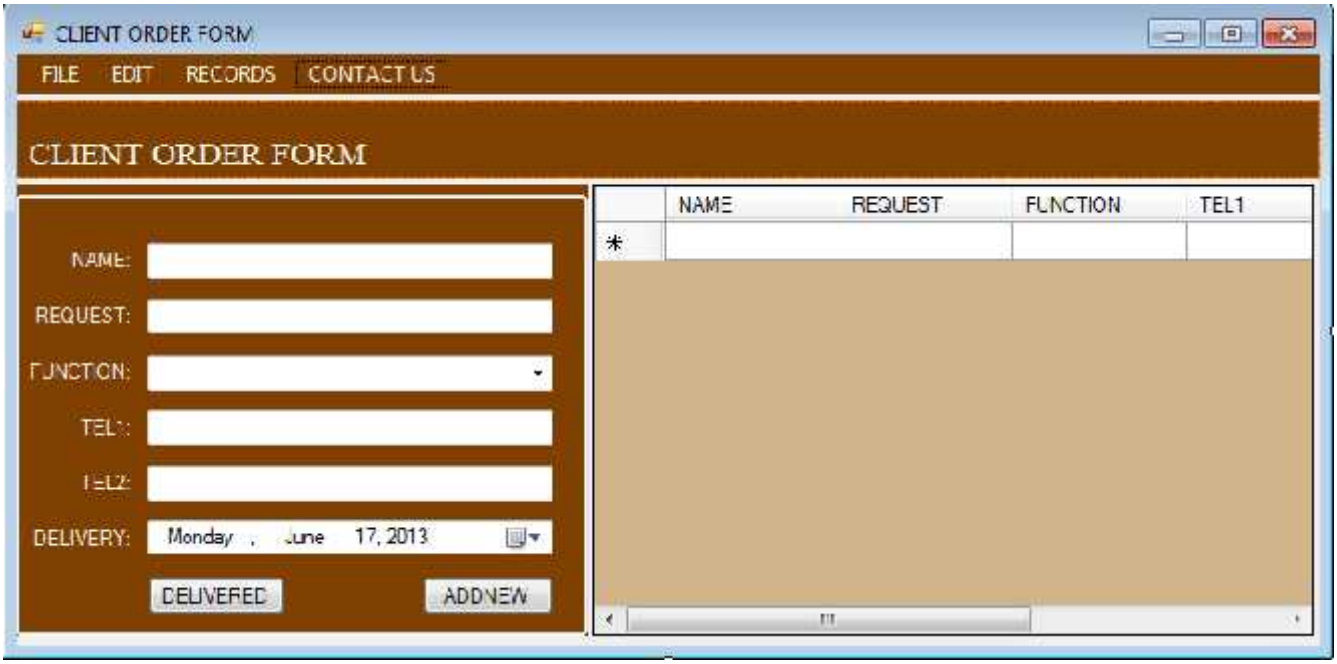


Figure 4.9 Client Order Form

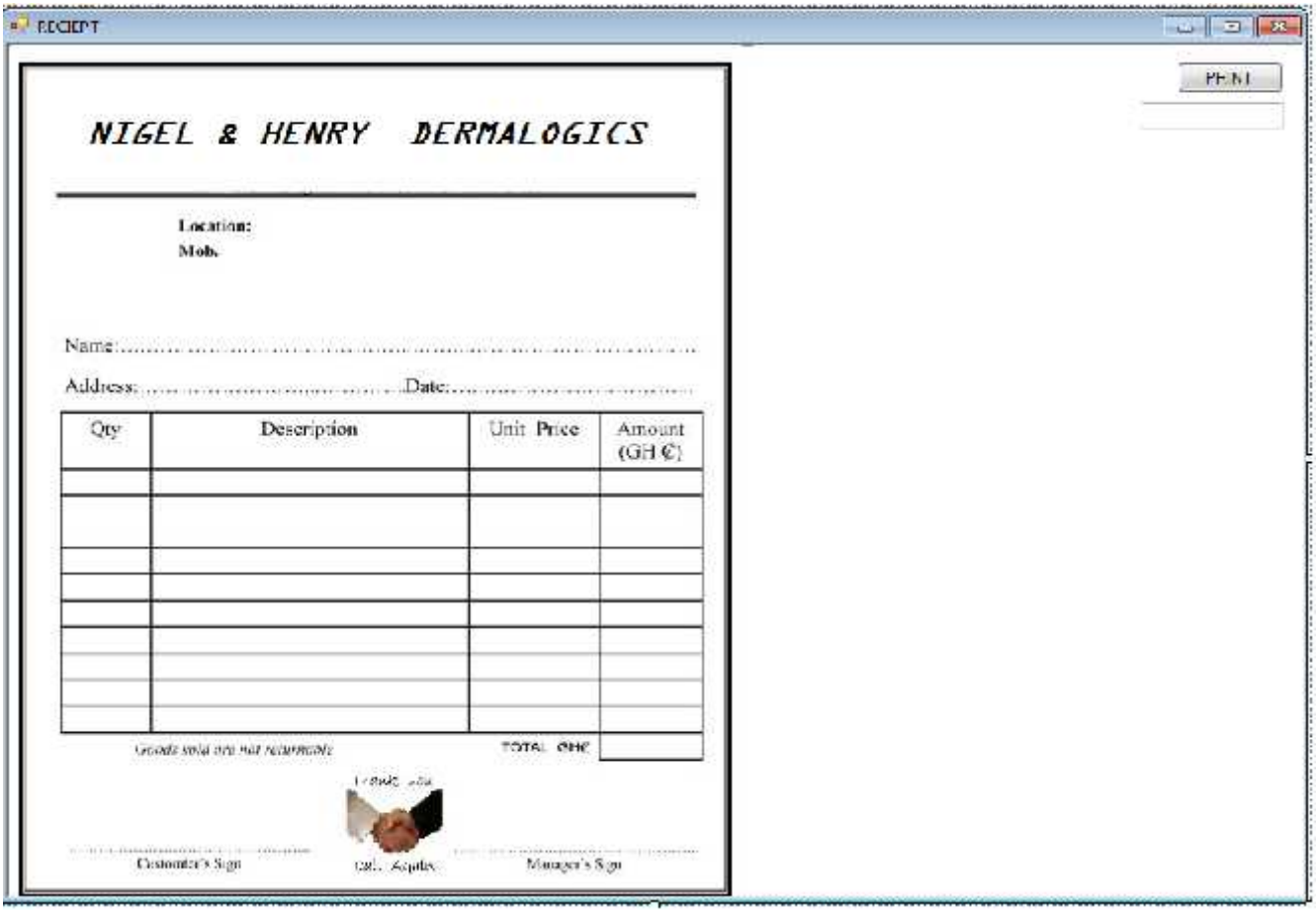


Figure 4.10 Receipt Print Form

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.0 SUMMARY

The developed system works efficiently and effectively to rectify the identify problem of technological convergence in the cosmetic industry to enable different class of cosmetic product customers to be adequately served with the right product at a click of a button.

Ultimately, this has eliminated hitherto, situation where untrained attendants administer products to their clients at their peril due to lack of adequate information on both sides.

System owners, system users and clients find the system very useful and the interface user friendly.

The proposed and finally developed system is time efficient, a repository of data, a decision making tool. It is a reliable and efficient tool for managing both cosmetic shop and clients.

The system is meant to progressively run with any manual system of administering cosmetic products. This is because the dynamics of churning out different kinds of products, brands from different manufactures and need to meet different client's requirements will not cease until a shop attendants fully masters the use and functionalities of all cosmetic products in stock at any given time.

5.1 LIMITATIONS

The successful and timely completion of the project was encumbered by time constraints, resources constraints and financial constraint. The need for timely and adequate information required that we contacted all the stakeholders in our area of study including professional and non-professionals inter-alia. The need to practical expatiate our project problem statement, ideas and concepts of our system to them was arduous because most of our resource personnel's seemed not to have noticed or identify the cause of the problem we sought to rectify and developing our project around. In spite of all these, most people still felt reluctant to give us the required information whilst others totally withheld vital information from us.

5.2 RECOMMENDATION

Culture, medicine and technology: fundamental factors that influence the fast rate of convergence in the various inter-related fields of knowledge, consequentially we infer that the cosmetic industry would continue to experience increasing degree of convergence in processes of manufacturing and churning out more and more multi-functional and advanced-

cosmetic products for the five skin-type needs of less informed users. The resultant effect of these products would thus be the need to rely solely on expert advices and dispensing akin to what current exist in the pharmaceutical industry. Hence a reliable system such as our Dermalogic System would go a long way to enhance the patronage of cosmetic products. Future changes in user requirement would necessitate changes in the system features.

We would recommend that future system or add ons, integrate web-based electronic commerce (e-commerce) and mobile technologies to enhance transcend our original scope and functionalities so to make the system more accessible to all difference classes and category of users and stakeholders in the cosmetic industry.

REFERENCES

10 Things The Beauty Industry Won't Tell You- Annamaria Andriotis, 2011

Articles On Cosmetics-Through-The-Ages- Michael Rupkalvis, 2007

Astd Learning System Module-8

Brian Henderson-Sellers And Julian M. Edwards," The Object-Oriented Systems Life Cycle

Business Resources Online,2009

Cosmetics Side Effects And Allergies On Human Skin

David Skyrme Associates, Web Page Viewed 13 March 2008)

Definition Of Knowledge Base - Margaret Rouse, 2007

Http://Blog.Simslearningconnections.Com

Http://Www.Hse.Gov.Uk/Skin/Professional/Causes/Structure.Htm

Http://Www.Scribd.Com/Doc/17822460/Cosmetics-Side-Effects-And-Allergies-On-Human-Skin

http://Www.Unc.Edu/~Sunnyliu/Inls258/Introduction_To_Knowledge_Management.Html

Jennex, Knowledge Management In Modern Organizations, 2005, P.4

Knowledge Management And Knowledge Management System: A Conceptual Foundation And Research Issues -Alavi And Leidner, 2001

Madelyn Blair, Pelerei Inc.

Managing Information Systems-Chapter 11, Foster School Of Business, University Of Washington

Patricia Ward And George Dafoulas (2006) In Their Book Entitled "Database Management Systems" Pp. 3

Purchasing Cosmetic Products: A Preliminary Perspective Of Gen-Y. Pp. 7

Roger Pressman, Software Engineering: A Practitioners Approach, Mcgraw Hill, 2002

Strategies For Knowledge Management- Annie Brooking, 1999

The Concise Handbook Of Human Anatomy - R.M.H, R.T Hutchings, B.M Logan – Book Power-2003, Isbn 1-84076-042-7

The Daily Galaxy Front Page Story - Luke Mckinney, 30 January 2010.

Types Of Management Information System- Alfred Sarkissian

U.S. Department Of Health And Human Service, 30/11/ 2012

U.S. Department Of Human And Human Service, 2011

COSMETIC SHOP KNOWLEDGE-BASED MANAGEMENT SYSTEM

SOURCE CODE

1.0 MAIN MENU FORM

//CREATING THE MAIN MENU

```
Public Class FrmMainMenu

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        FrmGoodsDetails.Show()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        FrmSalesOfGoods.Show()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        FrmDermaConsult.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        FrmStaffDetails.Show()
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        FrmClientOrderForm.Show()
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click
        Form1.Show()
    End Sub

    Private Sub FrmMainMenu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    End Sub
End Class
```

1.1 REPORT MENU

//CODES TO DISPLAY THE MENU REPORT

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        FrmStockValue.Show()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        FrmSalesRecords.Show()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Form2.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        Form3.Show()
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

    End Sub
End Class
```

1.2 STAFF RECORDS

// CREATING THE STAFF RECORD FORM

```
Public Class Form2

    Private Sub DETAILSBindingNavigatorSaveItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DETAILSBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.DETAILSBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.STAFFDataSet)

    End Sub
```

```
    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'STAFFDataSet.DETAILS' table. You can move, or remove it, as needed.
        Me.DETAILSTableAdapter.Fill(Me.STAFFDataSet.DETAILS)

    End Sub



    Private Sub ToolStripTextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles ToolStripTextBox1.KeyPress
        Me.DETAILSBindingSource.Filter = "NAME'" & ToolStripTextBox1.Text & "%'"
    End Sub

    Private Sub ToolStripTextBox1_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles ToolStripTextBox1.TextChanged
        Me.DETAILSBindingSource.Filter = "NAME'" & ToolStripTextBox1.Text & "%'"
    End Sub

    Private Sub ToolStripTextBox1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripTextBox1.Click

    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
        Me.PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        Me.PrintForm1.Print()
    End Sub
End Class
```

1.3 CLIENT ORDER REPORT

//CREATING THE CLIENT ORDER REPORT

```
    Private Sub DETAILSBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DETAILSBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.DETAILSBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.ORDERDataSet)

    End Sub

    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        'TODO: This line of code loads data into the 'ORDERDataSet.DETAILS' table. You can move, or remove it, as needed.
        Me.DETAILSTableAdapter.Fill(Me.ORDERDataSet.DETAILS)

    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
        Me.PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        Me.PrintForm1.Print()
    End Sub
End Class
```

1.4 ADMINISTRATOR AND USER FORM

//CREATING THE ADMINSTRATOR FORM

```
Public Class Form4

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        LoginForm2.Show()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        If TextBox1.Text = "ADMIN" Then
            LoginForm1.Show()
            TextBox1.Text = ""
        Else
            MsgBox("Access Denied!", MsgBoxStyle.Critical)
        End If
    End Sub

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub
End Class
```

1.5 CLIENT ORDER FORM

//CODES TO CREATE THE CLIENT ORDER FORM

```
Public Class FrmClientOrderForm

    Private Sub DETAILSBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Validate()
        Me.DETAILSBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.ORDERDataSet)

    End Sub

    Private Sub Form11_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'ORDERDataSet.DETAILS' table. You can move, or remove it, as needed.
        Me.DETAILSTableAdapter.Fill(Me.ORDERDataSet.DETAILS)

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Me.DETAILSBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.ORDERDataSet)

    End Sub



    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

        Me.DETAILSBindingSource.RemoveCurrent()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.ORDERDataSet)

    End Sub

    Private Sub LinkLabel1_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs)

    End Sub



    Private Sub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles GroupBox1.Enter

    End Sub
```

```
End Class
```

1.6 DERMACONSULT FORM

//CODES TO CREATE THE DERMA CONSULT FORM

```
Public Class FrmDermaConsult

    Private Sub PRODUCTBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PRODUCTBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PRODUCTBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)

    End Sub

    Private Sub Form7_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
        STATUSComboBox.Text = "SELECT"
        STATUS1ComboBox.Text = "SELECT"
        STATUS2ComboBox.Text = "SELECT"
        STATUS3ComboBox.Text = "SELECT"
    End Sub
    Private Sub STATUSComboBox_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles STATUSComboBox.SelectedIndexChanged
        If STATUSComboBox.Text = "NORMAL" Then
            TextBox1.Text = "N"
        End If
        If STATUSComboBox.Text = "DRY" Then
            TextBox1.Text = "D"
        End If
        If STATUSComboBox.Text = "OILY" Then
            TextBox1.Text = "O"
        End If
        If STATUSComboBox.Text = "SENSITIVE" Then
            TextBox1.Text = "S"
        End If
        If STATUSComboBox.Text = "COMBINATION" Then
            TextBox1.Text = "C"
        End If
    End Sub
```

```
    Private Sub STATUS1ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS1ComboBox.SelectedIndexChanged
        If STATUS1ComboBox.Text = "ORGANIC" Then
            TextBox2.Text = "0"
        End If
        If STATUS1ComboBox.Text = "INORGANIC" Then
            TextBox2.Text = "I"
        End If
    End Sub

    Private Sub STATUS2ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS2ComboBox.SelectedIndexChanged
        If STATUS2ComboBox.Text = "CLEANSER" Then
            TextBox3.Text = "C"
        End If
        If STATUS2ComboBox.Text = "EYE CARE" Then
            TextBox3.Text = "E"
        End If
        If STATUS2ComboBox.Text = "MASKING" Then
            TextBox3.Text = "M"
        End If
        If STATUS2ComboBox.Text = "TONERS" Then
            TextBox3.Text = "T"
        End If
        If STATUS2ComboBox.Text = "PROTECTION" Then
            TextBox3.Text = "P"
        End If
        If STATUS2ComboBox.Text = "EXFOLIATORS" Then
            TextBox3.Text = "E"
        End If
        If STATUS2ComboBox.Text = "MOISTURIZER" Then
            TextBox3.Text = "M"
        End If
    End Sub

    Private Sub STATUS3ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS3ComboBox.SelectedIndexChanged
        If STATUS3ComboBox.Text = "LOTION" Then
            TextBox4.Text = "L"
        End If
        If STATUS3ComboBox.Text = "GEL" Then
            TextBox4.Text = "G"
        End If
        If STATUS3ComboBox.Text = "SERUM" Then
            TextBox4.Text = "SE"
        End If
        If STATUS3ComboBox.Text = "CLAY" Then
            TextBox4.Text = "CL"
```

```
        End If
        If STATUS3ComboBox.Text = "CREAM" Then
            TextBox4.Text = "CR"
        End If
        If STATUS3ComboBox.Text = "FOAM" Then
            TextBox4.Text = "F"
        End If
        If STATUS3ComboBox.Text = "SCRUB" Then
            TextBox4.Text = "SC"
        End If
        If STATUS3ComboBox.Text = "SPRAY" Then
            TextBox4.Text = "SP"
        End If
    End Sub

    Private Sub TextBox5_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS4 LIKE '" & TextBox5.Text & "%'"
    End Sub


    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS4 LIKE '" & TextBox5.Text & "%'"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        TextBox5.Text = TextBox1.Text & TextBox2.Text & TextBox3.Text & TextBox4.Text
    End Sub

    Private Sub RELOADToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RELOADToolStripMenuItem.Click
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
    End Sub

    Private Sub EXITToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EXITToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub TextBox6_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS LIKE '" & TextBox6.Text & "%'"
    End Sub
```

```
    Private Sub TextBox6_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox6.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS LIKE '" & TextBox6.Text & "%'"
    End Sub

    Private Sub TextBox7_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox7.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS1 LIKE '" & TextBox7.Text & "%'"
    End Sub

    Private Sub TextBox7_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox7.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS1 LIKE '" & TextBox7.Text & "%'"
    End Sub

    Private Sub TextBox8_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox8.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS2 LIKE '" & TextBox8.Text & "%'"
    End Sub

    Private Sub TextBox8_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox8.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS2 LIKE '" & TextBox8.Text & "%'"
    End Sub

    Private Sub TextBox9_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox9.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS3 LIKE '" & TextBox9.Text & "%'"
    End Sub

    Private Sub TextBox9_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox9.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS3 LIKE '" & TextBox9.Text & "%'"
    End Sub


    Private Sub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox1.Enter

    End Sub
End Class
```

1.7 GOODS DETAILS FORM

//CODES TO CREATE GOOD'S DETAILS

```
Public Class FrmGoodsDetails

    Private Sub PRODUCTBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PRODUCTBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PRODUCTBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)

    End Sub

    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)

        PRODUCTBindingSource.AddNew()
    End Sub
    Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" & TextBox2.Text & "%'"
    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" & TextBox2.Text & "%'"
    End Sub

    Private Sub QUANTITYTextBox_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles QUANTITYTextBox.KeyPress
        If e.KeyChar = Chr(13) Then
            Button1_Click(Me, EventArgs.Empty)
        End If
    End Sub

    Private Sub QUANTITYTextBox_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles QUANTITYTextBox.TextChanged
        VALUETextBox.Text = Val(UNITPRICETextBox.Text) * Val(QUANTITYTextBox.Text)
    End Sub

    Private Sub UNITPRICETextBox_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UNITPRICETextBox.TextChanged
        VALUETextBox.Text = Val(UNITPRICETextBox.Text) * Val(QUANTITYTextBox.Text)
```

```
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        STATUS4TextBox.Text = TextBox1.Text & TextBox5.Text & TextBox3.Text &
TextBox4.Text
        TextBox1.Text = ""
        TextBox5.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        PRODUCTBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)

        MsgBox("Entry Saved", MsgBoxStyle.Information)

    End Sub

    Private Sub RELOADToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RELOADToolStripMenuItem.Click
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
    End Sub

    Private Sub SEARCHToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SEARCHToolStripMenuItem.Click
        Label2.Visible = True
        TextBox2.Visible = True
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
        PRODUCTBindingNavigator.Visible = True
    End Sub

    Private Sub SAVEToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SAVEToolStripMenuItem1.Click
        PRODUCTBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)
        MsgBox("Entry Saved", MsgBoxStyle.Information)
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
    End Sub

    Private Sub EXITToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles EXITToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub SAVEToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SAVEToolStripMenuItem.Click
        PRODUCTBindingSource.AddNew()
```

```
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)
        MsgBox("Entry Saved", MsgBoxStyle.Information)
    End Sub

    Private Sub STOCKVALUEToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STOCKVALUEToolStripMenuItem.Click
        FrmStockValue.SHOW()
    End Sub

    Private Sub STATUSComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUSComboBox.SelectedIndexChanged
        If STATUSComboBox.Text = "NORMAL" Then
            TextBox1.Text = "N"
        End If
        If STATUSComboBox.Text = "DRY" Then
            TextBox1.Text = "D"
        End If
        If STATUSComboBox.Text = "OILY" Then
            TextBox1.Text = "O"
        End If
        If STATUSComboBox.Text = "SENSITIVE" Then
            TextBox1.Text = "S"
        End If
        If STATUSComboBox.Text = "COMBINATION" Then
            TextBox1.Text = "C"
        End If
    End Sub


    Private Sub STATUS1ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS1ComboBox.SelectedIndexChanged
        If STATUS1ComboBox.Text = "ORGANIC" Then
            TextBox5.Text = "O"
        End If
        If STATUS1ComboBox.Text = "INORGANIC" Then
            TextBox5.Text = "I"
        End If
    End Sub

    Private Sub STATUS2ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS2ComboBox.SelectedIndexChanged
        If STATUS2ComboBox.Text = "CLEANSER" Then
            TextBox3.Text = "C"
        End If
        If STATUS2ComboBox.Text = "EYE CARE" Then
            TextBox3.Text = "E"
        End If
```

```
        If STATUS2ComboBox.Text = "MASKING" Then
            TextBox3.Text = "M"
        End If
        If STATUS2ComboBox.Text = "TONERS" Then
            TextBox3.Text = "T"
        End If
        If STATUS2ComboBox.Text = "PROTECTION" Then
            TextBox3.Text = "P"
        End If
        If STATUS2ComboBox.Text = "EXFOLIATORS" Then
            TextBox3.Text = "E"
        End If
        If STATUS2ComboBox.Text = "MOISTURIZER" Then
            TextBox3.Text = "M"
        End If
    End Sub

    Private Sub STATUS3ComboBox_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STATUS3ComboBox.SelectedIndexChanged
        If STATUS3ComboBox.Text = "LOTION" Then
            TextBox4.Text = "L"
        End If
        If STATUS3ComboBox.Text = "GEL" Then
            TextBox4.Text = "G"
        End If
        If STATUS3ComboBox.Text = "SERUM" Then
            TextBox4.Text = "SE"
        End If
        If STATUS3ComboBox.Text = "CLAY" Then
            TextBox4.Text = "CL"
        End If
        If STATUS3ComboBox.Text = "CREAM" Then
            TextBox4.Text = "CR"
        End If
        If STATUS3ComboBox.Text = "FOAM" Then
            TextBox4.Text = "F"
        End If
        If STATUS3ComboBox.Text = "SCRUB" Then
            TextBox4.Text = "SC"
        End If
        If STATUS3ComboBox.Text = "SPRAY" Then
            TextBox4.Text = "SP"
        End If
    End Sub
End Class
```

1.8 GROUP UPDATES

//CODES TO CREATE UPDATES

```
Public Class FrmGroupUpDates

    Private Sub PRODUCTBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PRODUCTBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PRODUCTBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)

    End Sub

    Private Sub Form6_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)

    End Sub

    Private Sub QUANTITYTextBox_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles QUANTITYTextBox.TextChanged
        VALUETextBox.Text = Val(UNITPRICETextBox.Text) * Val(QUANTITYTextBox.Text)
    End Sub

    Private Sub UNITPRICETextBox_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UNITPRICETextBox.TextChanged
        VALUETextBox.Text = Val(UNITPRICETextBox.Text) * Val(QUANTITYTextBox.Text)
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
        If Val(TextBox1.Text) < 1 Then
            MsgBox("Invalid Entry", MsgBoxStyle.Information)
            TextBox1.Text = ""
        End If
        QUANTITYTextBox.Text = Val(TextBox1.Text) + Val(ComboBox1.Text)
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        QUANTITYTextBox.Text = Val(TextBox1.Text) + Val(ComboBox1.Text)
    End Sub

    Private Sub ToolStripTextBox1_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles ToolStripTextBox1.KeyPress
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" & ToolStripTextBox1.Text & "%'"
    End Sub
```

```
    Private Sub ToolStripTextBox1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ToolStripTextBox1.TextChanged
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" &
ToolStripTextBox1.Text & "%'"
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        PRODUCTBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
        MsgBox("Entry Saved", MsgBoxStyle.Information)

        FrmSalesOfGoods.Enabled = True

    End Sub
End Class
```

1.9 RECEIPT FORM

//CODES TO CREATE THE RECEIPT FORM

```
Public Class FrmReceipt

    Private Sub INCOMEBindingNavigatorSaveItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles INCOMEBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.INCOMEBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.SALEDataSet)

    End Sub

    Private Sub Form8_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        'TODO: This line of code loads data into the 'SALEDataSet.INCOME' table. You can
move, or remove it, as needed.
        Me.INCOMETableAdapter.Fill(Me.SALEDataSet.INCOME)

    End Sub
```

```
    Private Sub UNITPRICE7Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE7Label1.TextChanged
        UNITPRICE7Label1.Text = Format(Val(UNITPRICE7Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL1Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL1Label1.TextChanged
        TOTAL1Label1.Text = Format(Val(TOTAL1Label1.Text), "fixed")
    End Sub


    Private Sub TOTALLabel1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTALLabel1.TextChanged
        TOTALLabel1.Text = Format(Val(TOTALLabel1.Text), "fixed")
    End Sub

    Private Sub TOTAL2Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL2Label1.TextChanged
        TOTAL2Label1.Text = Format(Val(TOTAL2Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL3Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL3Label1.TextChanged
        TOTAL3Label1.Text = Format(Val(TOTAL3Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL4Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL4Label1.TextChanged
        TOTAL4Label1.Text = Format(Val(TOTAL4Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL5Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL5Label1.TextChanged
        TOTAL5Label1.Text = Format(Val(TOTAL5Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL6Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL6Label1.TextChanged
        TOTAL6Label1.Text = Format(Val(TOTAL6Label1.Text), "fixed")
    End Sub

    Private Sub TOTAL7Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL7Label1.TextChanged
        TOTAL7Label1.Text = Format(Val(TOTAL7Label1.Text), "fixed")
    End Sub
```

```
    Private Sub TOTAL8Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TOTAL8Label1.TextChanged
        TOTAL8Label1.Text = Format(Val(TOTAL8Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICE1Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE1Label1.TextChanged
        UNITPRICE1Label1.Text = Format(Val(UNITPRICE1Label1.Text), "fixed")
    End Sub


    Private Sub UNITPRICE2Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE2Label1.TextChanged
        UNITPRICE2Label1.Text = Format(Val(UNITPRICE2Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICE3Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE3Label1.TextChanged
        UNITPRICE3Label1.Text = Format(Val(UNITPRICE3Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICE4Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE4Label1.TextChanged
        UNITPRICE4Label1.Text = Format(Val(UNITPRICE4Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICE5Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE5Label1.TextChanged
        UNITPRICE5Label1.Text = Format(Val(UNITPRICE5Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICE6Label1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICE6Label1.TextChanged
        UNITPRICE6Label1.Text = Format(Val(UNITPRICE6Label1.Text), "fixed")
    End Sub

    Private Sub UNITPRICELabel1_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles UNITPRICELabel1.TextChanged
        UNITPRICELabel1.Text = Format(Val(UNITPRICELabel1.Text), "fixed")
    End Sub

    Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
        'Me.INCOMEBindingSource.Filter = " GOOD12 LIKE '" & TextBox1.Text & "%'"
        'INCOMEBindingSource.MoveLast()
    End Sub
```

```
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged
        'Me.INCOMEBindingSource.Filter = " GOOD12 LIKE '" & TextBox1.Text & "%'"
        'INCOMEBindingSource.MoveLast()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Me.PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        Me.PrintForm1.Print()
    End Sub
End Class
```

1.10 SALES FORM

//CREATING THE SALES FORM

```
Public Class FrmSalesOfGoods

    Private Sub INCOMEBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles INCOMEBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.INCOMEBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.SALEDataSet)

    End Sub

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
        'TODO: This line of code loads data into the 'SALEDataSet.INCOME' table. You can move, or remove it, as needed.
        Me.INCOMETableAdapter.Fill(Me.SALEDataSet.INCOME)
        Me.DATETextBox.Text = TimeOfDay.ToLongTimeString
        QTY12TextBox.Text = Format(Now, "short Date")
        INCOMEBindingSource.AddNew()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If Val(TextBox2.Text) < 1 Then
```

```
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            QTY12TextBox.Text = Format(Now, "short Date")
            GOODSTextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICETextBox.Text = UNITPRICETextBox1.Text
            QUANTITYTextBox.Text = TextBox2.Text


            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = True
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False

    End Sub


    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        TextBox6.Text = GOODS12TextBox.Text

        Me.INCOMEBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.SALEDataSet)
        MsgBox("Entry Saved", MsgBoxStyle.Information)
        FrmReceipt.Show()
        FrmReceipt.TextBox1.Text = Me.TextBox6.Text
        Button1.Visible = True
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
```

```
        Button25.Visible = False
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox5.Checked = False
        CheckBox6.Checked = False
        CheckBox7.Checked = False
        CheckBox8.Checked = False
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        QUANTITYTextBox1.Text = Val(ComboBox1.Text) - Val(TextBox2.Text)
    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox2.TextChanged

        QUANTITYTextBox1.Text = Val(ComboBox1.Text) - Val(TextBox2.Text)
    End Sub
    Private Sub TextBox1_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" & TextBox1.Text & "%'"
    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox1.TextChanged
        Me.PRODUCTBindingSource.Filter = " DESCRIPTION LIKE '" & TextBox1.Text & "%'"
        TextBox2.Enabled = True
    End Sub



    Private Sub TextBox2_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress
        If e.KeyChar = Chr(13) Then
            Button1_Click(Me, EventArgs.Empty)
        End If
    End Sub


    Private Sub STOCKVALUEToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles STOCKVALUEToolStripMenuItem.Click
        FrmSalesRecords.Show()
    End Sub
```

```
    Private Sub RELOADToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RELOADToolStripMenuItem.Click
        Me.INCOMETableAdapter.Fill(Me.SALEDataSet.INCOME)
    End Sub

    Private Sub EXITToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles EXITToolStripMenuItem.Click
        Me.Close()
    End Sub

    Private Sub SAVEToolStripMenuItem1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SAVEToolStripMenuItem1.Click
        Me.INCOMEBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.SALEDataSet)
        MsgBox("Entry Saved", MsgBoxStyle.Information)
        Me.INCOMETableAdapter.Fill(Me.SALEDataSet.INCOME)
    End Sub

    Private Sub SAVEToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SAVEToolStripMenuItem.Click
        Me.INCOMEBindingSource.AddNew()
    End Sub


    Private Sub UNITPRICETextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles UNITPRICETextBox.TextChanged
        TOTAL1TextBox.Text = Val(QUANTITYTextBox.Text) *
Val(UNITPRICETextBox.Text)
    End Sub

    Private Sub QUANTITYTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QUANTITYTextBox.TextChanged
        TOTAL1TextBox.Text = Val(QUANTITYTextBox.Text) *
Val(UNITPRICETextBox.Text)
    End Sub

    Private Sub UNITPRICE1TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE1TextBox.TextChanged
        TOTAL2TextBox.Text = Val(QTY1TextBox.Text) * Val(UNITPRICE1TextBox.Text)
    End Sub

    Private Sub QTY1TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY1TextBox.TextChanged
        TOTAL2TextBox.Text = Val(QTY1TextBox.Text) * Val(UNITPRICE1TextBox.Text)
    End Sub
```

```
    Private Sub QTY7TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY7TextBox.TextChanged
        TOTAL8TextBox.Text = Val(QTY7TextBox.Text) * Val(UNITPRICE7TextBox.Text)
    End Sub

    Private Sub UNITPRICE7TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE7TextBox.TextChanged
        TOTAL8TextBox.Text = Val(QTY7TextBox.Text) * Val(UNITPRICE7TextBox.Text)
    End Sub

    Private Sub QTY6TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY6TextBox.TextChanged
        TOTAL7TextBox.Text = Val(QTY6TextBox.Text) * Val(UNITPRICE6TextBox.Text)
    End Sub

    Private Sub UNITPRICE6TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE6TextBox.TextChanged
        TOTAL7TextBox.Text = Val(QTY6TextBox.Text) * Val(UNITPRICE6TextBox.Text)
    End Sub

    Private Sub QTY5TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY5TextBox.TextChanged
        TOTAL6TextBox.Text = Val(QTY5TextBox.Text) * Val(UNITPRICE5TextBox.Text)
    End Sub

    Private Sub UNITPRICE5TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE5TextBox.TextChanged
        TOTAL6TextBox.Text = Val(QTY5TextBox.Text) * Val(UNITPRICE5TextBox.Text)
    End Sub

    Private Sub QTY4TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY4TextBox.TextChanged
        TOTAL5TextBox.Text = Val(QTY4TextBox.Text) * Val(UNITPRICE4TextBox.Text)
    End Sub

    Private Sub UNITPRICE4TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE4TextBox.TextChanged
        TOTAL5TextBox.Text = Val(QTY4TextBox.Text) * Val(UNITPRICE4TextBox.Text)
    End Sub

    Private Sub QTY3TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY3TextBox.TextChanged
        TOTAL4TextBox.Text = Val(QTY3TextBox.Text) * Val(UNITPRICE3TextBox.Text)
    End Sub

    Private Sub UNITPRICE3TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE3TextBox.TextChanged
        TOTAL4TextBox.Text = Val(QTY3TextBox.Text) * Val(UNITPRICE3TextBox.Text)
```

```
    End Sub

    Private Sub QTY2TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles QTY2TextBox.TextChanged
        TOTAL3TextBox.Text = Val(QTY2TextBox.Text) * Val(UNITPRICE2TextBox.Text)
    End Sub

    Private Sub UNITPRICE2TextBox_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles UNITPRICE2TextBox.TextChanged
        TOTAL3TextBox.Text = Val(QTY2TextBox.Text) * Val(UNITPRICE2TextBox.Text)
    End Sub

    Private Sub TOTAL1TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL1TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL7TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL7TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL6TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL6TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL5TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL5TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL4TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL4TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub
```

```
    Private Sub TOTAL3TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL3TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL2TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL2TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub TOTAL8TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTAL8TextBox.TextChanged
        TOTALTextBox.Text = Val(TOTAL1TextBox.Text) + Val(TOTAL2TextBox.Text) +
Val(TOTAL3TextBox.Text) + Val(TOTAL4TextBox.Text) + Val(TOTAL5TextBox.Text) +
Val(TOTAL6TextBox.Text) + Val(TOTAL7TextBox.Text) + Val(TOTAL8TextBox.Text)
    End Sub

    Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button19.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            QTY12TextBox.Text = Format(Now, "short Date")
            GOODS1TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE1TextBox.Text = UNITPRICETextBox1.Text
            QTY1TextBox.Text = TextBox2.Text


            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = True
        Button21.Visible = False
        Button22.Visible = False
```

```
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False
    End Sub

    Private Sub Button20_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button20.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            QTY12TextBox.Text = Format(Now, "short Date")
            GOODS2TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE2TextBox.Text = UNITPRICETextBox1.Text
            QTY2TextBox.Text = TextBox2.Text


            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = True
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False
    End Sub

    Private Sub Button25_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button25.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            QTY12TextBox.Text = Format(Now, "short Date")
            GOODS7TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE7TextBox.Text = UNITPRICETextBox1.Text
            QTY7TextBox.Text = TextBox2.Text
```

```
            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
        Button25.Enabled = False
    End Sub

    Private Sub Button24_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button24.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            GOODS6TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE6TextBox.Text = UNITPRICETextBox1.Text
            QTY6TextBox.Text = TextBox2.Text
            QTY12TextBox.Text = Format(Now, "short Date")

            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
```

```
        Button24.Visible = False
        Button25.Visible = True
    End Sub

    Private Sub Button23_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button23.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            GOODS5TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE5TextBox.Text = UNITPRICETextBox1.Text
            QTY5TextBox.Text = TextBox2.Text
            QTY12TextBox.Text = Format(Now, "short Date")

            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = True
        Button25.Visible = False
    End Sub

    Private Sub Button21_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button21.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            GOODS3TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE3TextBox.Text = UNITPRICETextBox1.Text
            QTY3TextBox.Text = TextBox2.Text
            QTY12TextBox.Text = Format(Now, "short Date")

            PRODUCTBindingSource.AddNew()
            Me.Validate()
```

```
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = True
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False
    End Sub

    Private Sub Button22_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button22.Click
        If Val(TextBox2.Text) < 1 Then
            MsgBox("Enter the number being sold", MsgBoxStyle.Information)
        Else
            Me.DATETextBox.Text = Format(Now, "Long Date")
            GOODS4TextBox.Text = DESCRIPTIONTextBox.Text
            UNITPRICE4TextBox.Text = UNITPRICETextBox1.Text
            QTY4TextBox.Text = TextBox2.Text
            QTY12TextBox.Text = Format(Now, "short Date")

            PRODUCTBindingSource.AddNew()
            Me.Validate()
            Me.TableAdapterManager1.UpdateAll(Me.RETAILDataSet)
            MsgBox("Stock Updated", MsgBoxStyle.Information)
            Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)
            TextBox2.Enabled = False
        End If
        TextBox2.Text = ""
        TextBox1.Text = ""
        TextBox1.Select()
        Button1.Visible = False
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = True
        Button24.Visible = False
        Button25.Visible = False
    End Sub
```

```
    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            GOODSTextBox.Text = ""
            UNITPRICETextBox.Text = ""
            QUANTITYTextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
            Button25.Visible = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
            CheckBox4.Checked = False
            CheckBox5.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then
            GOODS1TextBox.Text = ""
            UNITPRICE1TextBox.Text = ""
            QTY1TextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
            Button25.Visible = False
            CheckBox1.Checked = False
            CheckBox3.Checked = False
            CheckBox4.Checked = False
            CheckBox5.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
```

```
        FrmGroupUpDates.Show()
        Me.Enabled = False
    End If
End Sub

Private Sub CheckBox7_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox7.CheckedChanged
    If CheckBox7.Checked = True Then
        GOODS6TextBox.Text = ""
        UNITPRICE6TextBox.Text = ""
        QTY6TextBox.Text = ""
        Button1.Visible = True
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
        CheckBox4.Checked = False
        CheckBox5.Checked = False
        CheckBox6.Checked = False
        CheckBox8.Checked = False
        FrmGroupUpDates.Show()
        Me.Enabled = False
    End If
End Sub

Private Sub CheckBox6_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox6.CheckedChanged
    If CheckBox6.Checked = True Then
        GOODS5TextBox.Text = ""
        UNITPRICE5TextBox.Text = ""
        QTY5TextBox.Text = ""
        Button1.Visible = True
        Button19.Visible = False
        Button20.Visible = False
        Button21.Visible = False
        Button22.Visible = False
        Button23.Visible = False
        Button24.Visible = False
        Button25.Visible = False
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
```

```
            CheckBox4.Checked = False
            CheckBox5.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub CheckBox5_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox5.CheckedChanged
        If CheckBox5.Checked = True Then
            GOODS4TextBox.Text = ""
            UNITPRICE4TextBox.Text = ""
            QTY4TextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
            Button25.Visible = False
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
            CheckBox4.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub CheckBox4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox4.CheckedChanged
        If CheckBox4.Checked = True Then
            GOODS3TextBox.Text = ""
            UNITPRICE3TextBox.Text = ""
            QTY3TextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
```

```
            Button25.Visible = False
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
            CheckBox5.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CheckBox3.CheckedChanged
        If CheckBox3.Checked = True Then
            GOODS2TextBox.Text = ""
            UNITPRICE2TextBox.Text = ""
            QTY2TextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
            Button25.Visible = False
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox4.Checked = False
            CheckBox5.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            CheckBox8.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub CheckBox8_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CheckBox8.CheckedChanged
        If CheckBox8.Checked = True Then
            GOODS7TextBox.Text = ""
            UNITPRICE7TextBox.Text = ""
            QTY7TextBox.Text = ""
            Button1.Visible = True
            Button19.Visible = False
            Button20.Visible = False
```

```
            Button21.Visible = False
            Button22.Visible = False
            Button23.Visible = False
            Button24.Visible = False
            Button25.Visible = False
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
            CheckBox4.Checked = False
            CheckBox5.Checked = False
            CheckBox6.Checked = False
            CheckBox7.Checked = False
            FrmGroupUpDates.Show()
            Me.Enabled = False
        End If
    End Sub

    Private Sub RECIEPTToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles RECIEPTToolStripMenuItem.Click
        FrmReceipt.Show()
    End Sub

    Private Sub TextBox5_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles TextBox5.KeyPress
        Me.PRODUCTBindingSource.Filter = " STATUS LIKE '" & TextBox5.Text & "%'"
    End Sub

    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox5.TextChanged
        Me.PRODUCTBindingSource.Filter = " STATUS LIKE '" & TextBox5.Text & "%'"
    End Sub

    Private Sub GOODS12TextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GOODS12TextBox.TextChanged

    End Sub

    Private Sub TOTALTextBox_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TOTALTextBox.TextChanged

    End Sub
End Class
```

1.11 SALES RECORDS

//CREATING THE SALES RECORD FORM

```
Public Class FrmSalesRecords

    Private Sub INCOMEBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles INCOMEBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.INCOMEBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.SALEDataSet)

    End Sub

    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'SALEDataSet.INCOME' table. You can move, or remove it, as needed.
        Me.INCOMETableAdapter.Fill(Me.SALEDataSet.INCOME)

    End Sub

    Private Sub FillByToolStripButton_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FillByToolStripButton.Click
        Try
            Me.INCOMETableAdapter.FillBy(Me.SALEDataSet.INCOME, New System.Nullable(Of Date)(CType(BeginningDateToolStripTextBox.Text, Date)), New System.Nullable(Of Date)(CType(EndingDateToolStripTextBox.Text, Date)))
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try

    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click
        BeginningDateToolStripTextBox.Text = DateTimePicker1.Value
        EndingDateToolStripTextBox.Text = DateTimePicker2.Value
        Try
            Me.INCOMETableAdapter.FillBy(Me.SALEDataSet.INCOME, New System.Nullable(Of Date)(CType(BeginningDateToolStripTextBox.Text, Date)), New System.Nullable(Of Date)(CType(EndingDateToolStripTextBox.Text, Date)))
        Catch ex As System.Exception
            System.Windows.Forms.MessageBox.Show(ex.Message)
        End Try
        Dim total As String = 0

        For i As Integer = 0 To INCOMEDataGridView.RowCount - 1

            total += INCOMEDataGridView.Rows(i).Cells(4).Value

        Next
```

```
        Label2.Text = total
    End Sub

    Private Sub TextBox6_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox6.KeyPress
        Me.INCOMEBindingSource.Filter = " QTY12 LIKE '" & TextBox6.Text & "%'"
    End Sub

    Private Sub TextBox6_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged
        Me.INCOMEBindingSource.Filter = " QTY12 LIKE '" & TextBox6.Text & "%'"
    End Sub

    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            TextBox6.Text = Format(Now, "short Date")
        End If
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
        Me.PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        Me.PrintForm1.Print()
    End Sub
End Class
```

1.12 STAFF DETAILS FORM

//CREATING THE STAFF DETAILS

```
Public Class FrmStaffDetails

    Private Sub DETAILSBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Me.Validate()
        Me.DETAILSBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.STAFFDataSet)

    End Sub

    Private Sub Form9_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'STAFFDataSet.DETAILS' table. You can move, or remove it, as needed.
        Me.DETAILSTableAdapter.Fill(Me.STAFFDataSet.DETAILS)

    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Me.DETAILSTableAdapter.Fill(Me.STAFFDataSet.DETAILS)
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Me.DETAILSBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.STAFFDataSet)

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button4.Click
        Form10.Show()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Try
            Dim fopen As New OpenFileDialog
            fopen.FileName = ""
            fopen.Filter = "Image Files (*.jpg)|*.jpg|(*.jpeg)|*.JPEG|(*.gif)|*.gif|(*.png)|*.png|All
Files (*.*)|*.*"
            fopen.ShowDialog()
            PICTUREPictureBox.Image = System.Drawing.Bitmap.FromFile(fopen.FileName)

        Catch ex As Exception

        End Try
    End Sub
```

1.13 STOCK VALUE FORM

//CREATING THE STOCK VALUE FORM

```
Public Class FrmStockValue

    Private Sub PRODUCTBindingNavigatorSaveItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles PRODUCTBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.PRODUCTBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.RETAILDataSet)

    End Sub
```

```
    Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'RETAILDataSet.PRODUCT' table. You can move, or remove it, as needed.
        Me.PRODUCTTableAdapter.Fill(Me.RETAILDataSet.PRODUCT)

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim total As String = 0

        For i As Integer = 0 To PRODUCTDataGridView.RowCount - 1

            total += PRODUCTDataGridView.Rows(i).Cells(3).Value

        Next

        Label2.Text = total
    End Sub

    Private Sub ToolStripButton1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click
        Me.PrintForm1.PrintAction = Printing.PrintAction.PrintToPreview
        Me.PrintForm1.Print()
    End Sub
End Class
```

1.14 LOGIN FORM

//CODES TO CREATE THE LOGIN FORM

```
    Private Sub LOGINBindingNavigatorSaveItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LOGINBindingNavigatorSaveItem.Click
        Me.Validate()
        Me.LOGINBindingSource.EndEdit()
        Me.TableAdapterManager.UpdateAll(Me.LOGINSDataSet)

    End Sub
```

```
    Private Sub LoginForm1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'TODO: This line of code loads data into the 'LOGINSDataSet.LOGIN' table. You can
move, or remove it, as needed.
        Me.LOGINTableAdapter.Fill(Me.LOGINSDataSet.LOGIN)
        LOGINBindingSource.AddNew()
    End Sub
    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox2.TextChanged
        If TextBox2.Text <> "" Then
            TextBox3.Text = MD5CalcString(TextBox2.Text)
        Else
        End If
    End Sub


    Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Cancel.Click
        USERNAMETextBox.Text = TextBox1.Text
        If RadioButton1.Checked = True Then
            LOGINTYPETextBox.Text = "ADMINISTRATOR"
        ElseIf RadioButton2.Checked = True Then
            LOGINTYPETextBox.Text = "USERS"


        End If

        USERNAMETextBox.Text = TextBox1.Text

        Try
            If My.Computer.FileSystem.FileExists("C:\log\" + TextBox1.Text + "\" + "LogInfo.txt")
= False Then
                My.Computer.FileSystem.CreateDirectory("C:\log\" + TextBox1.Text + "\")
                Dim path As String = "C:\log\" + TextBox1.Text + "\" + "LogInfo.txt"
                ' Create or overwrite the file.
                Dim fs As FileStream = File.Create(path)
                fs.Close()
            End If
        Catch ex As Exception
        End Try
        Try
            Dim Username As String = My.Computer.FileSystem.ReadAllText("C:\log\" +
TextBox1.Text + "\" + "LogInfo.txt")
            Dim Password As String = My.Computer.FileSystem.ReadAllText("C:\log\" +
TextBox1.Text + "\" + "LogInfo.txt")

            My.Computer.FileSystem.WriteAllText("C:\log\" + TextBox1.Text + "\" + "LogInfo.txt",
TextBox1.Text & ":" & TextBox3.Text, True)
```

```
            TextBox1.Text = ""
            TextBox2.Text = ""
            TextBox3.Text = ""
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        MsgBox("Account Craeted!", MsgBoxStyle.Information)



        LOGINBindingSource.AddNew()
        Me.Validate()
        Me.TableAdapterManager.UpdateAll(Me.LOGINSDataSet)

        Cancel.Enabled = False
        RadioButton2.Enabled = True
        RadioButton1.Enabled = True
    End Sub

    Private Sub OK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles OK.Click
        If TextBox1.Text = "" Then
            MsgBox("Please Enter Your UserName!", MsgBoxStyle.Critical)
        ElseIf TextBox1.Text = "" Then
            MsgBox("Please Enter Your PassWord!", MsgBoxStyle.Critical)
        Else
            Try
                Dim Username As String = My.Computer.FileSystem.ReadAllText("C:\log\" +
TextBox1.Text + "\" + "LogInfo.txt")
                Dim Password As String = My.Computer.FileSystem.ReadAllText("C:\log\" +
TextBox1.Text + "\" + "LogInfo.txt")

                Dim UseUsername As String = Username.Split(":").GetValue(0)
                Dim UsePassword As String = Username.Split(":").GetValue(1)

                If TextBox1.Text = Username.Split(":").GetValue(0) And TextBox3.Text =
Password.Split(":").GetValue(1) Then
                    MsgBox("Welcome!", MsgBoxStyle.Information)
                    Me.Hide()


                    FrmMainMenu.Show()

                Else
                    MsgBox("Username/PassWord Incorrect!", MsgBoxStyle.Critical)
                    TextBox1.Text = ""
                    TextBox2.Text = ""
```

```
                TextBox3.Text = ""
            End If
        Catch ex As Exception
        End Try
    End If

End Sub
Private Sub RadioButton1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton1.CheckedChanged
    If RadioButton1.Enabled = True Or RadioButton2.Enabled = True Then
        Cancel.Enabled = True
    Else
        Cancel.Enabled = False
    End If
End Sub

Private Sub RadioButton2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioButton2.CheckedChanged
    If RadioButton1.Enabled = True Or RadioButton2.Enabled = True Then
        Cancel.Enabled = True
    Else
        Cancel.Enabled = False
    End If
End Sub



End Class
```