`

LOCAL SHOPPING ZONE WITH VOICE RECOGNITION

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING

Submitted by

AKULA SUPRIYA
Roll No.16551A0503

R. VARSHINI
Roll No.16551A0591

G. NARESH
Roll No.16551A0564

Y. MANIKANTA
Roll No.16551A05B2

Under the Supervision of
Dr. J. M. S. V. RAVI KUMAR
Professor
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM, AP

Jawaharlal Nehru Technological University, Kakinada, AP, India

APRIL 2020

GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY(A)

CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296, AP

BONAFIDE CERTIFICATE

Certified that this project report **"Local Shopping Zone with voice Recognition "** is the Bonafide work of "**A.SUPRIYA (16551A0503), R.VARSHINI(16551A0591), G.NARESH (16551A0564),** and **Y.MANIKANTA(16551A05B2)"**,who carried out the project work under my supervision during the year 2019 to 2020, towards partial fulfilment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering as administered under the Regulations of Godavari Institute of Engineering & Technology, Rajamahendravaram, AP, India and award of the Degree from Jawaharlal Nehru Technological University, Kakinada. The results embodied in this report have not been submitted to any other University for the award of any degree.

Signature of the Head of the Department

DR. B. SUJATHA
HEAD OF THE DEPARTMENT
Department of Computer Science & Engineering.

Signature of the Supervisor

JMSV Ravikumar

Dr. J. M. S. V. RAVIKUMAR
SUPERVISOR
Professor,
Department of Computer Science & Engineering,
Godavari Institute of Engineering & Technology.

Date:

External Viva voice conducted on _____

Internal Examiner **External Examiner**

GODAVARI INSTITUTE OF ENGINEERING & TECHNOLOGY (A)
CHAITANYA KNOWLEDGE CITY, NH-16, RAJAMAHENDRAVARAM 533296, AP

CERTIFICATE OF AUTHENTICATION

We solemnly declare that this project report **"Local Shopping Zone with voice Recognition"** is the bonafide work done purely by us, carried out under the supervision of **Dr. J. M. S. V. RAVI KUMAR**, towards partial fulfilment of the requirements of the Degree of Bachelor of Technology in Computer Science & Engineering as administered under the Regulations of Godavari Institute of Engineering & Technology, Rajamahendravaram, AP, India and award of the Degree from Jawaharlal Nehru Technological University, Kakinada during the year 2019- 2020.

We also declare that no part of this document has been taken up verbatim from any source without permission from the author(s)/publisher(s). Wherever few sentences, findings, images, diagrams or any other piece of information has been used for the sake of completion of this work, we have adequately referred to the document source. In the event of any issue arising hereafter about this work, we shall be personally responsible.

It is further certified that this work has not been submitted, either in part of in full, to any other department of the Jawaharlal Nehru Technological University Kakinada, or any other University, institution or elsewhere, in India or abroad or for publication in any form.

Date: Signature of the Student(s)

AKULA SUPRIYA	**(16551A0503)**
R. VARSHINI	**(16551A0591)**
G. NARESH	**(16551A0564)**
Y. MANI KANTA	**(16551A05B2)**

ACKNOWLEDGEMENT

We feel immense pleasure to express our sincere thanks and profound sense of gratitude to all those people who played a valuable role for the successful completion of our project.

We would like to express our gratitude to **Dr. K.V.V. SATYANARAYANA RAJU**, chairman, Chaitanya group of Institutions, **Sri. K. SASI KIRAN VARMA**, Vice Chairman, GIET group of Institutions for providing necessary infrastructure.

We are thankful to Principal **Dr. P. M. M. S. SHARMA** for permitting and encouraging us in doing this project. Our special thanks to **Dr. N. LEELAVATHY**, Vice Principal (Academics), for her content guidance and motivation and also for providing an excellent environment. We are truly grateful for their valuable suggestions and advice.

We are proudly grateful to express our deep sense of gratitude and respect towards

Dr. B. SUJATHA, professor and Head of the Department, Computer Science and Engineering, whose motivation and constant encouragement has led to pursue a project in the field of software development. We are very fortunate, for having her to enlighten us in all the aspects of life and transforming us to be an ideal individual in this field.

We are much obliged and thankful to our internal guide and project coordinator **Dr. J. M. S. V. RAVI KUMAR,** professor, Department of CSE, for providing this opportunity and constant encouragement given by him during the course. We are grateful to his valuable guidance and suggestions during my project work.

Our parents have put us ahead of themselves, because of their hard work and dedication, we have had opportunities beyond our wildest dreams. Our heart full thanks to them for giving us all support we needed to be successful student and individual.

Finally, we express our thanks to other professors, classmates, friends, neighbours and non-teaching staff who helped us for the completion of our project and without infinite love and patience this would never have been possible.

ABSTRACT

Local Shopping Zone creates an ecosystem for the customers to interact and integrate with offline stores, which displays products in offline stores that are available to the customers. The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the products from the Local stores. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet. Thus the customer will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality.Aimed to connect all local store keepers ranging in all the categories.

Digitalization has sown its seed in the current world. With the growth of this digital plant, it has become difficult for a large section of people to adapt to this change. Many of them find it tedious to sustain in this world. Offline store owners are one among them. Keeping this problem in view, we as IT students have come up with a solution. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, since the application is easily accessible and always available.

Our platform displays the products in offline stores that are available to the customers. The customers may visit the local online stores to buy them. This would help the offline store owners to sustain and also provide the customers with better prices. This project is a desktop application based on shopping system for an existing local shop. The project objective is to deliver the online shopping application into desktop platform.

TABLE OF CONTENTS

CHAPTER N.O	TITLE	PAGE NO.
	ABSTARCT	**v**
1	**INTRODUCTION**	1
2	**LITERATURE SURVEY**	4
3	**SYSTEM ANALYSIS**	5
	3.1 Existing System	5
	3.2 Proposed System	6
4	**SYSTEM STUDY**	8
	4.1 Feasibility Study	8
	4.1.1 Technical Feasibility	9
	4.1.2 Economical Feasibility	9
	4.1.3 Operational Feasibility	9
	4.1.4 Social Feasibility	9
5	**SYSTEM DESIGN**	10
	5.1 Input Design	10
	5.1.1 System Architecture	11
	5.1.2 Output Design	12
	5.1.3 Block Diagram	14
	5.2 Flow Chart	15
	5.3 UML Diagrams	16
	5.3.1 Use Case Diagram	16
	5.3.2 Sequence Diagram	17
	5.3.3 Activity Diagram	18
6	**SYSTEM IMPLEMENTATION**	19
	6.1 Modules	27

CHAPTER NO.	TITLE	PAGE NO.
7	**SYSTEM REQUIREMENTS**	25
	7.1 Software Requirements	25
8	**SYSTEM TESTING**	28
	8.1 Types Of Testing	28
9	**RESULTS**	34
10	**CONCLUSION & FUTURE SCOPE**	41
	APPENDIX	42
	SOURCE CODE	73
	REFERENCES	74

LIST OF FIGURES

S.NO.	Figure.No.	NAME OF THE FIGURE	Page. No.
1	5.1.1	System Architecture	11
2	5.1.2(a)	Login Table	12
3	5.1.2(b)	User Table	12
4	5.1.2(c)	Procut Table	13
5	5.1.2(d)	Product Orders Table	13
6	5.1.3	Block Diagram	14
7	5.2	Class Diagram	15
8	5.3.1	Usecase Diagram	16
9	5.3.2	Sequence Diagram	17
10	5.3.3	Activity Diagram	18
11	6.1	Admin Module	19
12	6.2	Manage Moderator	20
13	6.3	Manage Products	21
14	6.4	Manage User	22
15	6.5	Manage Orders	23
16	6.6	Moderator	23
17	6.7	User Module	24
18	9.1	Sign Up Page	34
19	9.2	Login Page	34
20	9.3(a)	Products Page	35

21	9.3(b)	Products Page	35

S.NO.	Figure.No.	NAME OF THE FIGURE	Page. No.
22	9.3(c)	Products Page	36
23	9.3(d)	Products Page	36
24	9.4(a)	Cart Page	37
25	9.4(b)	Cart Page	37
26	9.5	Delivery Address Page	38
27	9.6	Shipping Page	38
28	9.7	Sign Out Page	39
29	9.8	Administraor Panel Page	39
30	9.9	Checking Stock Page	40
31	9.10	Billing Page	40

CHAPTER – 1
INTRODUCTION

Digitalization has sown its seed in the current world. With the growth of this digital plant our application creates an ecosystem for the customers to interact and integrate with the offline stores. Today online shopping becomes so enormous that effect badly on local and small shops. It has become difficult for a large section of people to adapt to this change. Our platform displays the products in offline stores that are available to the customers. The customers may visit the local online stores to buy them. This would help the offline store owners to sustain and also provide the customers with better prices.

Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. An online shop, e-shop, e-store, internet shop, online store, or virtual store evokes the physical analogy of buying products or services at online shopping centre. The process is called Business-to-Consumer Online Shopping.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the local shop anywhere through internet. Thus the customer will get the service of online shopping and home delivery from his favorite shop.

Project Objective:

The objective of the project is to make a web application to purchase items in an existing shop from our local stores. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online local shopping experience is the basic objective of the project. The application can be implemented in the form of a desktop application. To provide a desktop application for online shopping of products in an existing shop.

Project Overview:

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the items and articles of their desire from the store. The information pertaining to the products are stores on an SQLITE at the server side.

The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database. The application which is deployed at the customer database, the details of the items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, several reports could be generated as per the security.

Project Scope:

Online shopping is rising day by day in India. Because India is the country where computer users are increasing day by day so as the online shopping trends are also increasing. This project covers the online selling of cosmetics, fashion accessories etc. The project shows the product category and then product details. From the product details, the product can be added to cart and can be bought.

This system can be implemented to any shop in the locality having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, since the application is easily accessible and always available.

Problem Statement:

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands, If shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

Today online shopping become so enormous that effect badly local and small shops. With the growth of this digital plant, it has become difficult for a large section of people to adapt to this change. Many of them find it tedious to sustain in this world. Offline store owners are one among them. This may be due to lack of awareness about technology and unable to invest more amount of money for their store promotion. With the aim to strengthen the local market crashed by major eCommerce websites. Local Shopping Zone has aimed to connect all local store keepers ranging in all the categories.

Basic Definitions and Background:

- Electronic business is one of the fundamental criteria of unrest of data technology and correspondence in the field of economy.
- The basic ideology of this project is to connect various stores through a single platform.
- To help the offline stores to increase their sales.
- To give more preference to the local stores and make them sustain better than before.

CHAPTER - 2

LITERATURE SURVEY

Consumers favour online shopping is also due to price comparisons (Monsuwé et al., 2004; Palmer, 2013). The qualitative aspect of the research project was pursued through surveys and in-depth questionnaires. Barnes (2013) questions the reasoning why online shopping has become so popular. Many authors (Chaing and Dholakia, 2003, Monsuwé et al., 2004 and Poulter, 2013) believe one of the key reasons. Shopping online offers pronounced convenience Shim et al. 64 (2004) identify how previous online shopping experiences can have an impact on internet shopping. In the case of a positive outcome, consumers are likely to shop again. However, those experiencing frustrations may disregard the internet as a successful shopping channel.

Reference [1] says about the e-commerce technology. It involves buying and selling of goods and services, or the transmitting of funds or data, over a network, primarily it is a concept shift influencing both marketers and the customers. Rather e-commerce is more than just another way to enhance the existing business practices. It is foremost a complete change in traditional way of doing business. From the above research studies, we take the information about the impact of local stores and retailers due to heavy increase of online stores and implement that condition to our application to reduce the stated problem. The application of our project provides the importance for the local stores to sustain further. Devops is an environment used for the application to develop further.

CHAPTER – 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Sometimes products couldn't reach on time as per their expected schedule. Their track order facility is not so good. It doesn't work sometimes. Delay in delivery. The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item. Lack of significant discounts in online shops**,** Lack of interactivity, close examination.In online shopping if a user orders a product they may gets the product in different sizes. In local shopping zone they check the products from mobile itself and they buy the product by wearing it and checks does it fit or loose.

User must go to shop and select products.

1. It is less user-friendly.
3. It is difficult to identify the required product.
4. Description of the product limited.
5. It is a time consuming process
6. Not in reach of distant users.

Returns:

If the order placed arrives in poor condition, incomplete or sent something different you can choose to make a return but just as it is difficult to make a face to face, it is also something complex to return things when they are purchased online. You have to take into account the conditions of each page before buying something to avoid disappointment with what is received because it can also cost money to make the return.

Shipping costs:

It has shipping options in most cases where they are free within the country, but if you buy something abroad costs can rise too much, to the point that it is cheaper to leave home and go to the branch nearest.

Shipping time:

When buying online we have to take into account that depending on the receiving address, we will have to wait some time to receive our purchase. Not always met the estimated deadline and that can bring us problems, especially when we travel.

Disadvantages of Existing System

- Delay in delivery.
- Lack of significant discounts in **online** shops.
- Lack of touch and feel of merchandise in **online shopping**.
- Lack of interactivity in **online shopping**.
- Lack of **shopping** experience.
- Lack of close examination in **online shopping**
- Frauds in **online shopping**.
- This system effect badly on local and small shops.

3.2 PROPOSED SYSTEM

With the aim to strengthen the local market crashed by major eCommerce websites. The basic ideology of this project is to connect various stores through a single platform. Our System creates an ecosystem for the customers to interact and integrate with offline stores, which displays products in offline stores that are available to the customers. To help the **offline stores** to increase their sales and also provide the customers with better prices. Electronic business is one of the fundamental criteria of unrest of data technology and correspondence in the field of economy. We can discover almost every retail item accessible, at the least expensive costs and we can do as such from any area without stressing over heading off to the store. To give more preference to the local stores and make them sustain better than before.

Customers are keen on highly customized content according to their preferences. It's like contributing them a customized communication in the language they speak and understand. User-centric organization is critical in making their experience amusing. Organizing can be based on a user's attention, behaviour, location, culture etc. Local Shopping Zone creates an ecosystem for the customers to interact and integrate with offline stores, which displays products in offline stores that are available to the customers.

The central concept of the application is to allow the customer to shop virtually using the Internet and allow customers to buy the products from the Local stores. In comparison to the present system the proposed system will be less time consuming and is more efficient. Analysis will be very easy in proposed system as it is automated. The proposed system is very secure as no chances of loss of data as it is dependent on the administrator only.

In the proposed system customer need not go to the shop for buying the products. He can order the product, he wish to buy through the desktop application. The shop owner will be admin of the system. Shop owner can appoint moderators who will help owner in managing the customers and product orders. The system also recommends a home delivery system for the purchased products.

The application was designed into two modules first is for the customers who wish to buy the products. Second is for the administrator who maintains and updates the information pertaining to the products and those of the customer and aimed to connect all local store keepers ranging in all the categories like grocery, electronics, cosmetics etc.

3.2.1 Advantages of Proposed System

- Lower prices.
- More saving options.
- Research capabilities.
- Increased variety.
- Convenience.
- Saves time.
- Better prices.
- More variety.
- More control.
- Easy price comparisons.
- No crowds.
- No pressure.

The main advantage is to increase the sales of Local Stores and to help them to sustain better than before.

CHAPTER 4
SYSTEM STUDY

4.1 Feasibility Study

In successful business feasibility study of projects is a critical factor. Sometimes we noticed that several projects be unsuccessful because of their incorrect facts or incorrect assumptions. Therefore, the basic features in any feasibility study must be ensuring that we are working with accurate information, exact statement, and the latest financial records. The main objective of feasibility study is to evaluate three types of feasibility; technical, operational and economic feasibility. In this stage, only the project cost of the development and implementation cost of the project is considered. Feasibility study may also depend on the numerous factors, like the risks of the project before and after project and potential returns of the investments. It is in simply verify the changes before the project design. Importance of performing a Feasibility Study. Some major benefits of conducting a feasibility study are the following. Conduct a feasibility study is always important because of it gives a clear idea of the dreamed project. It helps to identify the valid and proper reason to select the project. With the support of feasibility study project teams' members become more focused. It helps to decision-making on the project. Improve the achievement time by calculating several restrictions.

General criteria:

- The user's demonstrable requirements and how the proposed system meets them?
- What are possible resources are available and the present project utilized these resources?
- Likely impact of the proposed system on the organization and it is really fit with the organization.

The proper investigation and evaluation procedure of the proposed project is mainly identification of the possible problem, specification, expected performance and, the cost of each sub-system and selection of the best and efficient procedures. The outcome of the study is generally considered as formal proposal of the expected system. This is basically a report about the scope of the projected solution. The proposals are summarizing with the expected so. It consists of the following:

1. Brief statement of the proposed project –a carefully worded and may be the graphical statement of the proposed problem.
2. Summary of the outcome - a list of the major outcome from the study.

3. Details of findings- an outline of the proposed methods and procedures of the desired plan undertaken to complete task with all possible finding. This report also included the possible structures, costs estimation and benefits of the desire projects.

4.Recommendations and final conclusions- Few specific advice and recommendations based on its outcome along with the personnel assignments, projected costs and target dates. Basically, it is a predictable project that demonstrate conclusion of the assessment, examine and evaluation of a designed method and terminate if this method is accurately feasible, profitable and productive.

4.1.1 Technical Feasibility

Technical feasibility mainly associated with the technologically evaluates the project. In this subject area generally a group of engineers or technical expert study the whole projects and technical aspects. This study facilitates said organizations to proper assess. The industrial possessions may assemble capability. Based on the results it decides whether the technical team can convert the idea into real.

4.1.2 Economic Feasibility

Economic feasibility study related with price, and all kind of expenditure related with the scheme before the project start. This study also improves project reliability. It is also helpful for the decision-makers to decide the planned scheme processed latter or now, depending financial condition of the organization. This evaluation process also studies the price benefits of the proposed scheme.

4.1.3 Operational Feasibility

Operational Feasibility may employ the responsibility to examine and decide whether the proposed methods fulfill all kind of business requirements. Its actions forecast all possible schemes to recognized and resolve troubles. These studies may also examine and verify how the project planned guarantee the method development is feasible or not.

4.1.4 Social Feasibility

Social feasibility is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

SYSTEM DESIGN

SYSTEM DESIGN:

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. It emphasis on translating design. Specifications to performance specification. System design has two phases of development

- Logical design
- Physical design

During logical design phase the analyst describes inputs (sources), output (destinations), databases (data sores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

5.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

5.1.1 SYSTEM ARCHITECTURE

A system architecture diagram explains about the relationship between various mechanisms.

Generally they are formed for systems which include hardware and software and these are represented in the diagram to illustrate the interaction between them. on the other hand, it can also be formed for web applications.

For a desktop app, the system architecture design would include workings such as, database, application server, web server, internet etc.

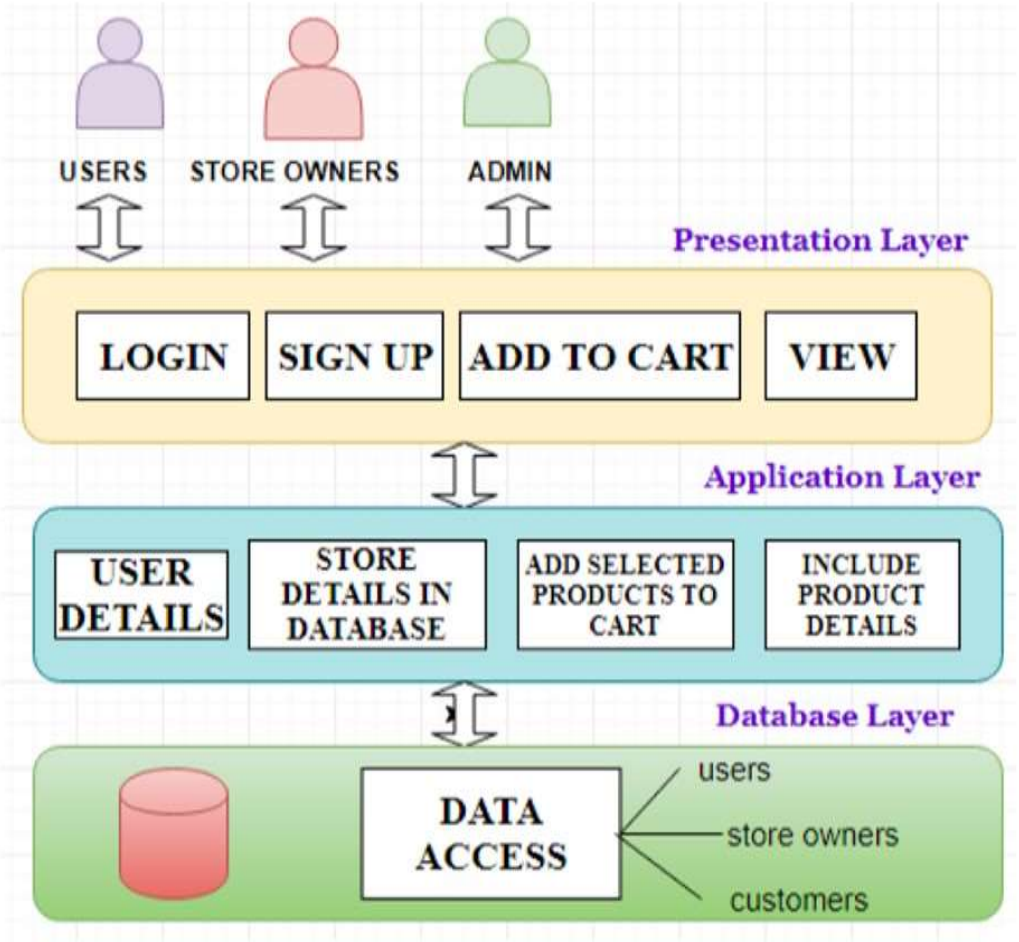

Fig 5.1.1: System Architecture

5.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

DATABASE DESIGN:

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

Primary key- The field that is unique for all the record occurrences.

Foreign key-The field used to set relation between tables.

Normalization is a technique to avoid redundancy in the tables.

TABLES

- **LOGIN TABLE**

login

Field Name	Field Type	Default	AllowNull	PriKey	Extra
uid	int(11) FIRST		NO	YES	auto_increment
uname	varchar(50) AFTER `uid`		NO	NO	
upass	varchar(50) AFTER `uname`		NO	NO	
utype	varchar(20) AFTER `upass`		NO	NO	
enabled	varchar(10) AFTER `utype`		NO	NO	

Fig 5.1.2(a): Login Table

- **USER DETAILS TABLE**

userdetails

Field Name	Field Type	Default	AllowNull	PriKey	Extra
uid	int(11) FIRST		NO	YES	
uloginname	varchar(50) AFTER `uid`		NO	NO	
uemailid	varchar(100) AFTER `uloginname`		NO	NO	
umobno	varchar(30) AFTER `uemailid`		NO	NO	

Fig 5.1.2(b): User Table

- **PRODUCT DETAILS TABLE**

product_details

Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
name	varchar(100) AFTER `id`		NO	NO	
description	varchar(500) AFTER `name`		YES	NO	
price	varchar(50) AFTER `description`		YES	NO	
photo	varchar(100) AFTER `price`		YES	NO	

Fig 5.1.2(c):: Product Table

- **PRODUCT ORDERS TABLE**

product_orders

Field Name	Field Type	Default	AllowNull	PriKey	Extra
id	int(11) FIRST		NO	YES	auto_increment
product_id	int(11) AFTER `id`		NO	NO	
user_id	int(11) AFTER `product_id`		NO	NO	
deliver_address	varchar(500) AFTER `user_id`		YES	NO	

Fig 5.1.2(d): Product orders Table

5.1.3 BLOCK DIAGRAM

A **block diagram** is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic devise, software design, and process flow diagram. Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation. Contrast this with the schematic diagram and layout diagram used in electrical engineering, which show the implementation details of electrical components and physical construction.

The following figure 5.4 shows Block Diagram for Local Shopping Zone

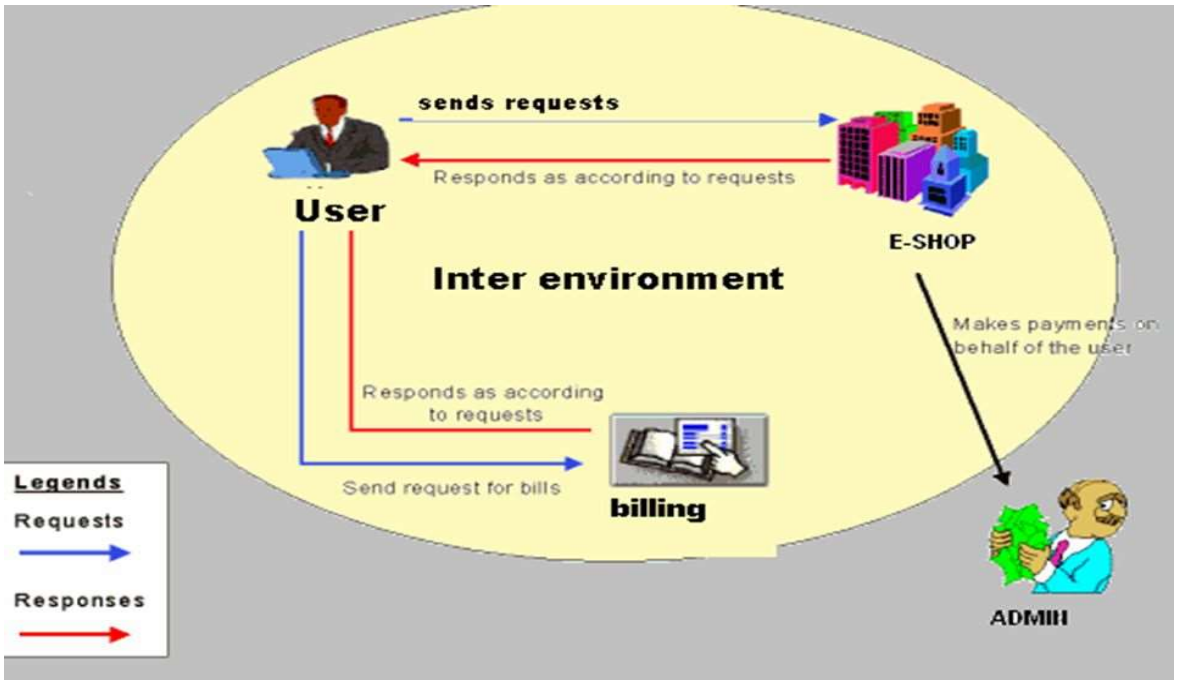

Fig 5.1.3: Block Diagram

The information regarding the products such as its name, model, price information, its features etc. The **ADMIN** has the authority to Add, Delete, Update etc. The **USER** can only view the Products, add to cart only those in the stock etc. and users to type a username and password to login. Here User can select any number of Products and add to the cart. He can also remove from the cart if he dislikes it later. This gives the details regarding the products available for sale to the Administrator.

5.2 Flow Chart Diagram

A simple flowchart representing a process for dealing with a non-functioning lamp. A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields.

Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

- A processing step, usually called activity, and denoted as a rectangular box
- A decision usually denoted as a diamond.

The following figure 5.7 shows Flow Chart Diagram for Local Shopping Zone

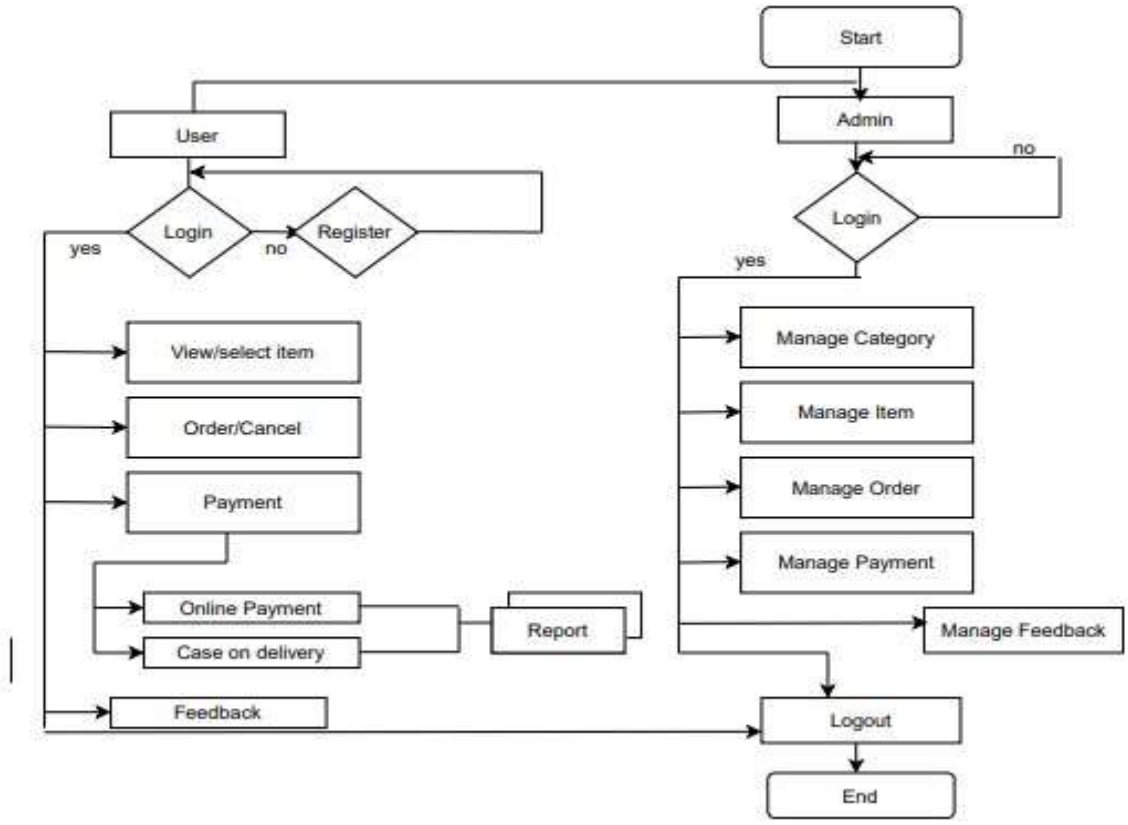

Fig 5.2: Flow Chart

5.3 UML DIAGRAMS

5.3.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as **Actors**.

The following figure 5.10 shows UML Diagram for Local Shopping Zone.

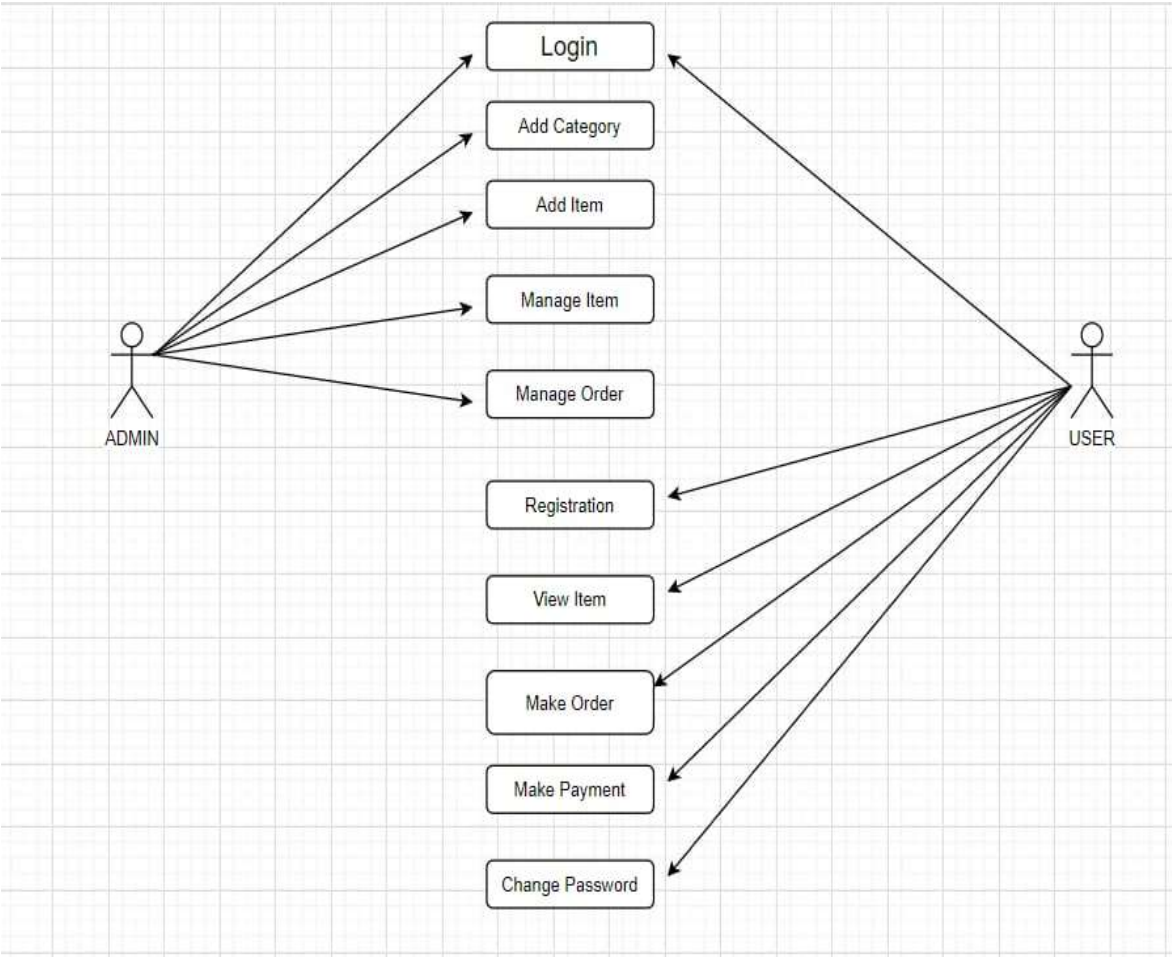

Fig 5.3.1: Usecase diagram.

5.3.2 Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective.

The following figure shows UML Diagram for Local Shopping Zone

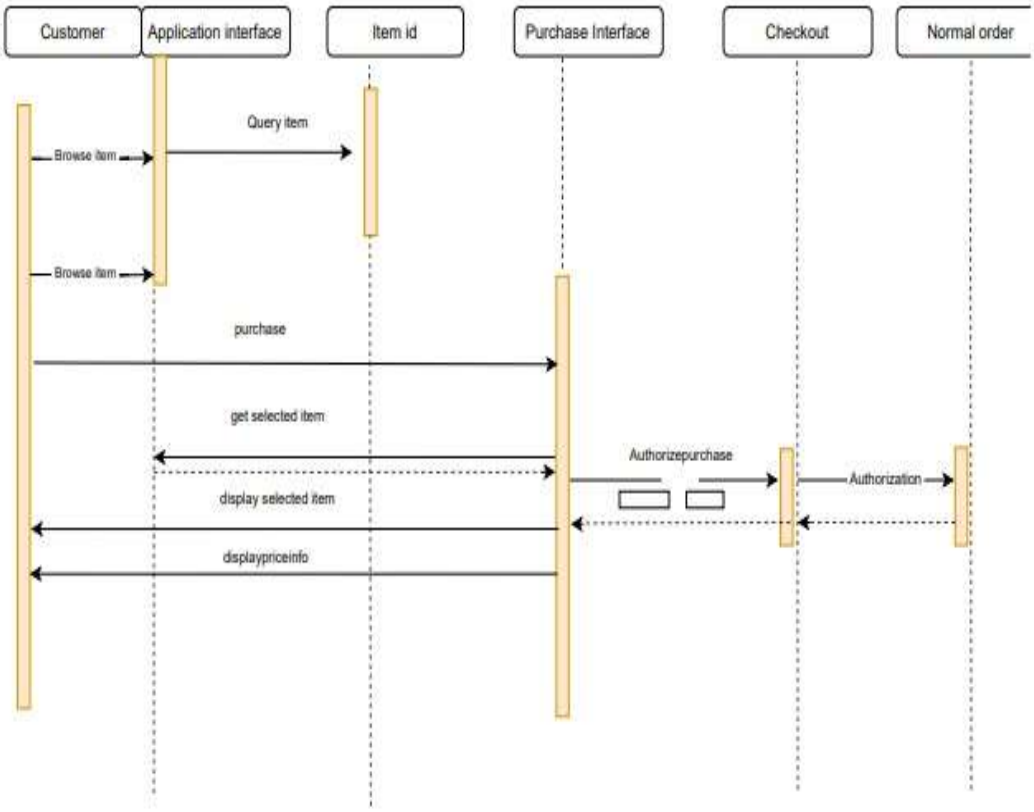

Fig 5.3.2: Sequence diagram.

5.3.3 Activity Diagram

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

The following figure shows UML Diagram for Local Shopping Zone

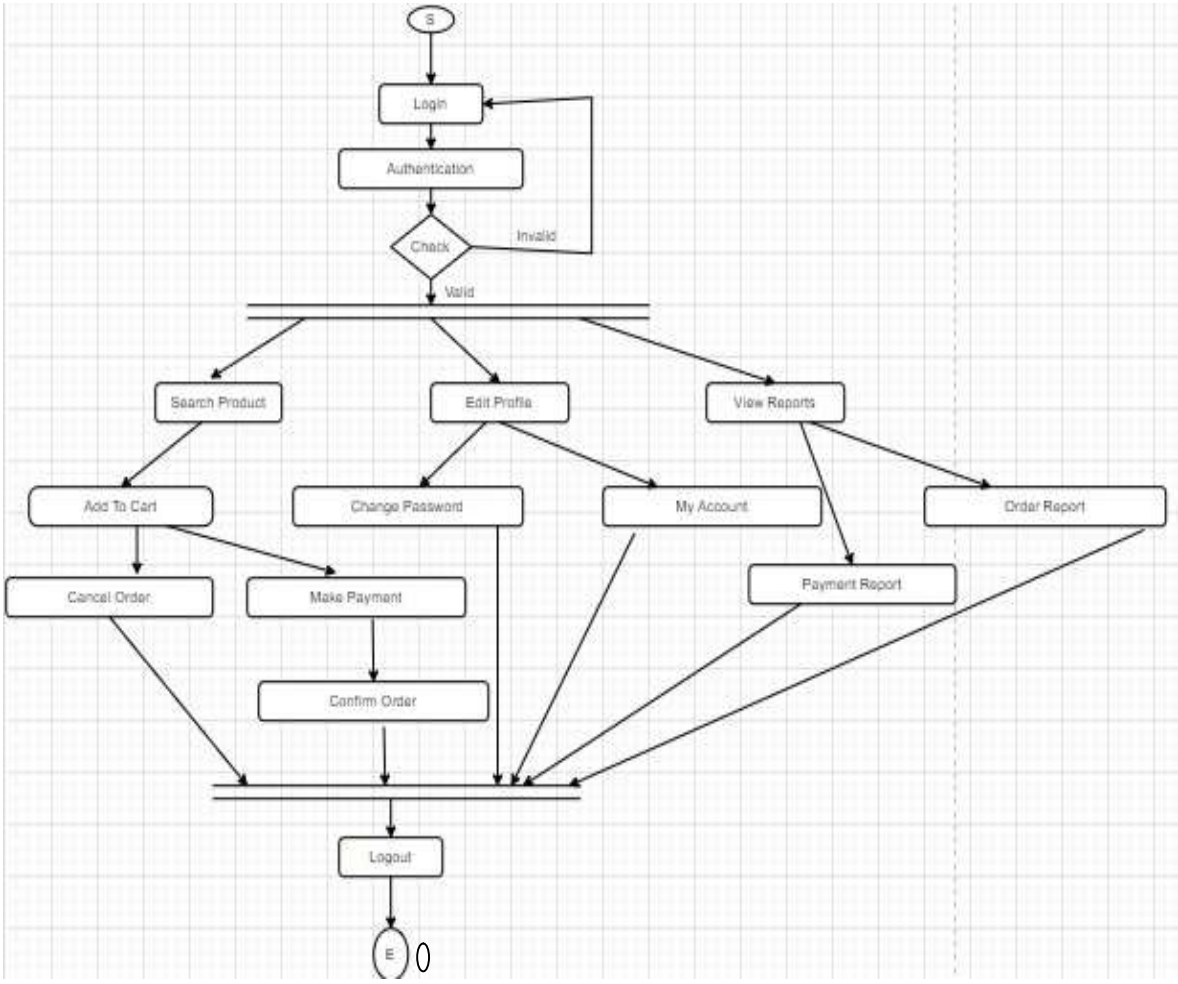

Fig 5.3.3: Activity Diagram

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Modules

The system after careful analysis has been identified to be presented with the following modules and roles. The modules involved are:

- Administrator
- Moderators
- Users

Administrator:

The administrator is the super user of this application. Only admin have access into this admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about all products. This module is divided into different sub-modules.

1. Manage Moderators
2. Manage Products
3. Manage Users
4. Manage Orders

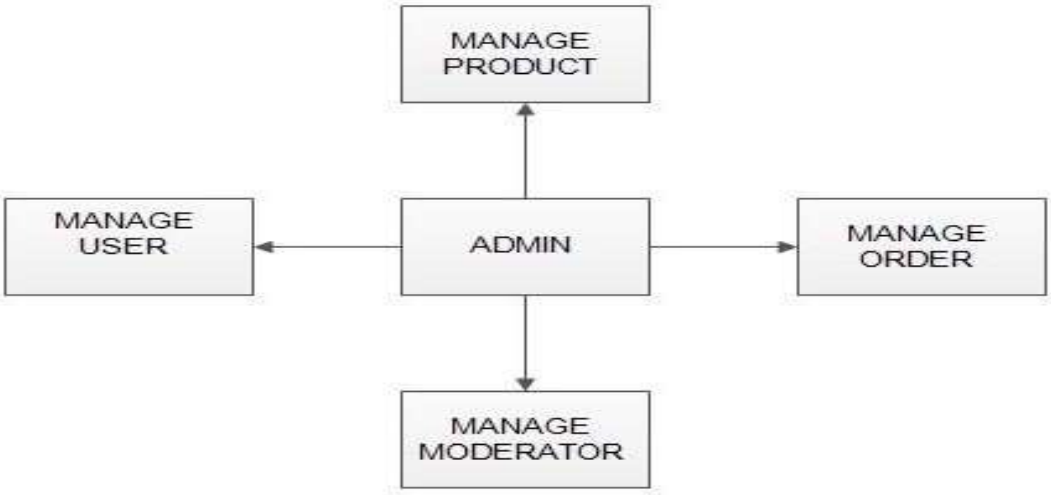

Fig 6.1: Admin module

MANAGE MODERATOR:

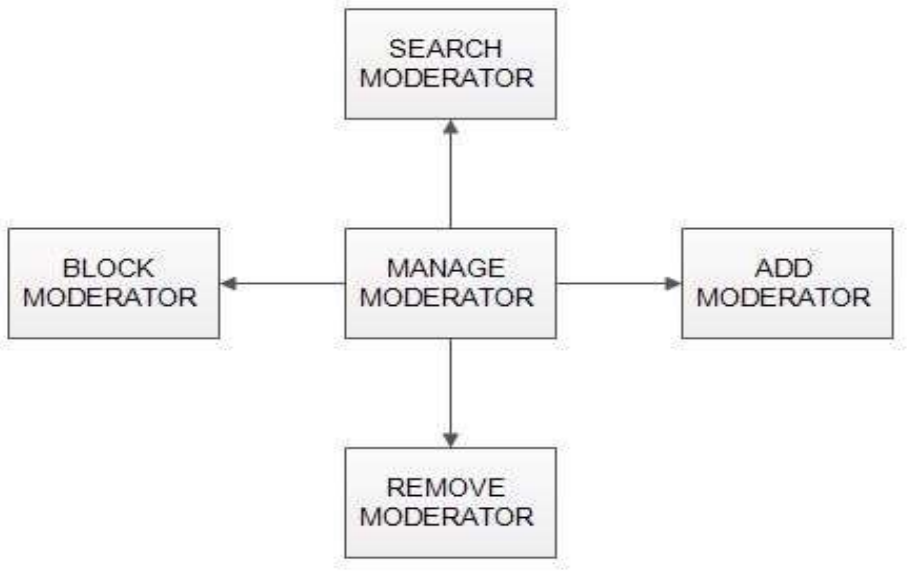

Fig 6.2: Manage Moderator

- **Add Moderator:**

Only admin is having the privilege to add a moderator. A moderator can be considered as a staff who manages the orders or owner of a group of products.

- **Block moderator**

Admin can restrict a moderator from managing the orders by blocking them. Admin can unblock a blocked user if needed.

- **Remove Moderator**

Admin has privilege to delete a moderator who was added.

- **Search moderator:**

All existing moderators can be viewed by the administrator as a list. If there is number of moderators and admin need to find one of them, the admin can search for a moderator by name.

Manage Products:

- **Add Products**

Shopping cart project contains different kind of products. The products can be classified into different categories by name. Admin can add new products into the existing system with all its details

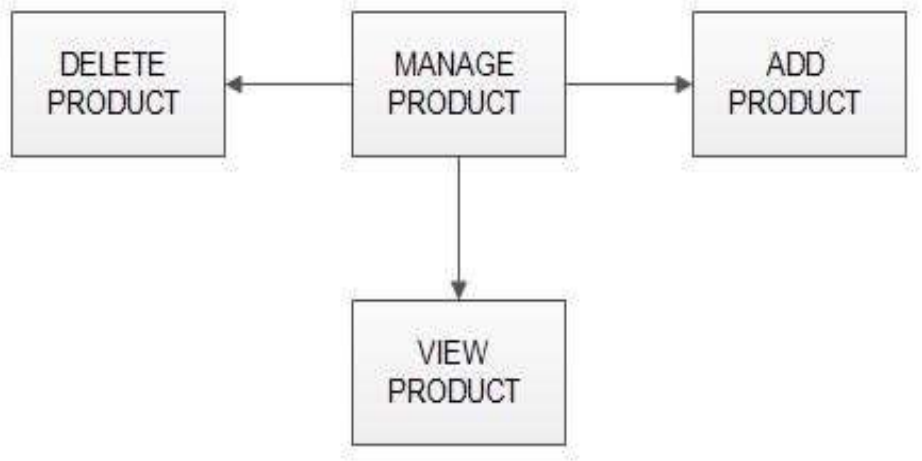

Fig 6.3: Manage Products

- **Delete Products:**

Administrator can delete the products based on the stock of that particular product.

- **Search Products:**

Admin will have a list view of all the existing products. He can also search for a particular product by name.

MANAGE USER:

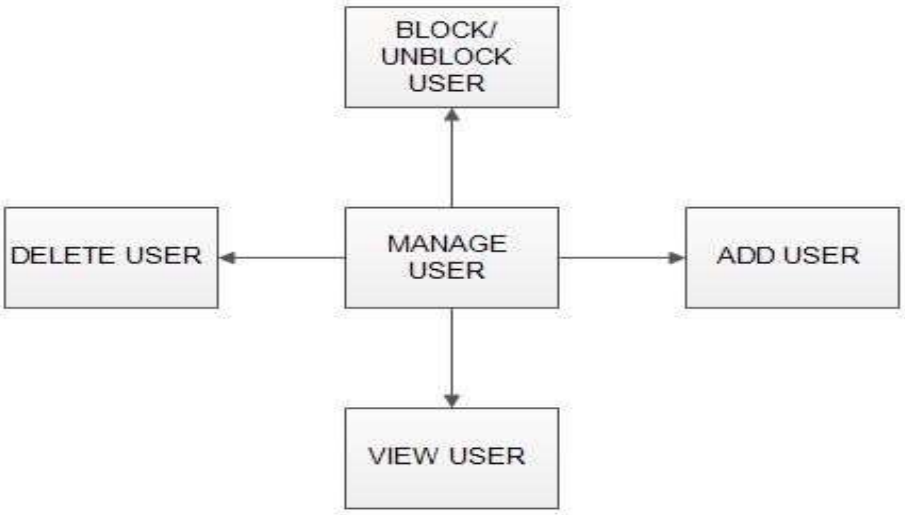

Fig 6.4: Manage User

- **View Users**

The admin will have a list view of all the users registered in the system. Admin can view all the details of each user in the list except password.

- **Add Users**

Admin has privileges to add a user directly by providing the details.

- **Delete &Block Users**

Administrator has a right to delete or block a user. The default status of a new user registered is set as blocked. The admin must accept the new user by unblocking him.

MANAGE ORDERS:

Fig 6.5: Manage Orders

- **View Order**

Administrator can view the Orders which is generated by the users. He can verify the details of the purchase.

- **Delete order**

Admin can delete order from the orders list when the product is taken for delivery.

MODERATORS

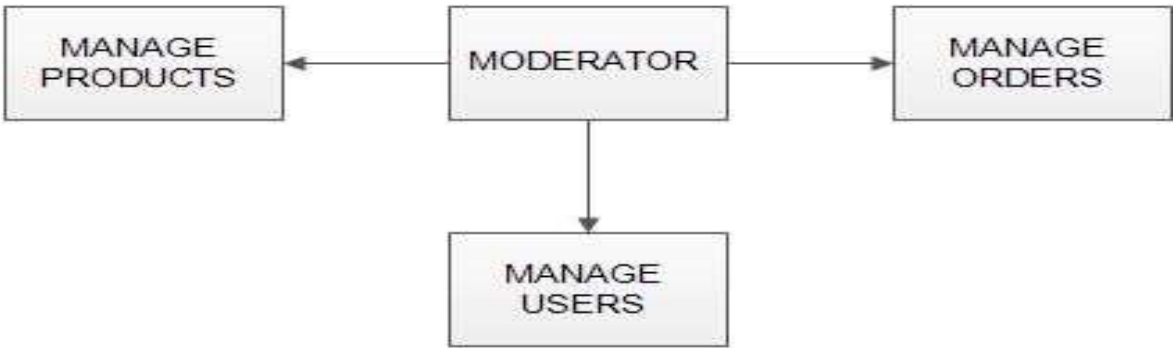

Fig 6.6: Moderator.

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the work load of admin. Now moderator has all the privilege an admin having except managing other moderators. He can add products and users. He can also check the orders and edit his profile.

- Manage products
- Manage users
- Manage orders

USER:

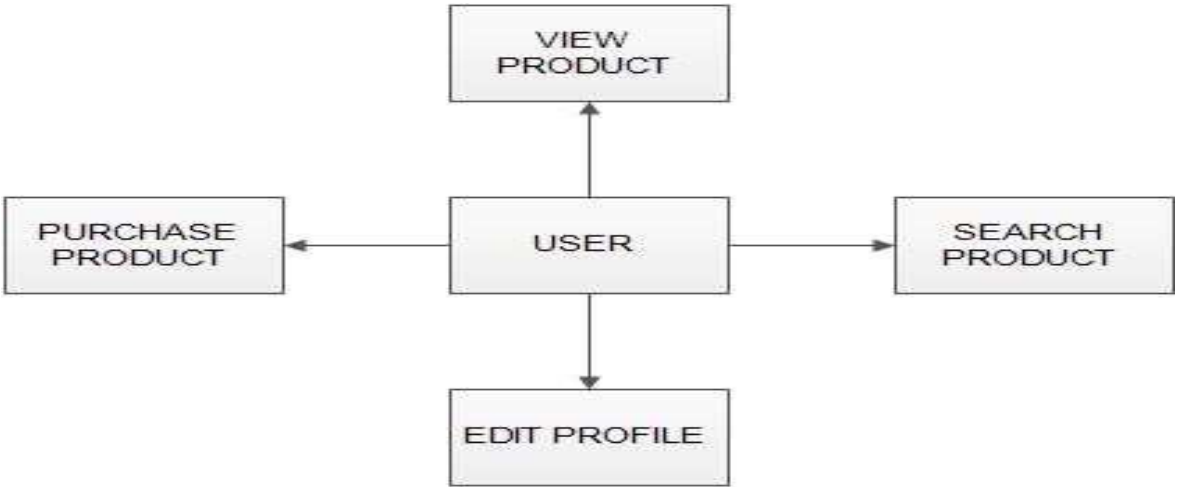

Fig 6.7: User Module

- **Registration**

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

- **Login**

A user must login with his user name and password to the system after registration.

- **View Products**

User can view the list of products based on their names after successful login. A detailed description of a particular product with product name, products details, product image, price can be viewed by users.

- **Search Product**

Users can search for a particular product in the list by name.

- **Add to cart:**

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove.

Submit Cart:

After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

CHAPTER - 7

SYSTEM REQUIREMENTS

NON FUNCTIONAL REQUIREMENTS

i. EFFICIENCY REQUIREMENT

When an online local shopping zone application implemented, customer can purchase product in an efficient manner.

ii. RELIABILITY REQUIREMENT

The system should provide a reliable environment to both customers and owner. All orders should be reaching at the admin without any errors.

iii. USABILITY REQUIREMENT

The android application is designed for user friendly environment and ease of use.

iv. IMPLEMENTATION REQUIREMENT

Implementation of the system using xml and java applets in front end with java as back end and it will be used for database connectivity. And the database part is developed by sqlite. Responsive desktop application is used for making the application compatible for any type of screen.

v. DELIVERY REQUIREMENT

The whole system is expected to be delivered in four months of time with a weekly evaluation by the project guide.

FUNCTIONAL REQUIREMENTS

USER

- **USER LOGIN**

Description of feature

This feature used by the user to login into system. A user must login with his user name and password to the system after registration. If they are invalid, the user not allowed to enter the system.

Functional requirement

Username and password will be provided after user registration is confirmed.

Password should be hidden from others while typing it in the field.

- **REGISTER NEW USER**

Description of feature

A new user will have to register in the system by providing essential details in order to view the products in the system. The admin must accept a new user by unblocking him.

Functional requirement

System must be able to verify and validate information.

The system must encrypt the password of the customer to provide security.

- **PURCHASING AN ITEM**

Description of feature

The user can add the desired product into his cart by clicking add to cart option on the product. He can view his cart by clicking on the cart button. All products added by cart can be viewed in the cart. User can remove an item from the cart by clicking remove. After confirming the items in the cart the user can submit the cart by providing a delivery address. On successful submitting the cart will become empty.

Functional requirement

System must ensure that, only a registered customer can purchase items.

ADMIN :

- **MANAGE USER**

Description of feature:

The administrator can add user, delete user, view user and block user.

- **MANAGE MODERATOR**

The administrator can add moderator, delete moderator, block moderator and search for a moderator.

- **MANAGE PRODUCTS**

The administrator can add product, delete product and view product.

- **MANAGE ORDERS**

The administrator can view orders and delete orders.

Functional requirements

The system must identify the login of the admin.

Admin account should be secured so that only owner of the shop can access that account.

MODERATOR:

A moderator is considered as a staff who can manage orders for the time being. As a future update moderator may give facility to add and manage his own products. Moderators can reduce the work load of admin. Now moderator has all the privilege of an admin having except managing other moderators. He can manage users and manage products. He can also check the orders and edit his profile.

Functional requirement

The system must identify the login of a moderator.

7.1 Software Requirements:

- **Front end** : XML, Java Applets
- **Backend** : JAVA
- **Data Base** : SQLITE
- **Connection** : JDBC
- **Tools** : Eclipse JEE

CHAPTER-8

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement**.**

Testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process where that test data is prepared and is used for testing the modules individually and later the validations given for the fields. Then the testing takes place, which make sure that all components of the system properly function as a unit by actually trying to force the system to fail. Testing is the best performed when employee departments are asked to assist in identifying all possible situations that mightarise.

a) Testing is a process of executing a program with the intent of findingerrors.
b) A good test is one that has high probability of finding an undiscoverederror.
c) A successful test is one that uncovers yet undiscoverederror.

8.1 TYPES OF TESTING:

All tests should be planned long before. Testing should begin “in the small” and progress towards testing “in the large”. The first tests planned and executed generally, focus on individual program modules. As testing progresses, testing shift focus in an attempt to find errors in integrated clusters modules of ultimately in the entire system. Testing of debugged program is one of the most critical aspects of computer programming because without that work, the system will never produce the output for which it is designed. The philosophy behind testing is to find error. Hence, a successful test is one that finds errors. Actually testing is the state of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Any engineered product can be tested in one of the twoways.

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration testing is a systematic technique for constructing the program structures while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by thedesign. Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be

considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

Once individual module testing is completed, modules are assembled and integrated to perform as a system. Then the top-down testing begins from upper-level to lower-level module testing was carried out to check whether the entire system is performing satisfactorily. This is the test done for the whole system integration after completion of the moduletesting.

Test Data and Output:

For all types to test dummy data were keyed in and after the testing was over live data were keyed in from the source documents and concerned Department people were followed to key in and work on the system.

ModuleTesting:

To locate errors, focus is given on the modules, independent of other modules. This is known as module testing. This enables us to detect error and correct it without affecting any other modules. When ever the program was not executing the required function, it was corrected to get the required results. Thus all the modules are tested from bottom to top starting with the smallest and lowest level modules and proceeding to next level. Individual like purchase order rising was done successfu11y for connection to the next modules.

SpecificationTesting:

Executing the specification starting what the program should do and how it should perform undervarious conditions. Test cases for various situations and combination of condition in all the modules were tested.

Code Testing:

This examines the logic of the program. For example, the logic for calculating profit, rate taking the percentages entered by employee was tested and verified with the data that

was carried out by manual methods

User InterfaceTesting:

The data that is given by the user should re-lie on the particular range. When the employee gives wrong data, the system will give error message. According to this, the program was developed. The value of the data should fall on the given range of value

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

All tests should be planned long before. Testing should begin "in the small" and progress towards testing "in the large". The first tests planned and executed generally, focus on individual program modules. As testing progresses, testing shift focus in an attempt to find errors in integrated clusters modules of ultimately in the entire system. Testing of debugged program is one of the most critical aspects of computer programming because without that work, the system will never produce the output for which it is designed. The philosophy behind testing is to find error. Hence, a successful test is one that finds errors. Actually testing is the state of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Any engineered product can be tested in one of the twoways.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Black Box Testing attempts to find errors in following categories,

i. In-correct or missingerrors
ii. Errors in data structures or external databaseaccess.
iii. Performance errors that Initialization and terminationerrors.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration testing is a systematic technique for constructing the program structures while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by thedesign.

Top-Down Integration:

It is an incremental approach to the construction of program structure. Modules are integrated by moving down ward through the control hierarchy, beginning with main control. Modules subordinate to the main control modules are incorporated into the structure in either depth first or breadth firstmanner.

Bottom-Up Integration:

Bottom up integration, as suggests, begins construction and testing atomic modules. Since modules are integrated from bottom up, processing required for modules subordinates to a given level is always available and the need for stubs iseliminated.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance test is conducted to enable the customer to validate all requirements. It can range from an informal test drive to a planned and systematically executed series of tests they are Alpha testing and beta testing at the developer site. The software is used in a natural setting with the developer looking over the shoulder of the employee and recording errors and usage problems. These tests are conducted sites by the software in an environment, which cannot be controlled by the developer.

Testing is a process which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of test cases, output of the program for the test cases is evaluated to determine if the program is perform as it is expected to perform and order to make sure that the system does not have error

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Validation Testing:

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected, and a final series of software testing validation testing may begin. Validation succeeds when the software function in a manner that can be reasonably accepted by a customer. Software validation is achievedthrough a series of black box tests that demonstrate conformity with requirements. After each validation testing, one of the two possible conditions exit.

The function or performance characteristics conform to the specification and are accepted. A deviation from specification is uncovered and deficiency list is created. The validation test was carried out on all modules. Most of the characteristics conformed to the specifications and where a deviation was encountered, the appropriated corrective measure was taken. In all validations test was executed successfully.

CHAPTER 9
RESULTS

SIGNUP PAGE:

This module is for the users who do not have their account. The account creation is done by filling the registration form with user details.

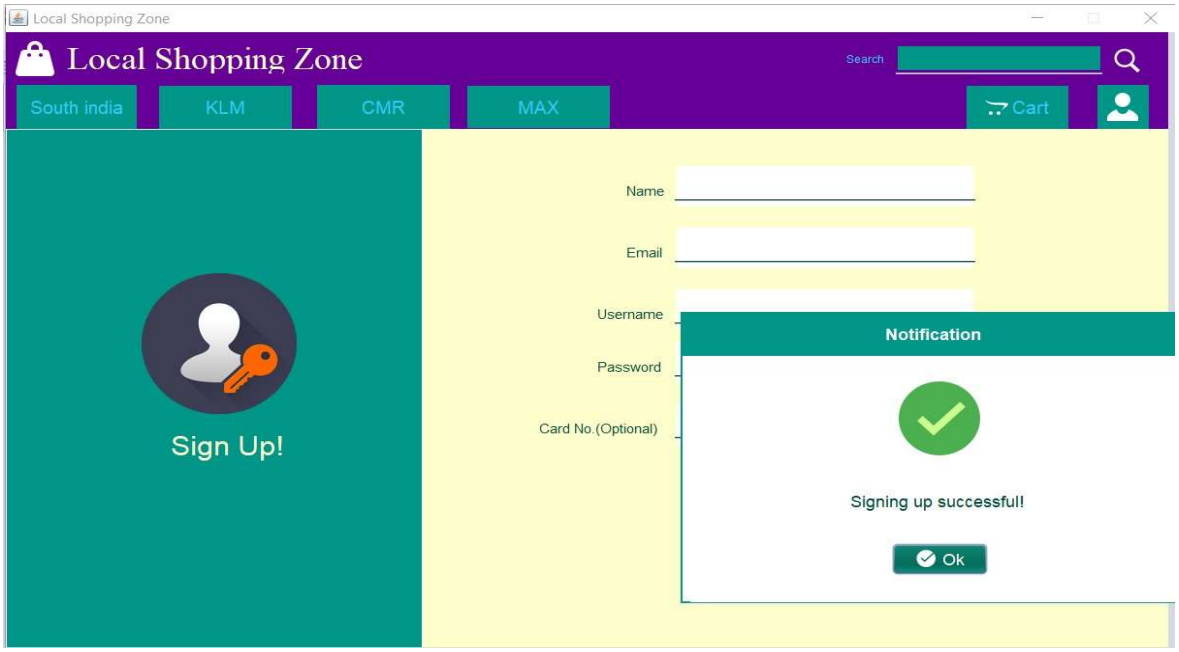

Fig 9.1: Sign up Page

LOGIN PAGE:

The login page allows a user to gain access to an application by entering their username and password or by authenticating using a social media login.

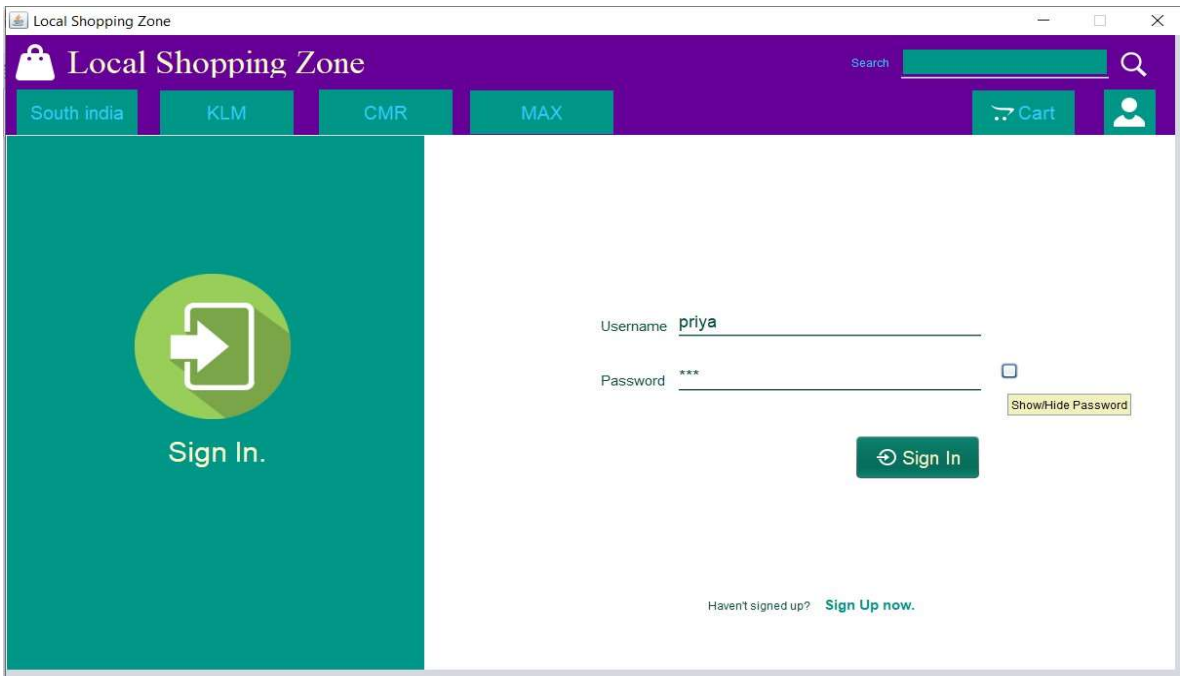

Fig 9.2: Login Page

PRODUCTS PAGES:

This module has information regarding the products such as its name, model, price information, its features etc. The ADMIN has the authority to Add, Delete, Update etc. The USER can only view the Mobile, add to cart only those in the stock etc.

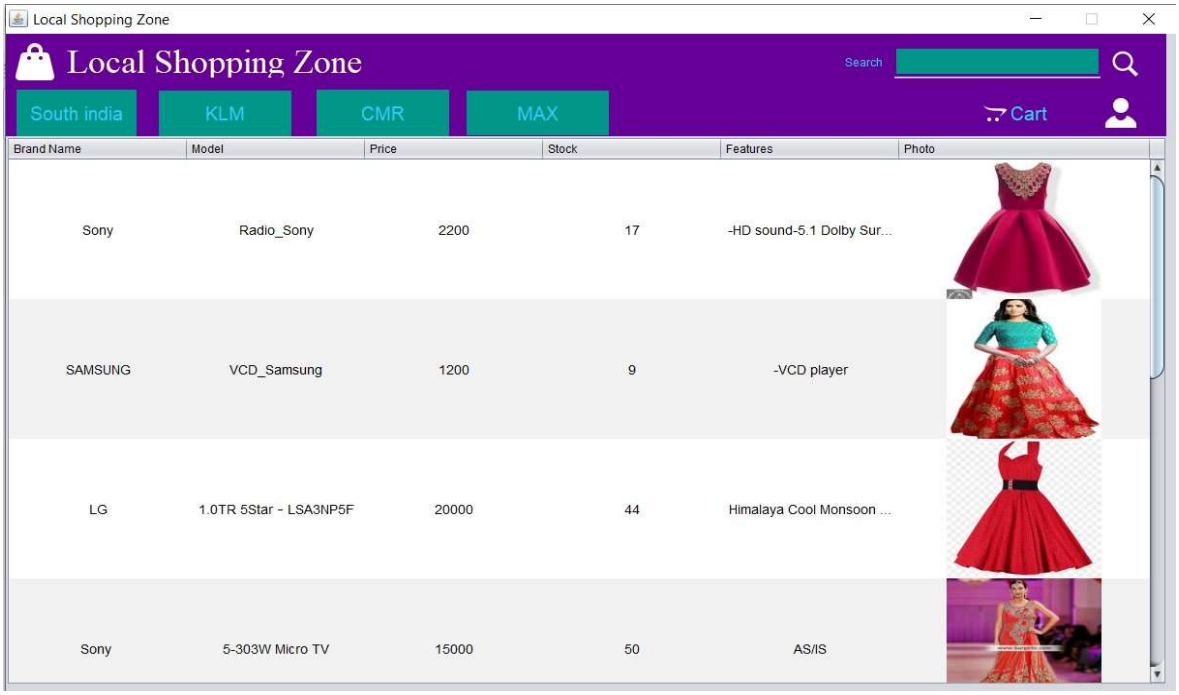

Fig 9.3(a): Products Page

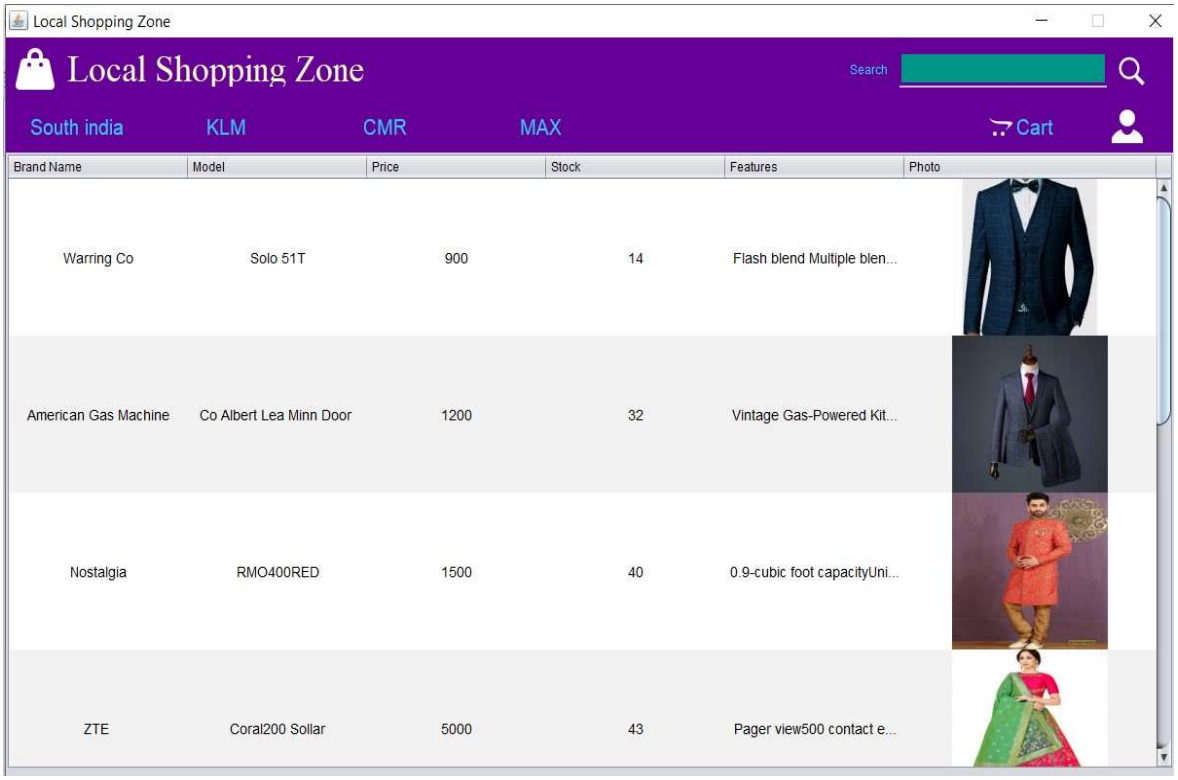

Fig 9.3(b): Products Pages

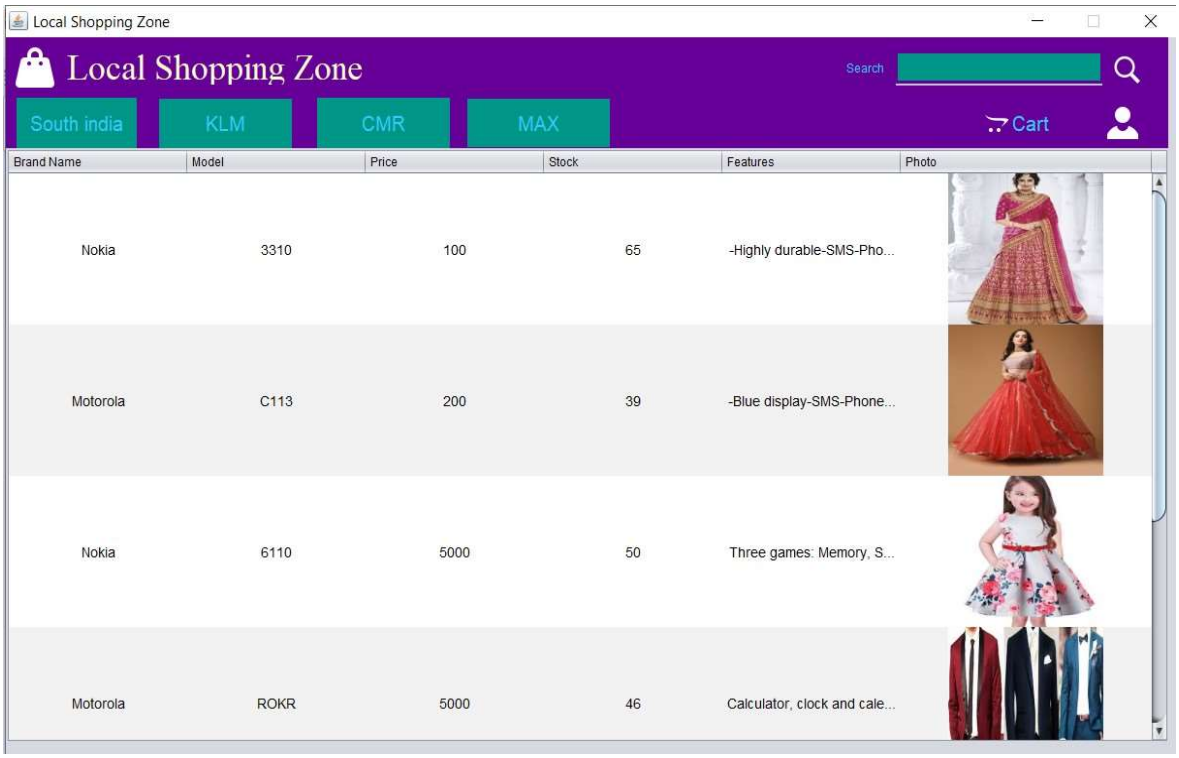

Fig 9.3(c): Products Page

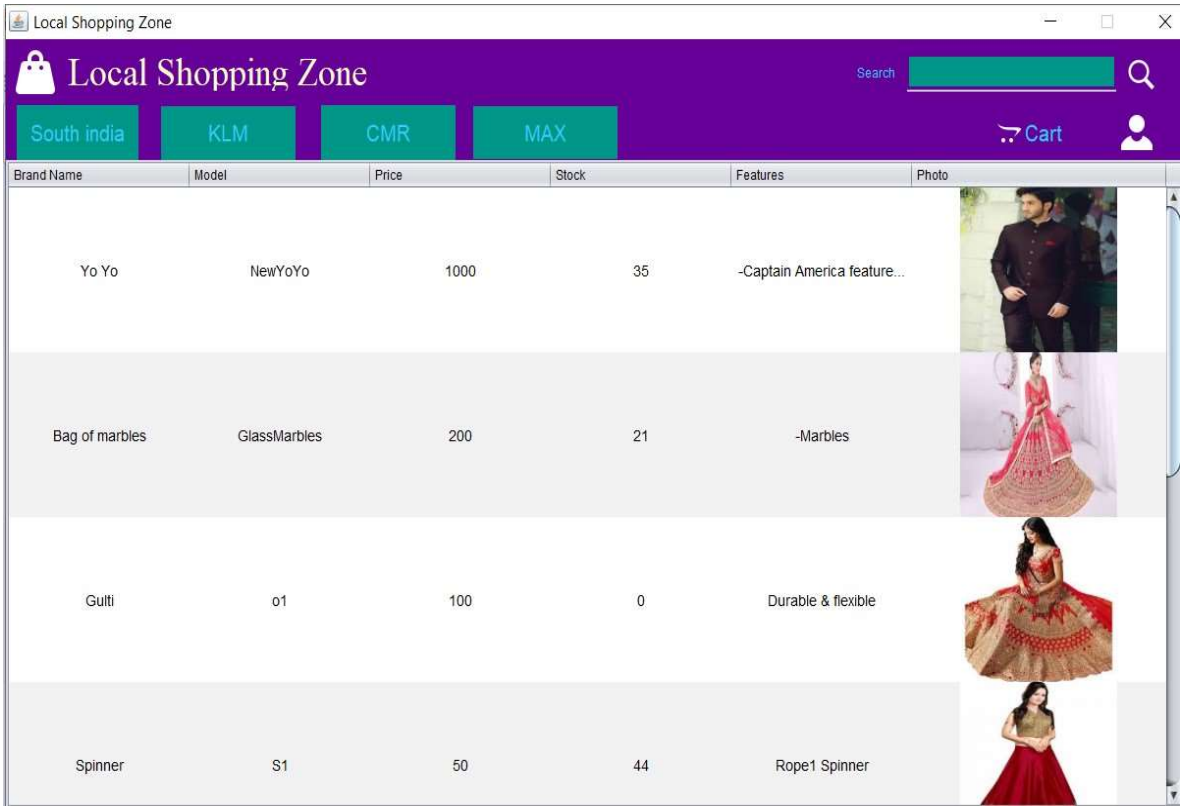

Fig 9.3(d): Products Page

CART PAGE:

User can select any number of Products and add to the cart. He can also remove from the cart if he dislikes it later.

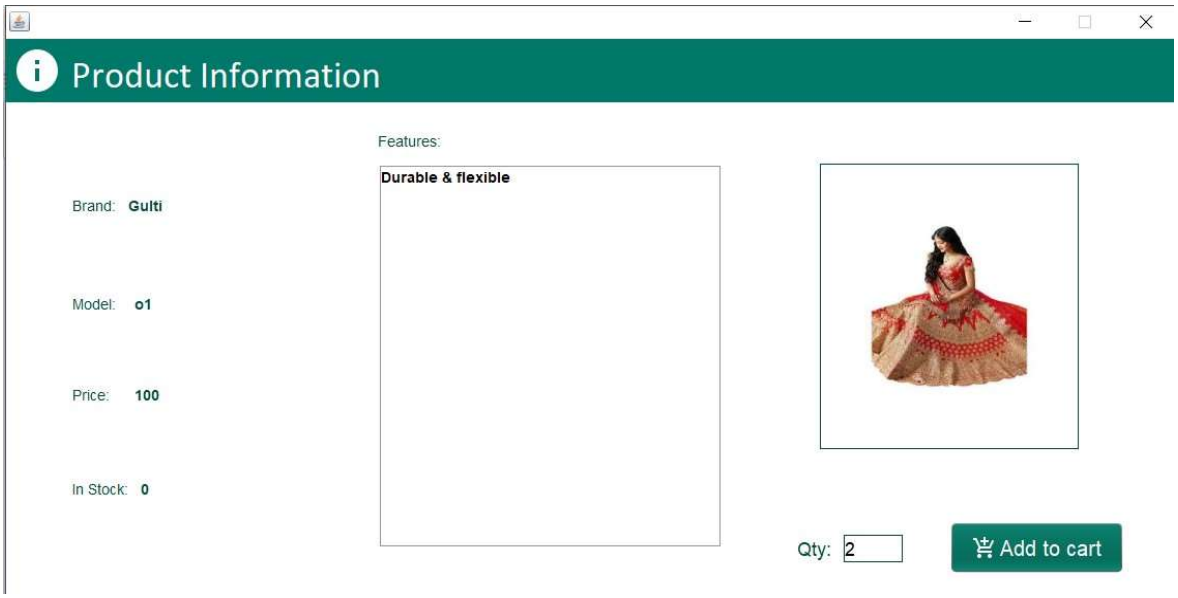

Fig 9.4(a): Cart Page

Fig 9.4(b): Cart Page

DELIVERY ADDRESS PAGE :

The delivery address is the address that the buyer wants the order or shipment to arrive at.

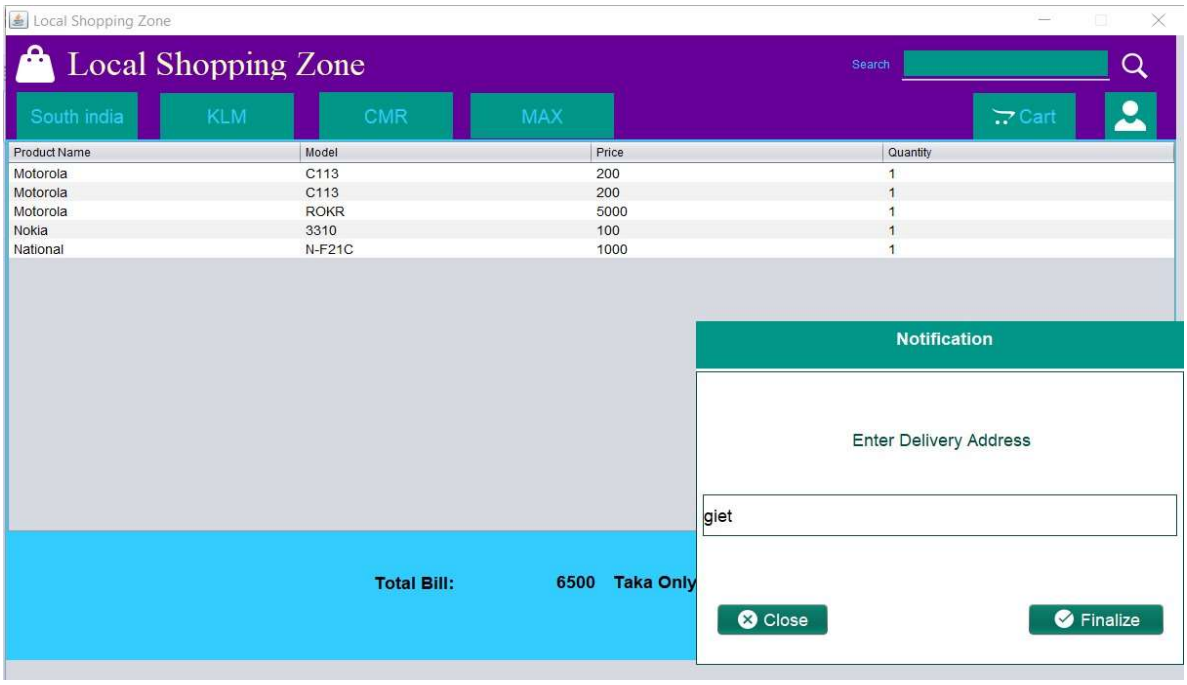

Fig 9.5: Delivery Address Page

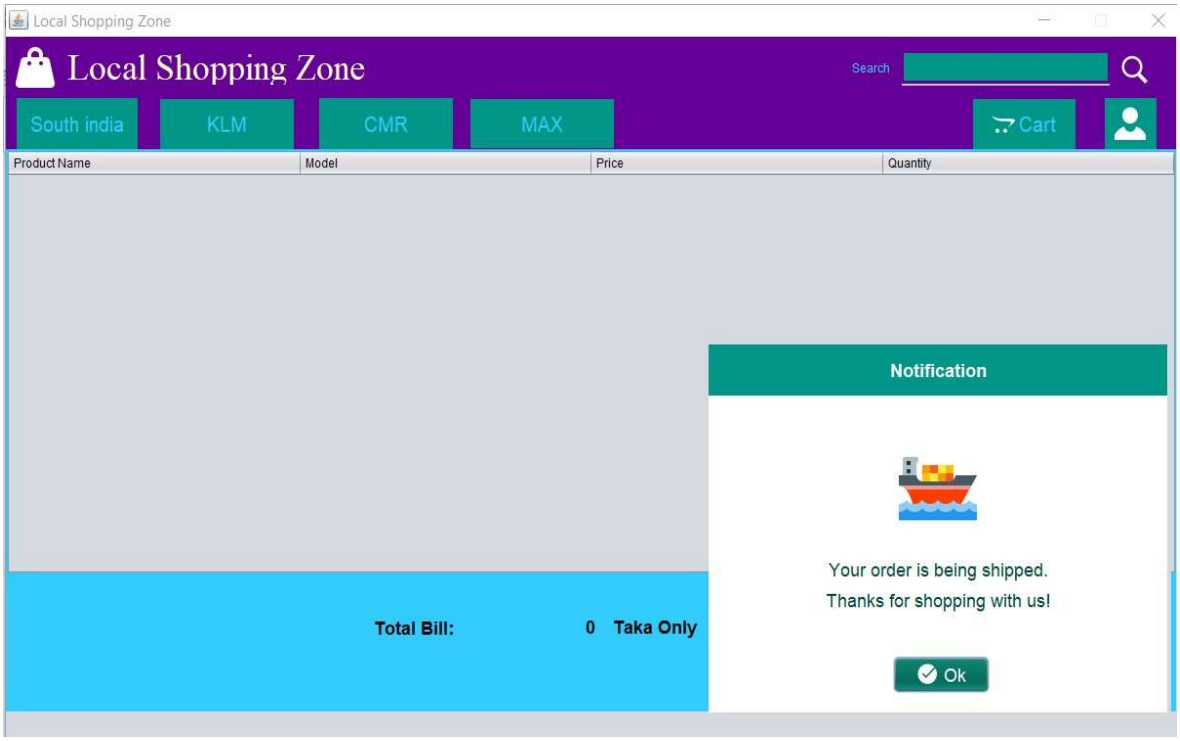

Fig 9.6: Shipping Page

SIGNOUT PAGE :

Loging **out** means to end access to a computer system or a website. Logging out informs the computer or **website** that the current user wishes to end the login session.

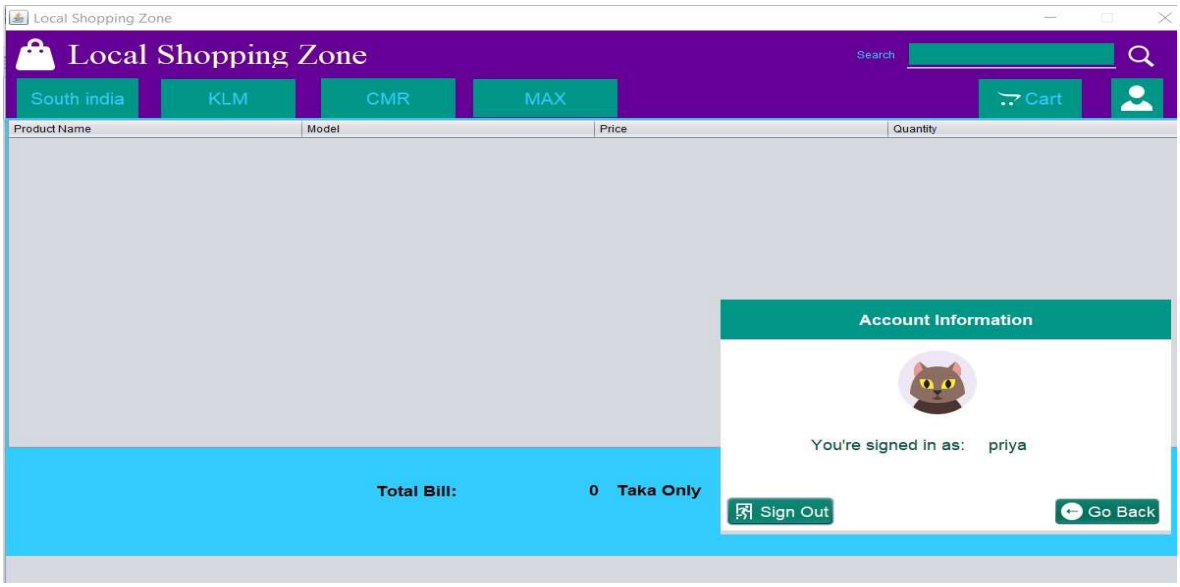

Fig 9.7: **Sign Out Page**

ADMINISTRATOR PANEL :

The **admin panel** is where new posts, categories, tags, pages, links and custom post types are created. It's also where theme files are changed, widgets are added, plugins are activated or updated, and reading/writing/general settings are changed.

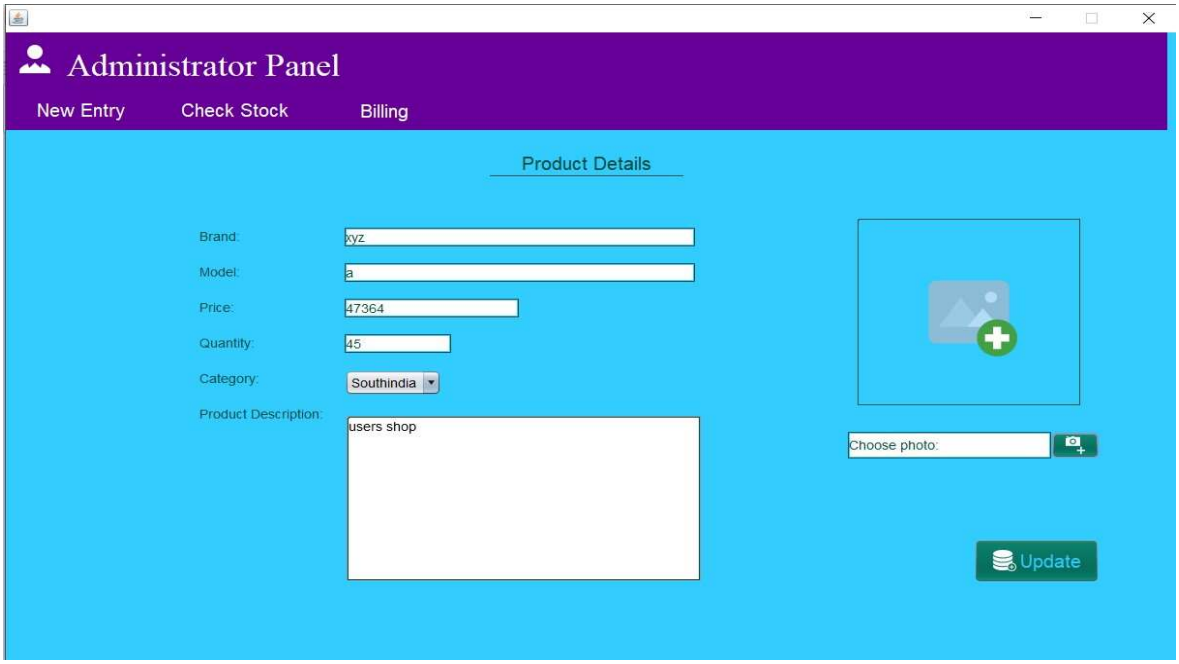

Fig 9.8: **Administrator Panel Page**

Checking Stock Page :

To check the current available stocks in the application.

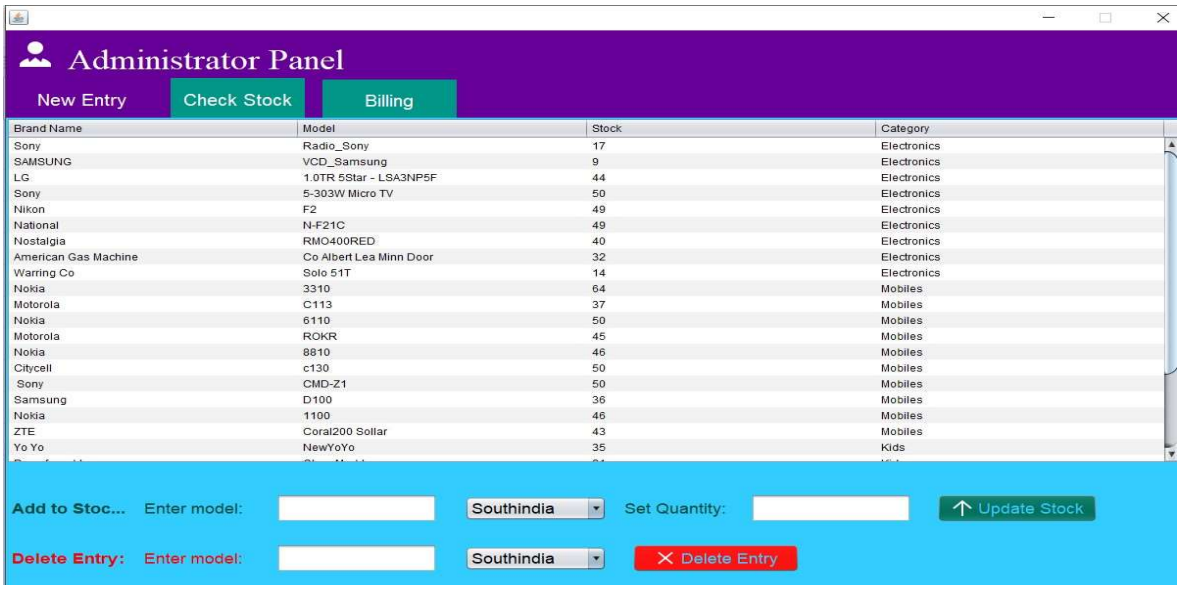

Fig 9.9: Checking Stock Page

BILLING PAGE :

Billing information means any data that enables any person to access a consumer's account, including but not limited to a credit card, checking, savings, share or similar account, utility bill, mortgage loan account, or debit card.

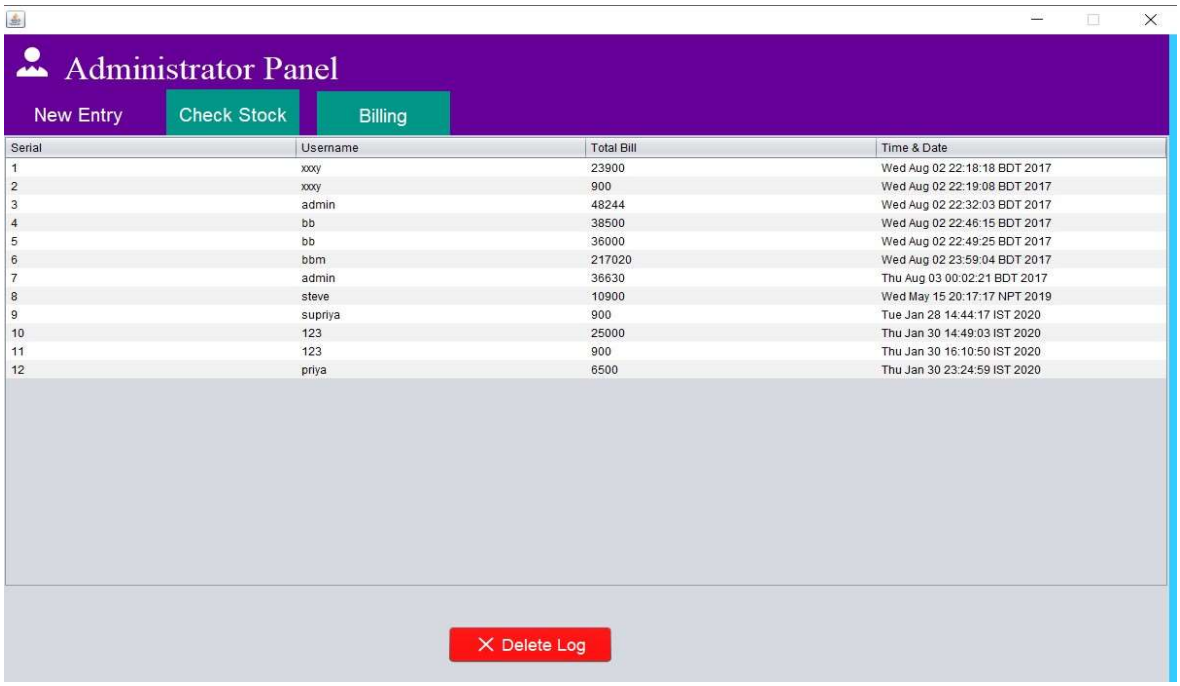

Fig 9.10: Billing Page

CHAPTER 10
CONCLUSION AND FUTURE SCOPE

The project entitled **Local shopping Zone** was completed successfully.

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application for purchasing items from a local shop.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using xml, usage of responsive templates, designing of android applications, and management of database using sqlite. The entire system is secured. Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications.

There is a scope for further development in our project to a great extend. A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limited us.

Future Scope:

- To add home delivery option
- To provide exciting offers so that they would be beneficial to customers as well as store owners.

APPENDIX

Source Code for Local Shopping Zone:

MainPage.java

```
package project;
import java.awt.Color;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import static project.MainPage.total;
public class AdminPanel extends javax.swing.JFrame {
    boolean imageChooser = false;
    String path;

    public AdminPanel() {
        initComponents();
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setResizable(false);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel3 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jPanel4 = new javax.swing.JPanel();
        updateButton = new javax.swing.JButton();
        checkStockButton = new javax.swing.JButton();
        billingButton = new javax.swing.JButton();
        cardAdminParentLayout = new javax.swing.JPanel();
        updatePanel = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jLabel4 = new javax.swing.JLabel();
        mBrand = new javax.swing.JTextField();
        mModel = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        mQty = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        mPhotoPath = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        mCat = new javax.swing.JComboBox<>();
        jScrollPane1 = new javax.swing.JScrollPane();
        mDescription = new javax.swing.JTextArea();
        selectedPhoto = new javax.swing.JLabel();
        photoSelection = new javax.swing.JButton();
        mPrice = new javax.swing.JTextField();
        LogInButton = new javax.swing.JButton();
```

```
        checkStockPanel = new javax.swing.JPanel();
        jScrollPane3 = new javax.swing.JScrollPane();
        checkStockTable = new javax.swing.JTable();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        stockUpdateModel = new javax.swing.JTextField();
        stockStatusCat = new javax.swing.JComboBox<>();
        updateStockButton = new javax.swing.JButton();
        delEntry = new javax.swing.JButton();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        delEntryModel = new javax.swing.JTextField();
        delUpdateCat = new javax.swing.JComboBox<>();
        jLabel14 = new javax.swing.JLabel();
        stockUpdateSetQty = new javax.swing.JTextField();
        billingPanel = new javax.swing.JPanel();
        jScrollPane2 = new javax.swing.JScrollPane();
        billingTable = new javax.swing.JTable();
        delEntry1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel3.setBackground(new java.awt.Color(51, 204, 255));
        jPanel3.setMinimumSize(new java.awt.Dimension(1199, 699));
        jPanel3.setPreferredSize(new java.awt.Dimension(1199, 699));
        jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel2.setBackground(new java.awt.Color(102, 0, 153));
        jPanel2.setPreferredSize(new java.awt.Dimension(1200, 72));

        jLabel1.setFont(new java.awt.Font("Times New Roman", 0, 36)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(255, 255, 255));
        jLabel1.setText("Administrator Panel");
        jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel1MouseClicked(evt);
            }
        });

        jLabel2.setForeground(new java.awt.Color(51, 204, 255));
        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Admin_48px.png"))); //
NOI18N

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 455,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(678, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addContainerGap(16, Short.MAX_VALUE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        jPanel3.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1200, 60));
```

```
jPanel4.setBackground(new java.awt.Color(102, 0, 153));

updateButton.setBackground(new java.awt.Color(102, 0, 153));
updateButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
updateButton.setForeground(new java.awt.Color(255, 255, 255));
updateButton.setText("New Entry");
updateButton.setToolTipText("Enter new product");
updateButton.setBorder(null);
updateButton.setBorderPainted(false);
updateButton.setContentAreaFilled(false);
updateButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
updateButton.setOpaque(true);
updateButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        updateButtonMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        updateButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        updateButtonMouseExited(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        updateButtonMousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        updateButtonMouseReleased(evt);
    }
});
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});

checkStockButton.setBackground(new java.awt.Color(102, 0, 153));
checkStockButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
checkStockButton.setForeground(new java.awt.Color(255, 255, 255));
checkStockButton.setText("Check Stock");
checkStockButton.setToolTipText("Stock status");
checkStockButton.setBorder(null);
checkStockButton.setBorderPainted(false);
checkStockButton.setContentAreaFilled(false);
checkStockButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
checkStockButton.setOpaque(true);
checkStockButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseExited(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {

        checkStockButtonMousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseReleased(evt);
    }
});
checkStockButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        checkStockButtonActionPerformed(evt);
    }
});
billingButton.setBackground(new java.awt.Color(102, 0, 153));
billingButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
billingButton.setForeground(new java.awt.Color(255, 255, 255));
billingButton.setText("Billing");
```

```
        billingButton.setToolTipText("Total billings ");
        billingButton.setBorder(null);
        billingButton.setBorderPainted(false);
        billingButton.setContentAreaFilled(false);
        billingButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        billingButton.setOpaque(true);
        billingButton.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                billingButtonMouseClicked(evt);
            }
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                billingButtonMouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                billingButtonMouseExited(evt);
            }
            public void mousePressed(java.awt.event.MouseEvent evt) {
                billingButtonMousePressed(evt);
            }
            public void mouseReleased(java.awt.event.MouseEvent evt) {
                billingButtonMouseReleased(evt);
            }
        });
        billingButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                billingButtonActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(34, 34, 34)
                .addComponent(checkStockButton, javax.swing.GroupLayout.PREFERRED_SIZE,
137, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(billingButton, javax.swing.GroupLayout.PREFERRED_SIZE, 137,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(740, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILI
NG)
                    .addComponent(billingButton, javax.swing.GroupLayout.PREFERRED_SIZE,
52, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
                        .addComponent(checkStockButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 54,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE,
54, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(0, 0, Short.MAX_VALUE))
        );

        jPanel3.add(jPanel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(-10, 60, 1210,
50));

        cardAdminParentLayout.setBackground(new java.awt.Color(51, 204, 255));
        cardAdminParentLayout.setLayout(new java.awt.CardLayout());

        updatePanel.setBackground(new java.awt.Color(51, 204, 255));
        updatePanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel3.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
```

```
        jLabel3.setForeground(new java.awt.Color(0, 77, 64));
        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel3.setText("Product Details");
        updatePanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 24, 1200,
-1));

        jSeparator1.setBackground(new java.awt.Color(0, 77, 64));
        jSeparator1.setForeground(new java.awt.Color(0, 77, 64));
        jSeparator1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jSeparator1.setOpaque(true);
        jSeparator1.setPreferredSize(new java.awt.Dimension(0, 1));
        updatePanel.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(500,
51, 200, -1));

        jLabel4.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(0, 77, 64));
        jLabel4.setText("Brand:");
        updatePanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 110,
104, -1));

        mBrand.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mBrand.setForeground(new java.awt.Color(0, 77, 64));
        mBrand.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mBrand.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mBrandActionPerformed(evt);
            }
        });
        updatePanel.add(mBrand, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 110,
362, -1));

        mModel.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mModel.setForeground(new java.awt.Color(0, 77, 64));
        mModel.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mModelActionPerformed(evt);
            }
        });
        updatePanel.add(mModel, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 150,
362, -1));

        jLabel5.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 77, 64));
        jLabel5.setText("Model:");
        updatePanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 150,
104, -1));

        mQty.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mQty.setForeground(new java.awt.Color(0, 77, 64));
        mQty.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 77,
64)));
        mQty.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mQtyActionPerformed(evt);
            }
        });
        updatePanel.add(mQty, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 230,
110, -1));

        jLabel6.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel6.setForeground(new java.awt.Color(0, 77, 64));
        jLabel6.setText("Category:");
        updatePanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 270,
104, -1));

        mPhotoPath.setEditable(false);
        mPhotoPath.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mPhotoPath.setForeground(new java.awt.Color(0, 77, 64));
        mPhotoPath.setText("Choose photo:");
```

```
        mPhotoPath.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 77, 64)));
        mPhotoPath.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mPhotoPathActionPerformed(evt);
            }
        });
        updatePanel.add(mPhotoPath, new org.netbeans.lib.awtextra.AbsoluteConstraints(870,
340, 210, 30));

        jLabel7.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel7.setForeground(new java.awt.Color(0, 77, 64));
        jLabel7.setText("Price:");
        updatePanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 190,
104, -1));

        jLabel8.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel8.setForeground(new java.awt.Color(0, 77, 64));
        jLabel8.setText("Quantity:");
        updatePanel.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 230,
104, -1));

        jLabel9.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel9.setForeground(new java.awt.Color(0, 77, 64));
        jLabel9.setText("Product Description:");
        updatePanel.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 310,
140, -1));

        mCat.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));
        updatePanel.add(mCat, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 270, -1, -
1));

        mDescription.setColumns(20);
        mDescription.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mDescription.setLineWrap(true);
        mDescription.setRows(5);
        mDescription.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 77,
64), 1, true));
        jScrollPane1.setViewportView(mDescription);

        updatePanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(350,
320, 370, 190));

        selectedPhoto.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        selectedPhoto.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Image_100px.png"))); //
NOI18N
        selectedPhoto.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 77,
64), 1, true));
        selectedPhoto.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        updatePanel.add(selectedPhoto, new org.netbeans.lib.awtextra.AbsoluteConstraints(880,
100, 230, 210));

        photoSelection.setBackground(new java.awt.Color(0, 77, 64));
        photoSelection.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        photoSelection.setForeground(new java.awt.Color(51, 204, 255));
        photoSelection.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Camera_20px.png"))); //
NOI18N
        photoSelection.setBorder(null);
        photoSelection.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        photoSelection.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                photoSelectionActionPerformed(evt);
            }
        });
        updatePanel.add(photoSelection, new
org.netbeans.lib.awtextra.AbsoluteConstraints(1080, 340, 50, 30));

        mPrice.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
```

```
        mPrice.setForeground(new java.awt.Color(0, 77, 64));
        mPrice.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mPrice.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mPriceActionPerformed(evt);
            }
        });
        updatePanel.add(mPrice, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 190,
180, -1));

        LogInButton.setBackground(new java.awt.Color(0, 77, 64));
        LogInButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
        LogInButton.setForeground(new java.awt.Color(51, 204, 255));
        LogInButton.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Database_24px.png")));
// NOI18N
        LogInButton.setText("Update");
        LogInButton.setBorder(null);
        LogInButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        LogInButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                LogInButtonActionPerformed(evt);
            }
        });
        updatePanel.add(LogInButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(1000,
460, 130, 50));

        cardAdminParentLayout.add(updatePanel, "card3");

        checkStockPanel.setBackground(new java.awt.Color(51, 204, 255));

        checkStockTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Brand Name", "Model", "Stock", "Category"
            }
        ));
        checkStockTable.setEnabled(false);
        checkStockTable.setRowHeight(20);
        jScrollPane3.setViewportView(checkStockTable);

        jLabel10.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel10.setForeground(new java.awt.Color(0, 77, 64));
        jLabel10.setText("Enter model:");

        jLabel11.setFont(new java.awt.Font("Roboto", 1, 18)); // NOI18N
        jLabel11.setForeground(new java.awt.Color(0, 77, 64));
        jLabel11.setText("Add to Stock:");

        stockUpdateModel.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockUpdateModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                stockUpdateModelActionPerformed(evt);
            }
        });

        stockStatusCat.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockStatusCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));

        updateStockButton.setBackground(new java.awt.Color(0, 77, 64));
        updateStockButton.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        updateStockButton.setForeground(new java.awt.Color(51, 204, 255));
        updateStockButton.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Up_24px.png"))); // NOI18N
        updateStockButton.setText("Update Stock");
        updateStockButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                updateStockButtonActionPerformed(evt);
```

```
            }
        });

        delEntry.setBackground(new java.awt.Color(255, 0, 0));
        delEntry.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntry.setForeground(new java.awt.Color(51, 204, 255));
        delEntry.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Delete_24px.png"))); //
NOI18N
        delEntry.setText("Delete Entry");
        delEntry.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntryActionPerformed(evt);
            }
        });

        jLabel12.setFont(new java.awt.Font("Roboto", 1, 18)); // NOI18N
        jLabel12.setForeground(new java.awt.Color(255, 0, 0));
        jLabel12.setText("Delete Entry:");

        jLabel13.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel13.setForeground(new java.awt.Color(255, 0, 0));
        jLabel13.setText("Enter model:");

        delEntryModel.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntryModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntryModelActionPerformed(evt);
            }
        });

        delUpdateCat.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delUpdateCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));

        jLabel14.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel14.setForeground(new java.awt.Color(0, 77, 64));
        jLabel14.setText("Set Quantity:");

        stockUpdateSetQty.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockUpdateSetQty.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                stockUpdateSetQtyActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout checkStockPanelLayout = new
javax.swing.GroupLayout(checkStockPanel);
        checkStockPanel.setLayout(checkStockPanelLayout);
        checkStockPanelLayout.setHorizontalGroup(

checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addComponent(jScrollPane3)
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addGroup(checkStockPanelLayout.createSequentialGroup()
                        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(delEntryModel,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(28, 28, 28)
                        .addComponent(delUpdateCat, javax.swing.GroupLayout.PREFERRED_SIZE,
145, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addGap(26, 26, 26)
                .addComponent(delEntry, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(stockUpdateModel,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(28, 28, 28)
                .addComponent(stockStatusCat,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(stockUpdateSetQty,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(26, 26, 26)
            .addComponent(updateStockButton)
            .addContainerGap(77, Short.MAX_VALUE))
        );
        checkStockPanelLayout.setVerticalGroup(

checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addGap(38, 38, 38)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(stockUpdateModel,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
checkStockPanelLayout.createSequentialGroup()
                .addGap(1, 1, 1)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(stockStatusCat,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(stockUpdateSetQty,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(updateStockButton))))
            .addGap(27, 27, 27)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
```

```
.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                    .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(delEntryModel,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
                    .addComponent(delUpdateCat, javax.swing.GroupLayout.PREFERRED_SIZE,
31, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(delEntry)))
            .addGap(0, 36, Short.MAX_VALUE))
        );

        cardAdminParentLayout.add(checkStockPanel, "card4");

        billingTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Serial", "Username", "Total Bill", "Time & Date"
            }
        ));
        jScrollPane2.setViewportView(billingTable);

        delEntry1.setBackground(new java.awt.Color(255, 0, 0));
        delEntry1.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntry1.setForeground(new java.awt.Color(255, 255, 255));
        delEntry1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Delete_24px.png"))); //
NOI18N
        delEntry1.setText("Delete Log");
        delEntry1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntry1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout billingPanelLayout = new
javax.swing.GroupLayout(billingPanel);
        billingPanel.setLayout(billingPanelLayout);
        billingPanelLayout.setHorizontalGroup(

billingPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 1200,
Short.MAX_VALUE)
            .addGroup(billingPanelLayout.createSequentialGroup()
                .addGap(457, 457, 457)
                .addComponent(delEntry1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        billingPanelLayout.setVerticalGroup(

billingPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(billingPanelLayout.createSequentialGroup()
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 493,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 41,
Short.MAX_VALUE)
                .addComponent(delEntry1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(24, 24, 24))
        );
```

```
        cardAdminParentLayout.add(billingPanel, "card2");

        jPanel3.add(cardAdminParentLayout, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 110, 1200, 600));

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void billingButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_billingButtonActionPerformed
        // TODO add your handling code here:
            cardAdminParentLayout.removeAll();
            cardAdminParentLayout.add(billingPanel);
            cardAdminParentLayout.repaint();
            cardAdminParentLayout.revalidate();

            DefaultTableModel model =  (DefaultTableModel) billingTable.getModel();
            model.setRowCount(0);

            Object[] data = new Object[4];
            ArrayList<BillObject> list = new ArrayList<>();
            list = BillingDB.billlings();

        for(int i=0; i<list.size(); i++){
            data[0] = i+1;
            data[1] = list.get(i).getUname();
            data[2] = list.get(i).getBill();
            data[3] = list.get(i).getDate();

            model.addRow(data);
            billingTable.setRowHeight(20);
        }
    }//GEN-LAST:event_billingButtonActionPerformed

    private void billingButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseReleased
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_billingButtonMouseReleased
    private void billingButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMousePressed
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_billingButtonMousePressed

    private void billingButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseExited
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_billingButtonMouseExited

    private void billingButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseEntered
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_billingButtonMouseEntered

    private void billingButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseClicked
```

```
    }//GEN-LAST:event_billingButtonMouseClicked

    private void checkStockButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_checkStockButtonActionPerformed
        cardAdminParentLayout.removeAll();
        cardAdminParentLayout.add(checkStockPanel);
        cardAdminParentLayout.repaint();
        cardAdminParentLayout.revalidate();

        DefaultTableModel model =  (DefaultTableModel) checkStockTable.getModel();
        model.setRowCount(0);

        ArrayList<ProductList> list = ElectronicsDB.checkStock();

        Object data[] = new Object[4];

      for(int i=0; i<list.size(); i++){
        data[0] = list.get(i).getBrand();
        data[1] = list.get(i).getModel();
        data[2] = list.get(i).getQty();
        data[3] = "Electronics";

        model.addRow(data);
      }

      list.clear();

      list = MobileDB.checkStock();

      for(int i=0; i<list.size(); i++){
        data[0] = list.get(i).getBrand();
        data[1] = list.get(i).getModel();
        data[2] = list.get(i).getQty();
        data[3] = "Mobiles";

        model.addRow(data);
      }

      list.clear();

      list = KidsDB.checkStock();

      for(int i=0; i<list.size(); i++){
        data[0] = list.get(i).getBrand();
        data[1] = list.get(i).getModel();
        data[2] = list.get(i).getQty();
        data[3] = "Kids";

        model.addRow(data);
      }


    }//GEN-LAST:event_checkStockButtonActionPerformed

    private void checkStockButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseReleased
      // TODO add your handling code here:
      checkStockButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_checkStockButtonMouseReleased

    private void checkStockButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMousePressed
      // TODO add your handling code here:
      checkStockButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_checkStockButtonMousePressed

    private void checkStockButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseExited
      // TODO add your handling code here:
      checkStockButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_checkStockButtonMouseExited
```

```
    private void checkStockButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseEntered
        // TODO add your handling code here:
        checkStockButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_checkStockButtonMouseEntered

    private void checkStockButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_checkStockButtonMouseClicked

    private void updateButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_updateButtonActionPerformed
            cardAdminParentLayout.removeAll();
            cardAdminParentLayout.add(updatePanel);
            cardAdminParentLayout.repaint();
            cardAdminParentLayout.revalidate();
    }//GEN-LAST:event_updateButtonActionPerformed

    private void updateButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseReleased
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_updateButtonMouseReleased

    private void updateButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMousePressed
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_updateButtonMousePressed

    private void updateButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseExited
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_updateButtonMouseExited

    private void updateButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseEntered
        // TODO add your handling code here:;

        updateButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_updateButtonMouseEntered

    private void updateButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_updateButtonMouseClicked

    private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jLabel1MouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_jLabel1MouseClicked

    private void mBrandActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mBrandActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mBrandActionPerformed

    private void mModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mModelActionPerformed

    private void mQtyActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mQtyActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mQtyActionPerformed

    private void mPhotoPathActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mPhotoPathActionPerformed
```

```
        // TODO add your handling code here:
    }//GEN-LAST:event_mPhotoPathActionPerformed

    private void photoSelectionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_photoSelectionActionPerformed
        // TODO add your handling code here:
        JFileChooser fc = new JFileChooser();
        fc.showOpenDialog(this);
        File selectedImage = fc.getSelectedFile();
        path = selectedImage.getName();

        try {
            Image img = ImageIO.read(selectedImage);
            mPhotoPath.setText(path);
            selectedPhoto.setIcon(new
ImageIcon(img.getScaledInstance(selectedPhoto.getWidth(),
                    selectedPhoto.getHeight(), Image.SCALE_SMOOTH)));
        } catch (IOException ex) {
            Logger.getLogger(AdminPanel.class.getName()).log(Level.SEVERE, null, ex);
        }


    }//GEN-LAST:event_photoSelectionActionPerformed

    private void mPriceActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mPriceActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mPriceActionPerformed

    private void LogInButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_LogInButtonActionPerformed
        // TODO add your handling code here:
        try{
            if(mCat.getSelectedItem().equals("Mobile")){
            if(!path.equals(""))
                MobileDB.insertIntoMobileDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }

        else if(mCat.getSelectedItem().equals("Electronics")){
            if(!path.equals(""))
                ElectronicsDB.insertIntoElectronicsDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }
        else if(mCat.getSelectedItem().equals("Kids")){
            if(!path.equals(""))
                KidsDB.insertIntoKidsDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }

        mBrand.setText("");
        mModel.setText("");
        mQty.setText("");
        mPrice.setText("");
        mDescription.setText("");
        mPhotoPath.setText("");
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
            mBrand.setText("");
            mModel.setText("");
            mQty.setText("");
            mPrice.setText("");
            mDescription.setText("");
            mPhotoPath.setText("");
        }

    }//GEN-LAST:event_LogInButtonActionPerformed
```

```
    private void stockUpdateModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_stockUpdateModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_stockUpdateModelActionPerformed

    private void delEntryModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntryModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_delEntryModelActionPerformed

    private void updateStockButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_updateStockButtonActionPerformed
        // TODO add your handling code here:
        try{
            String model = stockUpdateModel.getText();
            int qty = Integer.parseInt(stockUpdateSetQty.getText());

            if(stockStatusCat.getSelectedItem().equals("Electronics")){
                ElectronicsDB.flag = true;
                ElectronicsDB.updateElectronicsDB(model, qty);
            }

            else if(stockStatusCat.getSelectedItem().equals("Mobiles")){
                MobileDB.flag = true;
                MobileDB.updateMobileDB(model, qty);
            }

            else if(stockStatusCat.getSelectedItem().equals("Kids")){
                KidsDB.flag = true;
                KidsDB.updateKidsDB(model, qty);
            }

            stockUpdateModel.setText("");
            stockUpdateSetQty.setText("");

        }catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }


    }//GEN-LAST:event_updateStockButtonActionPerformed

    private void stockUpdateSetQtyActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_stockUpdateSetQtyActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_stockUpdateSetQtyActionPerformed

    private void delEntryActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntryActionPerformed
        // TODO add your handling code here:
        try{
            String del = delEntryModel.getText();

        if(delUpdateCat.getSelectedItem().equals("Electronics")){
            ElectronicsDB.delete(del);
        }
        else if(delUpdateCat.getSelectedItem().equals("Mobiles")){
            MobileDB.delete(del);
        }
        else if(delUpdateCat.getSelectedItem().equals("Kids")){
            KidsDB.delete(del);
        }
        delEntryModel.setText("");
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }//GEN-LAST:event_delEntryActionPerformed

    private void delEntry1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntry1ActionPerformed
        // TODO add your handling code here:
        BillingDB.deleteBillings();
```

```
    }//GEN-LAST:event_delEntry1ActionPerformed

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton LogInButton;
    private javax.swing.JButton billingButton;
    private javax.swing.JPanel billingPanel;
    private javax.swing.JTable billingTable;
    private javax.swing.JPanel cardAdminParentLayout;
    private javax.swing.JButton checkStockButton;
    private javax.swing.JPanel checkStockPanel;
    private javax.swing.JTable checkStockTable;
    private javax.swing.JButton delEntry;
    private javax.swing.JButton delEntry1;
    private javax.swing.JTextField delEntryModel;
    private javax.swing.JComboBox<String> delUpdateCat;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JTextField mBrand;
    private javax.swing.JComboBox<String> mCat;
    private javax.swing.JTextArea mDescription;
    private javax.swing.JTextField mModel;
    private javax.swing.JTextField mPhotoPath;
    private javax.swing.JTextField mPrice;
    private javax.swing.JTextField mQty;
    private javax.swing.JButton photoSelection;
    private javax.swing.JLabel selectedPhoto;
    private javax.swing.JComboBox<String> stockStatusCat;
    private javax.swing.JTextField stockUpdateModel;
    private javax.swing.JTextField stockUpdateSetQty;
    private javax.swing.JButton updateButton;
    private javax.swing.JPanel updatePanel;
    private javax.swing.JButton updateStockButton;
    // End of variables declaration//GEN-END:variables
}
```

AdminPanel.java

```
package project;
import java.awt.Color;
import java.awt.Image;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import static project.MainPage.total;

/**
 *
 * @author Void
 */
public class AdminPanel extends javax.swing.JFrame {

    /**
     * Creates new form AdminPanel
     */
    boolean imageChooser = false;
    String path;

    public AdminPanel() {
        initComponents();
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setResizable(false);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        jPanel3 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jPanel4 = new javax.swing.JPanel();
        updateButton = new javax.swing.JButton();
        checkStockButton = new javax.swing.JButton();
        billingButton = new javax.swing.JButton();
        cardAdminParentLayout = new javax.swing.JPanel();
        updatePanel = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jSeparator1 = new javax.swing.JSeparator();
        jLabel4 = new javax.swing.JLabel();
        mBrand = new javax.swing.JTextField();
        mModel = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();
        mQty = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        mPhotoPath = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        mCat = new javax.swing.JComboBox<>();
        jScrollPane1 = new javax.swing.JScrollPane();
        mDescription = new javax.swing.JTextArea();
        selectedPhoto = new javax.swing.JLabel();
```

```
        photoSelection = new javax.swing.JButton();
        mPrice = new javax.swing.JTextField();
        LogInButton = new javax.swing.JButton();
        checkStockPanel = new javax.swing.JPanel();
        jScrollPane3 = new javax.swing.JScrollPane();
        checkStockTable = new javax.swing.JTable();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        stockUpdateModel = new javax.swing.JTextField();
        stockStatusCat = new javax.swing.JComboBox<>();
        updateStockButton = new javax.swing.JButton();
        delEntry = new javax.swing.JButton();
        jLabel12 = new javax.swing.JLabel();
        jLabel13 = new javax.swing.JLabel();
        delEntryModel = new javax.swing.JTextField();
        delUpdateCat = new javax.swing.JComboBox<>();
        jLabel14 = new javax.swing.JLabel();
        stockUpdateSetQty = new javax.swing.JTextField();
        billingPanel = new javax.swing.JPanel();
        jScrollPane2 = new javax.swing.JScrollPane();
        billingTable = new javax.swing.JTable();
        delEntry1 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jPanel3.setBackground(new java.awt.Color(51, 204, 255));
        jPanel3.setMinimumSize(new java.awt.Dimension(1199, 699));
        jPanel3.setPreferredSize(new java.awt.Dimension(1199, 699));
        jPanel3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel2.setBackground(new java.awt.Color(102, 0, 153));
        jPanel2.setPreferredSize(new java.awt.Dimension(1200, 72));

        jLabel1.setFont(new java.awt.Font("Times New Roman", 0, 36)); // NOI18N
        jLabel1.setForeground(new java.awt.Color(255, 255, 255));
        jLabel1.setText("Administrator Panel");
        jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel1MouseClicked(evt);
            }
        });

        jLabel2.setForeground(new java.awt.Color(51, 204, 255));
        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Admin_48px.png"))); //
NOI18N

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 51,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 455,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(678, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addContainerGap(16, Short.MAX_VALUE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
```

```
jPanel3.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 1200, 60));

jPanel4.setBackground(new java.awt.Color(102, 0, 153));

updateButton.setBackground(new java.awt.Color(102, 0, 153));
updateButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
updateButton.setForeground(new java.awt.Color(255, 255, 255));
updateButton.setText("New Entry");
updateButton.setToolTipText("Enter new product");
updateButton.setBorder(null);
updateButton.setBorderPainted(false);
updateButton.setContentAreaFilled(false);
updateButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
updateButton.setOpaque(true);
updateButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        updateButtonMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        updateButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        updateButtonMouseExited(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        updateButtonMousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        updateButtonMouseReleased(evt);
    }
});
updateButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        updateButtonActionPerformed(evt);
    }
});

checkStockButton.setBackground(new java.awt.Color(102, 0, 153));
checkStockButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
checkStockButton.setForeground(new java.awt.Color(255, 255, 255));
checkStockButton.setText("Check Stock");
checkStockButton.setToolTipText("Stock status");
checkStockButton.setBorder(null);
checkStockButton.setBorderPainted(false);
checkStockButton.setContentAreaFilled(false);
checkStockButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
checkStockButton.setOpaque(true);
checkStockButton.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseClicked(evt);
    }
    public void mouseEntered(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseEntered(evt);
    }
    public void mouseExited(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseExited(evt);
    }
    public void mousePressed(java.awt.event.MouseEvent evt) {
        checkStockButtonMousePressed(evt);
    }
    public void mouseReleased(java.awt.event.MouseEvent evt) {
        checkStockButtonMouseReleased(evt);
    }
});
checkStockButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        checkStockButtonActionPerformed(evt);
    }
});

billingButton.setBackground(new java.awt.Color(102, 0, 153));
```

```
        billingButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
        billingButton.setForeground(new java.awt.Color(255, 255, 255));
        billingButton.setText("Billing");
        billingButton.setToolTipText("Total billings ");
        billingButton.setBorder(null);
        billingButton.setBorderPainted(false);
        billingButton.setContentAreaFilled(false);
        billingButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        billingButton.setOpaque(true);
        billingButton.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                billingButtonMouseClicked(evt);
            }
            public void mouseEntered(java.awt.event.MouseEvent evt) {
                billingButtonMouseEntered(evt);
            }
            public void mouseExited(java.awt.event.MouseEvent evt) {
                billingButtonMouseExited(evt);
            }
            public void mousePressed(java.awt.event.MouseEvent evt) {
                billingButtonMousePressed(evt);
            }
            public void mouseReleased(java.awt.event.MouseEvent evt) {
                billingButtonMouseReleased(evt);
            }
        });
        billingButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                billingButtonActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE, 113,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(34, 34, 34)
                .addComponent(checkStockButton, javax.swing.GroupLayout.PREFERRED_SIZE,
137, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(billingButton, javax.swing.GroupLayout.PREFERRED_SIZE, 137,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(740, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILI
NG)
                    .addComponent(billingButton, javax.swing.GroupLayout.PREFERRED_SIZE,
52, javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
                    .addComponent(checkStockButton,
javax.swing.GroupLayout.PREFERRED_SIZE, 54,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(updateButton, javax.swing.GroupLayout.PREFERRED_SIZE,
54, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGap(0, 0, Short.MAX_VALUE))
        );

        jPanel3.add(jPanel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(-10, 60, 1210,
50));

        cardAdminParentLayout.setBackground(new java.awt.Color(51, 204, 255));
        cardAdminParentLayout.setLayout(new java.awt.CardLayout());
```

```
        updatePanel.setBackground(new java.awt.Color(51, 204, 255));
        updatePanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel3.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
        jLabel3.setForeground(new java.awt.Color(0, 77, 64));
        jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel3.setText("Product Details");
        updatePanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 24, 1200,
-1));

        jSeparator1.setBackground(new java.awt.Color(0, 77, 64));
        jSeparator1.setForeground(new java.awt.Color(0, 77, 64));
        jSeparator1.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jSeparator1.setOpaque(true);
        jSeparator1.setPreferredSize(new java.awt.Dimension(0, 1));
        updatePanel.add(jSeparator1, new org.netbeans.lib.awtextra.AbsoluteConstraints(500,
51, 200, -1));

        jLabel4.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel4.setForeground(new java.awt.Color(0, 77, 64));
        jLabel4.setText("Brand:");
        updatePanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 110,
104, -1));

        mBrand.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mBrand.setForeground(new java.awt.Color(0, 77, 64));
        mBrand.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mBrand.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mBrandActionPerformed(evt);
            }
        });
        updatePanel.add(mBrand, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 110,
362, -1));

        mModel.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mModel.setForeground(new java.awt.Color(0, 77, 64));
        mModel.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mModelActionPerformed(evt);
            }
        });
        updatePanel.add(mModel, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 150,
362, -1));

        jLabel5.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 77, 64));
        jLabel5.setText("Model:");
        updatePanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 150,
104, -1));

        mQty.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mQty.setForeground(new java.awt.Color(0, 77, 64));
        mQty.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 77,
64)));
        mQty.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mQtyActionPerformed(evt);
            }
        });
        updatePanel.add(mQty, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 230,
110, -1));

        jLabel6.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel6.setForeground(new java.awt.Color(0, 77, 64));
        jLabel6.setText("Category:");
        updatePanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 270,
104, -1));
```

```
        mPhotoPath.setEditable(false);
        mPhotoPath.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mPhotoPath.setForeground(new java.awt.Color(0, 77, 64));
        mPhotoPath.setText("Choose photo:");
        mPhotoPath.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 77, 64)));
        mPhotoPath.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mPhotoPathActionPerformed(evt);
            }
        });
        updatePanel.add(mPhotoPath, new org.netbeans.lib.awtextra.AbsoluteConstraints(870,
340, 210, 30));

        jLabel7.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel7.setForeground(new java.awt.Color(0, 77, 64));
        jLabel7.setText("Price:");
        updatePanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 190,
104, -1));

        jLabel8.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel8.setForeground(new java.awt.Color(0, 77, 64));
        jLabel8.setText("Quantity:");
        updatePanel.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 230,
104, -1));

        jLabel9.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        jLabel9.setForeground(new java.awt.Color(0, 77, 64));
        jLabel9.setText("Product Description:");
        updatePanel.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(200, 310,
140, -1));

        mCat.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));
        updatePanel.add(mCat, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 270, -1, -
1));

        mDescription.setColumns(20);
        mDescription.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mDescription.setLineWrap(true);
        mDescription.setRows(5);
        mDescription.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 77,
64), 1, true));
        jScrollPane1.setViewportView(mDescription);

        updatePanel.add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(350,
320, 370, 190));

        selectedPhoto.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        selectedPhoto.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Image_100px.png"))); //
NOI18N
        selectedPhoto.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 77,
64), 1, true));
        selectedPhoto.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
        updatePanel.add(selectedPhoto, new org.netbeans.lib.awtextra.AbsoluteConstraints(880,
100, 230, 210));

        photoSelection.setBackground(new java.awt.Color(0, 77, 64));
        photoSelection.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        photoSelection.setForeground(new java.awt.Color(51, 204, 255));
        photoSelection.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Camera_20px.png"))); //
NOI18N
        photoSelection.setBorder(null);
        photoSelection.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        photoSelection.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                photoSelectionActionPerformed(evt);
            }
```

```
        });
        updatePanel.add(photoSelection, new
org.netbeans.lib.awtextra.AbsoluteConstraints(1080, 340, 50, 30));

        mPrice.setFont(new java.awt.Font("Roboto", 0, 14)); // NOI18N
        mPrice.setForeground(new java.awt.Color(0, 77, 64));
        mPrice.setBorder(javax.swing.BorderFactory.createLineBorder(new java.awt.Color(0,
77, 64)));
        mPrice.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                mPriceActionPerformed(evt);
            }
        });
        updatePanel.add(mPrice, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 190,
180, -1));

        LogInButton.setBackground(new java.awt.Color(0, 77, 64));
        LogInButton.setFont(new java.awt.Font("Roboto", 0, 20)); // NOI18N
        LogInButton.setForeground(new java.awt.Color(51, 204, 255));
        LogInButton.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Add_Database_24px.png")));
// NOI18N
        LogInButton.setText("Update");
        LogInButton.setBorder(null);
        LogInButton.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND_CURSOR));
        LogInButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                LogInButtonActionPerformed(evt);
            }
        });
        updatePanel.add(LogInButton, new org.netbeans.lib.awtextra.AbsoluteConstraints(1000,
460, 130, 50));

        cardAdminParentLayout.add(updatePanel, "card3");

        checkStockPanel.setBackground(new java.awt.Color(51, 204, 255));

        checkStockTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Brand Name", "Model", "Stock", "Category"
            }
        ));
        checkStockTable.setEnabled(false);
        checkStockTable.setRowHeight(20);
        jScrollPane3.setViewportView(checkStockTable);

        jLabel10.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel10.setForeground(new java.awt.Color(0, 77, 64));
        jLabel10.setText("Enter model:");

        jLabel11.setFont(new java.awt.Font("Roboto", 1, 18)); // NOI18N
        jLabel11.setForeground(new java.awt.Color(0, 77, 64));
        jLabel11.setText("Add to Stock:");

        stockUpdateModel.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockUpdateModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                stockUpdateModelActionPerformed(evt);
            }
        });

        stockStatusCat.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockStatusCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));

        updateStockButton.setBackground(new java.awt.Color(0, 77, 64));
        updateStockButton.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        updateStockButton.setForeground(new java.awt.Color(51, 204, 255));
```

```
        updateStockButton.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Up_24px.png"))); // NOI18N
        updateStockButton.setText("Update Stock");
        updateStockButton.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                updateStockButtonActionPerformed(evt);
            }
        });

        delEntry.setBackground(new java.awt.Color(255, 0, 0));
        delEntry.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntry.setForeground(new java.awt.Color(51, 204, 255));
        delEntry.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Delete_24px.png"))); //
NOI18N
        delEntry.setText("Delete Entry");
        delEntry.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntryActionPerformed(evt);
            }
        });

        jLabel12.setFont(new java.awt.Font("Roboto", 1, 18)); // NOI18N
        jLabel12.setForeground(new java.awt.Color(255, 0, 0));
        jLabel12.setText("Delete Entry:");

        jLabel13.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel13.setForeground(new java.awt.Color(255, 0, 0));
        jLabel13.setText("Enter model:");

        delEntryModel.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntryModel.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntryModelActionPerformed(evt);
            }
        });

        delUpdateCat.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delUpdateCat.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] {
"Southindia", "KLM", "CMR", "MAX" }));

        jLabel14.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        jLabel14.setForeground(new java.awt.Color(0, 77, 64));
        jLabel14.setText("Set Quantity:");

        stockUpdateSetQty.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        stockUpdateSetQty.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                stockUpdateSetQtyActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout checkStockPanelLayout = new
javax.swing.GroupLayout(checkStockPanel);
        checkStockPanel.setLayout(checkStockPanelLayout);
        checkStockPanelLayout.setHorizontalGroup(

checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addComponent(jScrollPane3)
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addContainerGap()

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                    .addGroup(checkStockPanelLayout.createSequentialGroup()
                        .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
```

```
                    .addComponent(delEntryModel,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(28, 28, 28)
                    .addComponent(delUpdateCat, javax.swing.GroupLayout.PREFERRED_SIZE,
145, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(26, 26, 26)
                    .addComponent(delEntry, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(checkStockPanelLayout.createSequentialGroup()
                    .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(stockUpdateModel,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(28, 28, 28)
                    .addComponent(stockStatusCat,
javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)
                    .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(stockUpdateSetQty,
javax.swing.GroupLayout.PREFERRED_SIZE, 164,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(26, 26, 26)
            .addComponent(updateStockButton)
            .addContainerGap(77, Short.MAX_VALUE))
        );
        checkStockPanelLayout.setVerticalGroup(

checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
            .addGroup(checkStockPanelLayout.createSequentialGroup()
                .addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED_SIZE, 436,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
                .addGroup(checkStockPanelLayout.createSequentialGroup()
                    .addGap(38, 38, 38)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(stockUpdateModel,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
checkStockPanelLayout.createSequentialGroup()
                    .addGap(1, 1, 1)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                        .addComponent(stockStatusCat,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel14, javax.swing.GroupLayout.PREFERRED_SIZE,
29, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(stockUpdateSetQty,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(updateStockButton))))
```

```
            .addGap(27, 27, 27)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)

.addGroup(checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment
.BASELINE)
                .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(delEntryModel,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
checkStockPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELIN
E)
                .addComponent(delUpdateCat, javax.swing.GroupLayout.PREFERRED_SIZE,
31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(delEntry)))
            .addGap(0, 36, Short.MAX_VALUE))
        );

        cardAdminParentLayout.add(checkStockPanel, "card4");

        billingTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Serial", "Username", "Total Bill", "Time & Date"
            }
        ));
        jScrollPane2.setViewportView(billingTable);

        delEntry1.setBackground(new java.awt.Color(255, 0, 0));
        delEntry1.setFont(new java.awt.Font("Roboto", 0, 18)); // NOI18N
        delEntry1.setForeground(new java.awt.Color(255, 255, 255));
        delEntry1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icons/icons8_Delete_24px.png"))); //
NOI18N
        delEntry1.setText("Delete Log");
        delEntry1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                delEntry1ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout billingPanelLayout = new
javax.swing.GroupLayout(billingPanel);
        billingPanel.setLayout(billingPanelLayout);
        billingPanelLayout.setHorizontalGroup(

billingPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 1200,
Short.MAX_VALUE)
            .addGroup(billingPanelLayout.createSequentialGroup()
                .addGap(457, 457, 457)
                .addComponent(delEntry1, javax.swing.GroupLayout.PREFERRED_SIZE, 171,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        billingPanelLayout.setVerticalGroup(

billingPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(billingPanelLayout.createSequentialGroup()
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 493,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 41,
Short.MAX_VALUE)
```

```
            .addComponent(delEntry1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(24, 24, 24))
        );

        cardAdminParentLayout.add(billingPanel, "card2");

        jPanel3.add(cardAdminParentLayout, new
org.netbeans.lib.awtextra.AbsoluteConstraints(0, 110, 1200, 600));

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void billingButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_billingButtonActionPerformed
        // TODO add your handling code here:
            cardAdminParentLayout.removeAll();
            cardAdminParentLayout.add(billingPanel);
            cardAdminParentLayout.repaint();
            cardAdminParentLayout.revalidate();

            DefaultTableModel model =  (DefaultTableModel) billingTable.getModel();
            model.setRowCount(0);

            Object[] data = new Object[4];
            ArrayList<BillObject> list = new ArrayList<>();
            list = BillingDB.billlings();

        for(int i=0; i<list.size(); i++){
            data[0] = i+1;
            data[1] = list.get(i).getUname();
            data[2] = list.get(i).getBill();
            data[3] = list.get(i).getDate();

            model.addRow(data);
            billingTable.setRowHeight(20);
        }
    }//GEN-LAST:event_billingButtonActionPerformed

    private void billingButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseReleased
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_billingButtonMouseReleased

    private void billingButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMousePressed
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_billingButtonMousePressed

    private void billingButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseExited
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_billingButtonMouseExited
```

```
    private void billingButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseEntered
        // TODO add your handling code here:
        billingButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_billingButtonMouseEntered

    private void billingButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_billingButtonMouseClicked

    }//GEN-LAST:event_billingButtonMouseClicked

    private void checkStockButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_checkStockButtonActionPerformed
            cardAdminParentLayout.removeAll();
            cardAdminParentLayout.add(checkStockPanel);
            cardAdminParentLayout.repaint();
            cardAdminParentLayout.revalidate();

            DefaultTableModel model =  (DefaultTableModel) checkStockTable.getModel();
            model.setRowCount(0);

            ArrayList<ProductList> list = ElectronicsDB.checkStock();

            Object data[] = new Object[4];

        for(int i=0; i<list.size(); i++){
            data[0] = list.get(i).getBrand();
            data[1] = list.get(i).getModel();
            data[2] = list.get(i).getQty();
            data[3] = "Electronics";

            model.addRow(data);
        }

        list.clear();

        list = MobileDB.checkStock();

        for(int i=0; i<list.size(); i++){
            data[0] = list.get(i).getBrand();
            data[1] = list.get(i).getModel();
            data[2] = list.get(i).getQty();
            data[3] = "Mobiles";

            model.addRow(data);
        }

        list.clear();

        list = KidsDB.checkStock();

        for(int i=0; i<list.size(); i++){
            data[0] = list.get(i).getBrand();
            data[1] = list.get(i).getModel();
            data[2] = list.get(i).getQty();
            data[3] = "Kids";

            model.addRow(data);
        }


    }//GEN-LAST:event_checkStockButtonActionPerformed

    private void checkStockButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseReleased
        // TODO add your handling code here:
        checkStockButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_checkStockButtonMouseReleased

    private void checkStockButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMousePressed
        // TODO add your handling code here:
```

```
        checkStockButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_checkStockButtonMousePressed

    private void checkStockButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseExited
        // TODO add your handling code here:
        checkStockButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_checkStockButtonMouseExited

    private void checkStockButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseEntered
        // TODO add your handling code here:
        checkStockButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_checkStockButtonMouseEntered

    private void checkStockButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_checkStockButtonMouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_checkStockButtonMouseClicked

    private void updateButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_updateButtonActionPerformed
            cardAdminParentLayout.removeAll();
            cardAdminParentLayout.add(updatePanel);
            cardAdminParentLayout.repaint();
            cardAdminParentLayout.revalidate();
    }//GEN-LAST:event_updateButtonActionPerformed

    private void updateButtonMouseReleased(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseReleased
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_updateButtonMouseReleased

    private void updateButtonMousePressed(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMousePressed
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 77, 64));
    }//GEN-LAST:event_updateButtonMousePressed

    private void updateButtonMouseExited(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseExited
        // TODO add your handling code here:
        updateButton.setBackground(new Color(0, 150, 136));
    }//GEN-LAST:event_updateButtonMouseExited

    private void updateButtonMouseEntered(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseEntered
        // TODO add your handling code here:;
        updateButton.setBackground(new Color(0, 137, 123));
    }//GEN-LAST:event_updateButtonMouseEntered

    private void updateButtonMouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_updateButtonMouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_updateButtonMouseClicked

    private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-
FIRST:event_jLabel1MouseClicked
        // TODO add your handling code here:
    }//GEN-LAST:event_jLabel1MouseClicked

    private void mBrandActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mBrandActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mBrandActionPerformed

    private void mModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mModelActionPerformed
```

```
    private void mQtyActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mQtyActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mQtyActionPerformed

    private void mPhotoPathActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mPhotoPathActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mPhotoPathActionPerformed

    private void photoSelectionActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_photoSelectionActionPerformed
        // TODO add your handling code here:
        JFileChooser fc = new JFileChooser();
        fc.showOpenDialog(this);
        File selectedImage = fc.getSelectedFile();
        path = selectedImage.getName();

        try {
            Image img = ImageIO.read(selectedImage);
            mPhotoPath.setText(path);
            selectedPhoto.setIcon(new
ImageIcon(img.getScaledInstance(selectedPhoto.getWidth(),
                    selectedPhoto.getHeight(), Image.SCALE_SMOOTH)));
        } catch (IOException ex) {
            Logger.getLogger(AdminPanel.class.getName()).log(Level.SEVERE, null, ex);
        }


    }//GEN-LAST:event_photoSelectionActionPerformed

    private void mPriceActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_mPriceActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_mPriceActionPerformed

    private void LogInButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_LogInButtonActionPerformed
        // TODO add your handling code here:
        try{
            if(mCat.getSelectedItem().equals("Mobile")){
            if(!path.equals(""))
                MobileDB.insertIntoMobileDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }

        else if(mCat.getSelectedItem().equals("Electronics")){
            if(!path.equals(""))
                ElectronicsDB.insertIntoElectronicsDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }
        else if(mCat.getSelectedItem().equals("Kids")){
            if(!path.equals(""))
                KidsDB.insertIntoKidsDB(mBrand.getText(), mModel.getText(),
Integer.parseInt(mPrice.getText()),
                        Integer.parseInt(mQty.getText()), mDescription.getText(), path);
        }

        mBrand.setText("");
        mModel.setText("");
        mQty.setText("");
        mPrice.setText("");
        mDescription.setText("");
        mPhotoPath.setText("");
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
            mBrand.setText("");
            mModel.setText("");
            mQty.setText("");
            mPrice.setText("");
```

```
            mDescription.setText("");
            mPhotoPath.setText("");
        }


    }//GEN-LAST:event_LogInButtonActionPerformed

    private void stockUpdateModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_stockUpdateModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_stockUpdateModelActionPerformed

    private void delEntryModelActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntryModelActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_delEntryModelActionPerformed

    private void updateStockButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_updateStockButtonActionPerformed
        // TODO add your handling code here:
        try{
            String model = stockUpdateModel.getText();
            int qty = Integer.parseInt(stockUpdateSetQty.getText());

            if(stockStatusCat.getSelectedItem().equals("Electronics")){
                ElectronicsDB.flag = true;
                ElectronicsDB.updateElectronicsDB(model, qty);
            }

            else if(stockStatusCat.getSelectedItem().equals("Mobiles")){
                MobileDB.flag = true;
                MobileDB.updateMobileDB(model, qty);
            }

            else if(stockStatusCat.getSelectedItem().equals("Kids")){
                KidsDB.flag = true;
                KidsDB.updateKidsDB(model, qty);
            }

            stockUpdateModel.setText("");
            stockUpdateSetQty.setText("");

        }catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        }
    }//GEN-LAST:event_updateStockButtonActionPerformed

    private void stockUpdateSetQtyActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_stockUpdateSetQtyActionPerformed
        // TODO add your handling code here:
    }//GEN-LAST:event_stockUpdateSetQtyActionPerformed

    private void delEntryActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntryActionPerformed
        // TODO add your handling code here:
        try{
            String del = delEntryModel.getText();

        if(delUpdateCat.getSelectedItem().equals("Electronics")){
            ElectronicsDB.delete(del);
        }
        else if(delUpdateCat.getSelectedItem().equals("Mobiles")){
            MobileDB.delete(del);
        }
        else if(delUpdateCat.getSelectedItem().equals("Kids")){
            KidsDB.delete(del);
        }
        delEntryModel.setText("");
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }//GEN-LAST:event_delEntryActionPerformed
```

```
    private void delEntry1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_delEntry1ActionPerformed
        // TODO add your handling code here:
        BillingDB.deleteBillings();
    }//GEN-LAST:event_delEntry1ActionPerformed
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton LogInButton;
    private javax.swing.JButton billingButton;
    private javax.swing.JPanel billingPanel;
    private javax.swing.JTable billingTable;
    private javax.swing.JPanel cardAdminParentLayout;
    private javax.swing.JButton checkStockButton;
    private javax.swing.JPanel checkStockPanel;
    private javax.swing.JTable checkStockTable;
    private javax.swing.JButton delEntry;
    private javax.swing.JButton delEntry1;
    private javax.swing.JTextField delEntryModel;
    private javax.swing.JComboBox<String> delUpdateCat;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JPanel jPanel4;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JScrollPane jScrollPane3;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JTextField mBrand;
    private javax.swing.JComboBox<String> mCat;
    private javax.swing.JTextArea mDescription;
    private javax.swing.JTextField mModel;
    private javax.swing.JTextField mPhotoPath;
    private javax.swing.JTextField mPrice;
    private javax.swing.JTextField mQty;
    private javax.swing.JButton photoSelection;
    private javax.swing.JLabel selectedPhoto;
    private javax.swing.JComboBox<String> stockStatusCat;
    private javax.swing.JTextField stockUpdateModel;
    private javax.swing.JTextField stockUpdateSetQty;
    private javax.swing.JButton updateButton;
    private javax.swing.JPanel updatePanel;
    private javax.swing.JButton updateStockButton;
    // End of variables declaration//GEN-END:variables
}
```

REFERENCES:

For Journal Papers:

[1] F.M.A.Erich, C. Amrit M.Daneva,"A qualitative study of devops usage in practice", 28 June 2018.

[2] Kavya Suvarna, "A Conceptual Study on the Impact of Online Shopping Towards Retailers and Consumers International Journal of Advanced Trends in Engineering and Technology (IJATET)." 31 May 2017.

[3] J Doll. (2011, December 28). Drunk online shopping will save this wretched economy. VillageVoice.Retrieved(February14,2012)

[4] Anamika S. (2011). Top 10 benefits of online shopping. HubPages. Retrieved (February 14,2012).

For Conference Papers:

[5] Dr. Shahid Amin Bhat, Keshav Kanzhana, "A review paper on electronic commerce", Conference paper, February 2016.

[6] Vespa, M. (2010). Online shopping: Helping or hurting local economy. Central Illinois NewsCenter.Retrieved.

REFERENCE LINKS:

https://stackoverflow.com

https://w3schools.com

https://javatpoint.com