E-COMMERCE BILL GENERATION

A PROJECT REPORT

Submitted by

GODFREY ASHWANTH [Reg. No.: RA2112704010020]

Under the Guidance of

Dr. ELANGOVAN.G

(Assistant Professor, Department of Data Science and Business Systems)

In partial fulfilment of the Requirements for the Degree

Of

MASTERS OF TECHNOLOGY (INTEGRATED)

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled “**E-COMMERCE BILL GENERATION”** is the bonafide work of **“GODFREY ASHWANTH [Reg. No.: RA2112704010020]”**carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. Elangovan.G	Dr. M Lakshmi
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor
Dept. of Data Science and Business Systems	Dept. of Data Science and Business Systems

Signature of Examiner 1

Signature of Examiner 2

ABSTRACT

The project is based on the shopping bill generation using netbeans(GUI) for java programming. The projects are based on user interaction. The programming is done in such a way that its user friendly and dose not contain any complex work or dose not require the user to know any programming skills. Bill plays one of the most important roles in any shopping areas, be it from grocery store to restaurants. A bill can be generated in multiple ways. A Bill ensures that the customer has brought a product or material from the respective shop. This bill also helps the shop owners to maintain the sales account and purchase frequency. It basically stores all the details of the material soled and helps the owner to know which product is purchased the more. With this bill the customer can know what are the different purchase they have made and also know the cost they have paid for each material and will also know if there was any discount in the product brought. Hence forth this will be use full for both customers and the shop owner.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department**, Dr M. Lakshmi** Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisor**, Dr K Shantha Kumari, Ph.D.,** Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Elangovan.G, Ph.D.**, Data Science and Business Systems, SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentorship. He/She/They provided me with the freedom and support to explore the research topics of my interest. Her/His/Their passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

GODFREY ASHWANTH

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATIONS

JDK	Java Development Kit
JVM	Java Virtual Machine
JRE	Java Runtime Environment
API	Application Programming Interface
NLP	Natural Language Processing

CHAPTER 1
INTRODUCTION

Billing is an important part in the all the shopping medium. This project basically has user interaction that is the employee will have to fill the details of the product purchased, by the customer. Once the login is successfully then the window will shift for main page which will ask for the product details. The project has two phases one the a) Login b) The main page. The process follows in the flow from login to main page. If the purchased product is completed by the customer, then the product details are entered and later it is saved as a bill format which is later saved in required format which is then printed if required. It can be printed for any number of times. One printed bill is given to the customer, as they can have a record of the items they have brought and also know its MRP with any discount if applicable.

1.1 NEED OF THE PROJECT

Billing system is ruling the entire business environment, this bill generation gives a clear detail on what is the product that is purchased and with the actual price of that particular product. This bill gives a record of the selling times and the quantity to the customer as well the shop in charges /owner. The shop owner can have a track of most selling items and will help them in restoring it in the shop for availability so the customer purchases it again when they come for a nextvisit. From the customers view point they on whole can know the products purchased and the price they have paid for eachitem.

1.2 OBJECTIVE OF THE PROJECT

The objective of the project is to give knowledge regarding Billing system and its working with different process flow. This process is simple and user friendly. To provide the systematic billing system to the organization which includes the following points:

- To develop a better automated billing management system.
- To reduce workload of staff.
- To store data in centralized location.

CHAPTER 2

LITERATURE SURVEY

A literature review is a text of a scholarly paper, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature review are the secondary sources and do not report new or original experimental work.

"Design and Implementation of an Android Application for Smart Shopping"

Author Name: -Rajesh Kannan Megalingam; Souraj Vishnu; Swathi Sekhar; Vishnu Sasikumar; S. Sreekumar; Thejus R Nair

Year of Publish: 2016

Inference:

The growth of Android applications around the world is extraordinary. People turn towards technology for making their life more innovative and find solutions to their daily problems. When it comes to shopping, customers find it very difficult to find their products around the supermarket as well as to stand in long billing counter queues. Here we have come up with an android application which can be used in smart shopping carts that solve these dilemmas and provide a better shopping experience. Our smart app consists of two parts which mainly focuses on navigation to the item's location and automatic billing of the products that the user has purchased. The building environment of the app is the open source software called Android Studio software.

“Improved Billing System”

Author Name: Abhishek, Kumar Ishu, Kumar Vathsavi, Venkat

Year of Publish: 2014

Inference:

Paperless billing is a process businesses can use to get paid quickly and securely without dealing with the hassles of paper bills and check payments. Paperless bills allow bill delivery and payment to take place completely online, which improves efficiency while reducing costs. Paperless Billing generally involves integrating multiple systems including a billing system, banking system, a customer’s bank bill pay system, and an online interface for the Paperless Billing system. Paperless Billing is most helpful for businesses that send recurring bills to customers. Paperless Bills are an option for delivering bills. Bills can be presented either on a website or as an electronic document,such as a PDF file.

“Restaurant Billing System”

Author Name: Maharjan, Sujit

Year of Publish: 2014

Inference:

The main goal of this thesis was to develop a desktop based billing system for a restaurant. This desktop based application is designed to administer its users and customers. RBS is a billing system, made for the effective utilization of modern technology in the organization. It is an automated software that can handle a lot of information about the restaurant’s employees, order history, reservation data. It has the capability to process bills and gather information about its employees and billing history. It is designed for the sole purpose of efficiency, speed and accu-racy. This application allows the admin to view the detail history of day to day opera-tion in terms of sales, add or remove the employee, add new menu, view the detail of the employee like name, address, salary and so on. This application was devel-oped using Netbeans IDE 8.0.2 for the front end and MySQL database was used to store data.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is the act, process, or profession of studying an activity (as a procedure, a business, or a physiological function) typically by mathematical means in order to define its goals or purposes and to discover operations and procedure for accomplishing them most efficiently. It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

3.1 EXISTING SYSTEM

The present system billing works like the shopkeeper has POS software on which it stores the products data. It has information like product name, product price, and product stock. The software keeps updating the shopkeeper about stocks and expiry of the product. The shopkeeper uses the software to create bill for customers and keep track of their business. Working of the existing system is as follows:

1. The customer takes the products from the shop and goes to the billing counter.
2. The shop owner takes those products and scans bar code on the product using a barcode reader.
3. Using billing software they add the total cost of products.
4. Using that billing software, the process of adding products, creating bill, creating repository get automated.
5. After adding total cost and payment is made by customer, the shopkeeper creates a bill.
6. After creating the bill, customer pays the amount and the shopkeeper prints the bill using a printer.

3.2 PROPOSED SYSTEM

As efficient as the present system of billing is, it fails to help customer efficiently and also fails to conserve nature. The working of the project is as follows:

- First, we will create a system which is similar to a POS system.
- Second, we will generate a digital bill.
- Next, we will convert the digital bill into a QR code.
- Finally, the customer scans the QR code and bill will be transferred to the customer's
- phone.
- Some features of the project are:
- Billing system is user friendly and easy to run.
- Billing system can be operated without Internet.
- Billing system can be installed on computer or laptop so that accountants can use it
- anywhere.
- Billing system has facility to view the total invoices drawn for a particular day.
- Billing system has capability to view and reprint past invoices.
- Billing system has been set up to add discount to any item in a bill.
- Billing system has functionality to add multiple items in a single invoice.

3.3 PROBLEM STATEMENT:

There are many problems with the existing system. The paper used in billing to print the bill is costly and there is wastage of paper in this digital age. The paper used for printing bill has its own cost and the ink used is also costly. The printer is costly too. These costs take away a part of the profit from the shopkeeper. Paper is made from wood pulp which is produced after the cutting down of trees which has led to deforestation. Once the bill is printed, it cannot be corrected. Once billed, if any correction is needed then new bill needs to be printed. Once the bill is lost, there is no backup for it. If the customer need bill which they have kept at a place but the customer is somewhere else, then customer has to waste their time to go get it. Bill is not portable. After few days the bill printed ink starts fading away. Security of the bill is the biggest problem.

- Difficulties in controlling sales and purchases
- Keeping track the transaction of the business would be tough without a computer-based
- management system. Even the purchases of new orders would take hours recording and
- pricing.
- Low production cost is inefficiently high
- Evaluating the transaction per month is hard to do, thus low productivity and cost a lot
- of time to process the managing data given that it is valuable.
- Slow selling processes
- Due to the large number of customers, the selling process will be slow due to recording
- every transaction in a ledger.
- Paper bills are being used for records
- The bill paper if lost no back for customer to claim the transaction.
- Tampering the bill
- The bill can be easily tampered and can be used for fraud. With paper bill, it is easy to
- commit fraud.

CHAPTER 4

SYSTEM SPECIFICATIONS

4.1 INTRODUCTION

Improved billing system is an attempt to create an intuitive mobile and system application to make work easy, simple and digital. The idea is to design a system by keeping in the mind that shop owner has a computer with a webcam and customers have a smartphone with them. The User Interface of the application has to be kept simple and spontaneous with not a lot of effort required to operate it. To make it simpler and easier to use, the Android application will start with camera already working without any input from user. It will scan the QR code, decode it and automatically create the bill on phone. The user has to touch the save button to save bill on their phone.

4.2 HARDWARE REQUIREMENTS

Hardware requirements are the most common set of requirements defined by any operating system or software application. It is the physical computer resources, also known as hardware. A hardware requirements list is often complemented by a hardware compatibility list, especially for Operating Systems. The following are needed to efficiently use the application.

- Processor - Intel Core i3 and above
- Speed - 2.5 GHz
- RAM - 8 GB (min)
- Hard Disk - 50 GB
- Android phone with OS version 5.0 and above

4.3 SOFTWARE REQUIREMENTS

- Windows 7 and above
- JDK 1.7 and JDK 1.6
- Android Studio, Anaconda, Net-beans
- Java

4.4 TECHNOLOGIES USED

4.4.1 Java Programming Language

4.4.2 Net-beans

4.4.1 JAVA

Java is an object-oriented programming language developed initially by JamesGosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called Green OS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses.The reason for this was the introduction of a graphical Web browser called Mosaic, developed by NCSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic. It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java’s output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are insteadused to the Wintel platform, it will write the string passed to it to the user’s program.

Java consists of two things:

- Programming language
- Platform

The code and can bring about changes whenever felt necessary. Java is unusual in that each Java program is both co implied and interpreted. With a compiler, you translate a Java program into an intermediate language called Java byte codes – the platform independent codes interpreted by the Java interpreter.With an interpreter, each Java byte code instruction is parsed and run on the computer.

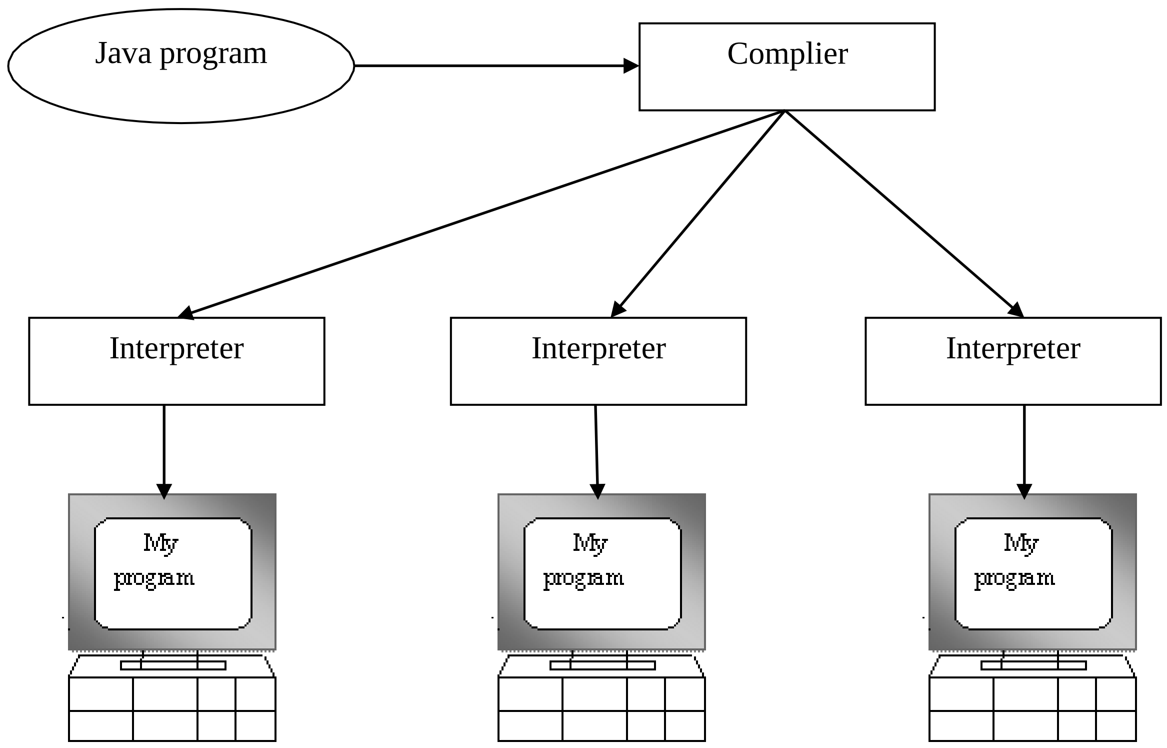


Fig 4.4.1.1

Java Machine

THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components:

- Java Virtual Machine (JVM)
- Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries **(packages)** of related components. Thefollowing figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

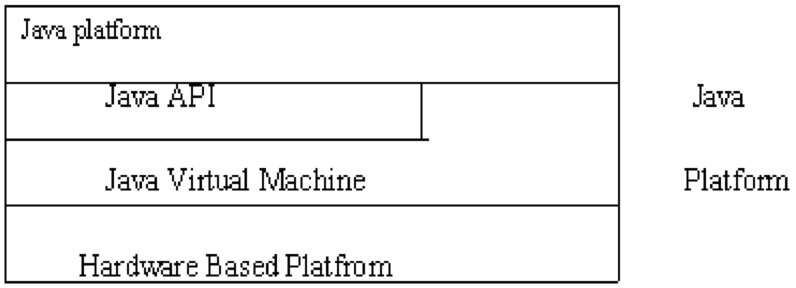


Fig 4.4.1.2
Java Platform

As a platform-independent environment, Java can be a bit slower than native code. However, smart compliers, well-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

4.4.2 NET-BEANS

The original business plan was to develop network-enabled JavaBeans components. Jaroslav Tulach, who designed the IDE's basic architecture, came up with the name **NetBeans** (from Network and Java Beans) to describe what the components would do. The IDE would be the way to deliver them.

When the specification for Enterprise Java Beans came out, it made more sense to work with the standard for such components than to compete with it, but the name stuck.

In the spring of 1999, NetBeans DeveloperX2 was released, adopting the javax.swing. **package names from the previous** com.sun.swing. ones. NetBeans was the first tool in on the market to support these new package names, and that increased a lot the awareness of NetBeans.

The performance improvements that came in JDK 1.3, released in the fall of 1999, made NetBeans a viable choice for development tools. By the summer of 1999, the team was hard at work re-architecting DeveloperX2 into the more modular NetBeans that forms the basis of the software today.

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5, and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports
development of all Java application types (Java SE (including JavaFX), Java
ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project
system, Maven support, refactorings, version
control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

Modularity: All the functions of the IDE are provided by modules. Each module provides a well-defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun

Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

License: The IDE is licensed under the Apache License 2.0. Previously, from July 2006 through 2007, NetBeans IDE was licensed under Sun's Common Development and Distribution License (CDDL), a license based on the Mozilla Public License (MPL). In October 2007, Sun announced that NetBeans would henceforth be offered under a dual license of the CDDL and the GPL version 2 licenses, with the GPL linking exception for GNU Classpath.

CHAPTER 5

SYSTEM DESIGN

System design is the process architecture, components, modules, interfaces and data for a system to satisfy specified requirements. System design could be seen as the application of systems theory to product development. System design is the process of defining the elements of a system such as architecture, modules, and components, the different interfaces of those components and the data goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system.

5.1 UML DIAGRAMS

5.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type ofbehavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

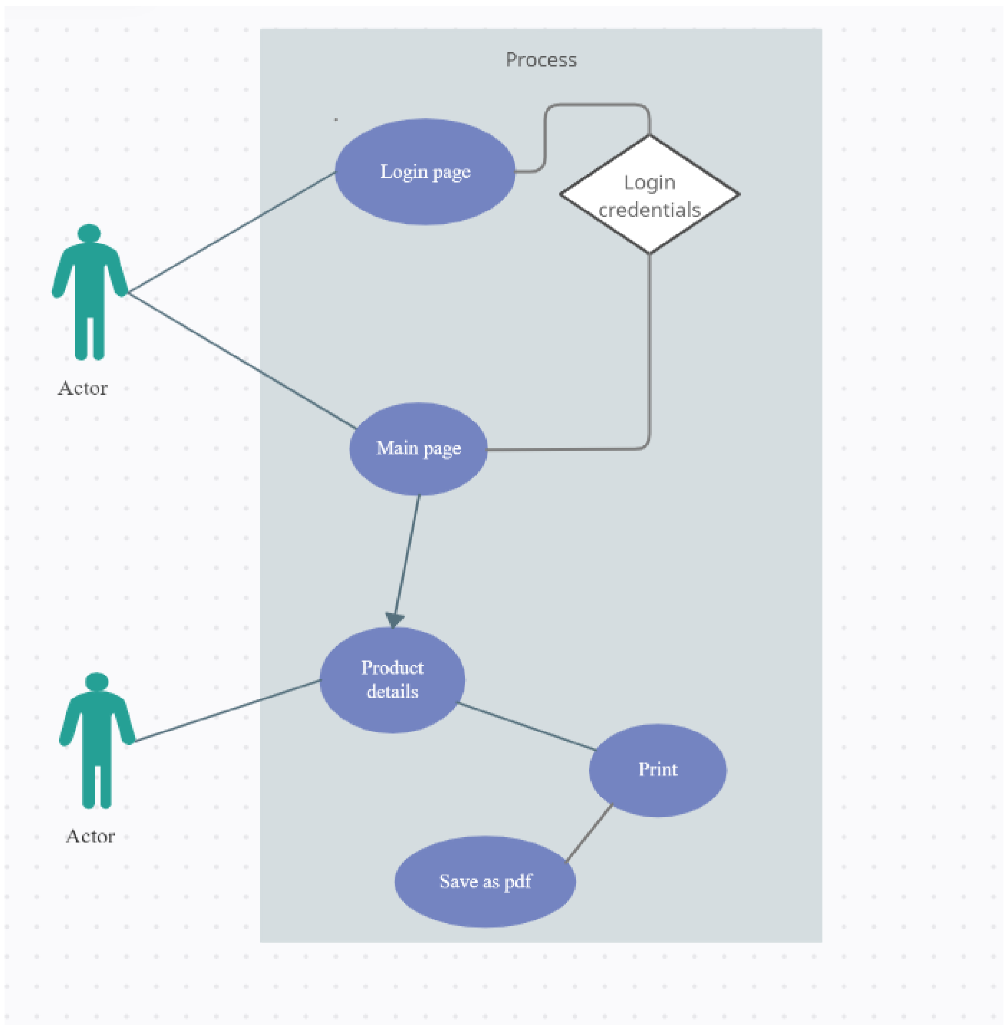


Fig 5.1.1

Use Case Diagram

5.1.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

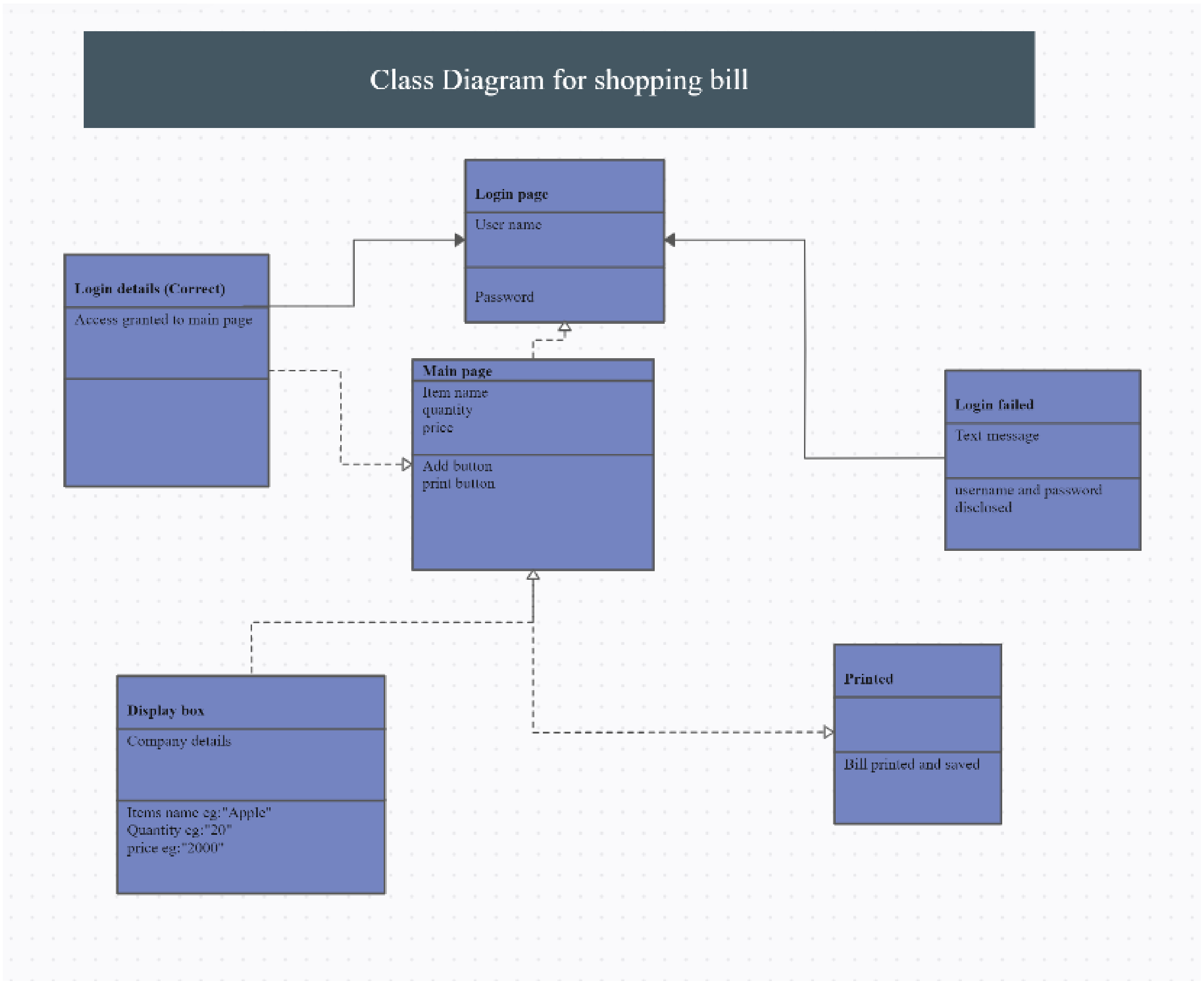


Fig 5.1.2

Class Diagram

5.1.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is type of flowchart, a graphical representation of the flow of data through an information system**.** It is also known as a data flow diagram, function diagram, or process diagram. Data flow diagrams are used to design the architecture of an information system and to document the functional aspects of it.

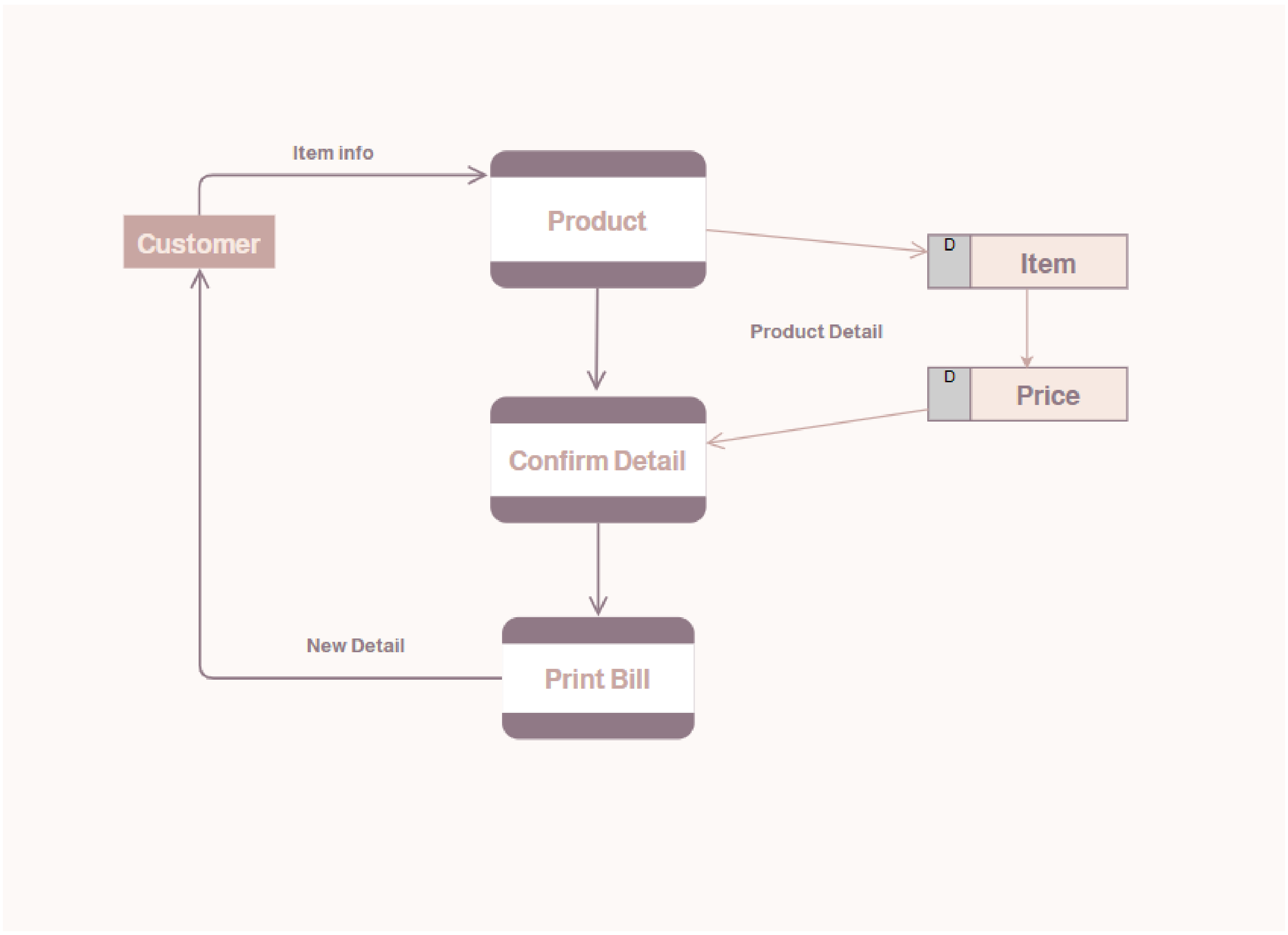


Fig 5.1.3

Data Flow Diagram

CHAPTER 6

SYSTEM MODULES

6.1 MODULES IMPLEMENTATION

A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components calledmodules that are integrated to satisfy problem requirements.

- Modularity is the single attribute of software that allows a program to be intellectually manageable.
- The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.
- The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

6.2 MODULE DESCRIPTION

- User Login
- Item Detail
- Calculated Price

6.2.1 USER LOGIN

In this module a login form contains only two fields, i.e., username and password. Each user should have a unique username that can be an email, phone number, or any custom username. After submitting the login form, the underlying code of the form checks whether the credentials are authentic or not to allow the user to access the restricted page. If the users provide unauthentic credentials, they will not be able to forward the login form.In Java, we can develop the login form by using Swing technology. The model will ask for user name and password for logging in the main page as billing is import and confidential, only people with authority can login with the correct credential and user the main server page.

6.2.2 ITEM DETAIL

This module the user has to enter the item details that is the product name, the quantity and the price that is given for that particular product. Once it is done the user can click the add button in the bottom to add the details in the text box beside. If there is any extra the user can enter it and upload it again. This process follows until the last item is added in the bill.

6.2.3 CALCULATED PRICE

If the entered domain information in the text box is sufficient and over, the user can click the print button in the bottom to print the details that is present in the text box. Once the button is clicked a new tab appears, which will ask for the format of the document to be save din the system (preferred to save in .pdf) and later on it gets saved in the respective location that is entered.

CHAPTER 7

SYSTEM TESTING

7.1 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand.
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

7.2 TEST PROCEDURE

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.3 TYPES OF TESTS

UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module.

FUNCTIONAL TEST

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and thespecial values, such as logically related inputs, files of identical elements, and empty files. Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test.

PERFORMANCE TEST

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

STRUCTURED TEST

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategywas employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or falsesides.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems andtextual errors in output information.

INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should betested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on.
The advantages of this approach are that, interface dispenses can be easily found
and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. There was a check on the interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowingthe specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is awhite box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis

SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested forinvulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

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 test-validation testing begins. Validation testing can be defined inmany ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. After validation test has been conducted, one of two conditions exists.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system.The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required.

CHAPTER 8

CONCLUSION

This project was about improving the existing billing system by making it digitalized. A better way of billing system was implemented. Instead of a printed bill, the customer obtained a digital copy of it, which reduced the wastage of paper. This also helped the seller to save the cost of buying paper. The bill was first stored in the seller’s system in an organized manner. This also ensured the customer wouldn’t lose their bill. This project was implemented to make the process of billing more efficient.

FUTURE ENHANCEMENT

This application has much potential for future enhancement. As the new technologies are easily available as open source, we can add many new features to it. Few of the enhancements we can work on are as follows:

• Add payment method to the Android application itself.

• Add feature to add product to cart using the application itself.

• Improve the GUI to make it modern or futuristic.

• Utilize cloud functionality to store the bill on cloud, which will save the phone storage.

• Make bill more secure by adding more security measures.

• Make it available on different platforms like iOS, Oak, Linux, etc

APPENDIX 1

SAMPLE CODING:

USER LOGIN:

```
package javabilling;

import javax.swing.JOptionPane;


/**

 *

 * @author Ashwanth C

 */

public class BillPage extends javax.swing.JFrame {


    /**

     * Creates new form BillPage

     */

    public BillPage() {

        initComponents();

    }
```

```
/**
 * 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">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    username = new javax.swing.JTextField();
    jLabel5 = new javax.swing.JLabel();
    password = new javax.swing.JPasswordField();
    jButton1 = new javax.swing.JButton();

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jScrollPane1.setViewportView(jTextArea1);
```

```
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


        jLabel1.setBackground(new java.awt.Color(0, 0, 0));

        jLabel1.setFont(new java.awt.Font("Monotype Corsiva", 1, 48)); // NOI18N

        jLabel1.setForeground(new java.awt.Color(204, 0, 204));

        jLabel1.setText("Ashwanth & Co");

        jLabel1.setBorder(new
    javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));

        jLabel1.setMaximumSize(new java.awt.Dimension(100, 25));


        jLabel2.setBackground(new java.awt.Color(255, 255, 255));

        jLabel2.setFont(new java.awt.Font("HP Simplified Hans", 2, 24)); // NOI18N

        jLabel2.setForeground(new java.awt.Color(153, 153, 255));

        jLabel2.setText("LOGIN PAGE");


        jLabel3.setFont(new java.awt.Font("Verdana", 0, 18)); // NOI18N

        jLabel3.setForeground(new java.awt.Color(0, 204, 204));

        jLabel3.setText("User name:");


        jLabel5.setBackground(new java.awt.Color(153, 0, 0));

        jLabel5.setFont(new java.awt.Font("Verdana", 0, 18)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 204, 204));
        jLabel5.setText("Password:");


        password.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                passwordActionPerformed(evt);
            }
        });
        jButton1.setBackground(new java.awt.Color(0, 0, 0));
        jButton1.setForeground(new java.awt.Color(255, 255, 255));
        jButton1.setText("Submit");
  javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(173, 173, 173)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
false)
                            .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                        .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGap(49, 49, 49)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                        .addComponent(username)
                        .addComponent(password, javax.swing.GroupLayout.DEFAULT_SIZE, 214,
Short.MAX_VALUE)))
                .addGroup(layout.createSequentialGroup()
                    .addGap(295, 295, 295)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE))))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGap(0, 216, Short.MAX_VALUE)
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 314,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(197, 197, 197))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
            .addGroup(layout.createSequentialGroup()
                .addGap(33, 33, 33)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 52,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(40, 40, 40)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 58,
Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel3)
                    .addComponent(username, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(54, 54, 54)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel5)
                    .addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(70, 70, 70)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(146, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
```

```
private void passwordActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String un=username.getText();
    String pass=password.getText();
    if(un.equals("Ashwanth")&&pass.equals("Achu")){
    Mainpage obj=new Mainpage();
    obj.setVisible(true);
    dispose();
    }
    else{
        JOptionPane.showConfirmDialog(rootPane,"Invalid Username or Password! Try Again");
        username.setText("");
        password.setText("");
    }
}
```

ITEM DETAILS:

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
```

```
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(BillPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(BillPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(BillPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
       java.util.logging.Logger.getLogger(BillPage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new BillPage().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JPasswordField password;
    private javax.swing.JTextField username;
    // End of variables declaration

  }
```

CALCULATED PRICE:

```
package javabilling;


/**
 *
 * @author Ashwanth C
 */
public class Mainpage extends javax.swing.JFrame {


    /**
     * Creates new form Mainpage
     */
    public Mainpage() {
        initComponents();


        billhead();


    }
    private void billhead(){
```

```
txtbill.setText("==================================================="+"\n"
+
                "\t"+"Ashwanth & Co"+"\n"
                +"\t"+"Contact No-7904614742"+"\n"+"\tGST No-
33AADTS3688K1Z0"+"\n"+"===================================================
===="+"\n");

    }




    /**
     * 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">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
```

```
jLabel4 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

txtbill = new javax.swing.JTextArea();

jButton2 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jLabel5 = new javax.swing.JLabel();

txtname = new javax.swing.JTextField();

txtquantity = new javax.swing.JTextField();

txtprice = new javax.swing.JTextField();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

setBackground(new java.awt.Color(255, 255, 255));

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

setPreferredSize(new java.awt.Dimension(990, 700));


jLabel1.setBackground(new java.awt.Color(255, 102, 153));

jLabel1.setFont(new java.awt.Font("Garamond", 3, 36)); // NOI18N

jLabel1.setForeground(new java.awt.Color(102, 102, 255));

jLabel1.setText("Shopping Bill ");


jLabel2.setBackground(new java.awt.Color(102, 0, 102));

jLabel2.setFont(new java.awt.Font("MS Reference Sans Serif", 1, 24)); // NOI18N
```

```
        jLabel2.setForeground(new java.awt.Color(102, 0, 102));

        jLabel2.setText("Item Name:");

        jLabel2.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
java.awt.Color(255, 255, 51)));


        jLabel3.setBackground(new java.awt.Color(153, 0, 153));

        jLabel3.setFont(new java.awt.Font("MS Reference Sans Serif", 1, 24)); // NOI18N

        jLabel3.setForeground(new java.awt.Color(102, 0, 102));

        jLabel3.setText("Quality     :");

        jLabel3.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
java.awt.Color(255, 255, 0)));


        jLabel4.setFont(new java.awt.Font("Microsoft Sans Serif", 1, 24)); // NOI18N

        jLabel4.setForeground(new java.awt.Color(102, 0, 102));

        jLabel4.setText("Price             :");

        jLabel4.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 1, 1, new
java.awt.Color(255, 255, 0)));


        txtbill.setColumns(20);

        txtbill.setRows(5);

        jScrollPane1.setViewportView(txtbill);


        jButton2.setFont(new java.awt.Font("Rockwell Condensed", 1, 18)); // NOI18N
        jButton2.setText("Add");

jButton2.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.
RAISED, new java.awt.Color(0, 153, 153), new java.awt.Color(255, 0, 0), new java.awt.Color(0,
153, 153), new java.awt.Color(255, 0, 0)));
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jButton1.setFont(new java.awt.Font("Rockwell Condensed", 1, 24)); // NOI18N
        jButton1.setText("Print");
        jButton1.setPreferredSize(new java.awt.Dimension(990, 600));
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        jLabel5.setFont(new java.awt.Font("Castellar", 1, 24)); // NOI18N
        jLabel5.setText("WELCOME");
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(374, 374, 374))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                        .addComponent(jLabel5)
                        .addGap(412, 412, 412))))
            .addGroup(layout.createSequentialGroup()
                .addGap(80, 80, 80)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(289, 289, 289)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 346,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(52, 52, 52)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                            .addComponent(txtname)
                            .addComponent(txtquantity)
                            .addComponent(txtprice, javax.swing.GroupLayout.DEFAULT_SIZE, 160,
Short.MAX_VALUE))
                        .addGap(124, 124, 124)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 346,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(86, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
    .addGroup(layout.createSequentialGroup()
        .addGap(39, 39, 39)
        .addComponent(jLabel5)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel1)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(110, 110, 110)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(txtname, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(96, 96, 96)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel3)
                    .addComponent(txtquantity, javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(92, 92, 92)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(txtprice, javax.swing.GroupLayout.PREFERRED_SIZE, 30,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(100, 100, 100)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 321,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(36, 36, 36)
                        .addComponent(jButton2))
                    .addGroup(layout.createSequentialGroup()
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 36,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(573, Short.MAX_VALUE))
        );


        pack();
    }// </editor-fold>


    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        txtbill.setText(
        txtbill.getText()+txtname.getText()+"-
```

```
"+"\t"+txtquantity.getText()+".qnt\t\t"+txtprice.getText()+".Rs\n");
        txtname.setText("");
        txtquantity.setText("");
        txtprice.setText("");


    }


    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{

txtbill.setText(txtbill.getText()+"==========================================
======"+"\n"+"Thank You, visit again !");
            txtbill.print();
        }catch(Exception e){
        }
    }


    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Mainpage.class.getName()).log(java.util.logging.Level.SEVERE
, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Mainpage.class.getName()).log(java.util.logging.Level.SEVERE
, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Mainpage.class.getName()).log(java.util.logging.Level.SEVERE
, null, ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Mainpage.class.getName()).log(java.util.logging.Level.SEVERE
, null, ex);

        }

        //</editor-fold>


        /* Create and display the form */

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new Mainpage().setVisible(true);

            }

        });

    }

    // Variables declaration - do not modify

    private javax.swing.JButton jButton1;

    private javax.swing.JButton jButton2;

    private javax.swing.JLabel jLabel1;

    private javax.swing.JLabel jLabel2;

    private javax.swing.JLabel jLabel3;

    private javax.swing.JLabel jLabel4;

    private javax.swing.JLabel jLabel5;

    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea txtbill;

    private javax.swing.JTextField txtname;

    private javax.swing.JTextField txtprice;

    private javax.swing.JTextField txtquantity;

    // End of variables declaration

}
```

APPENDIX 2

SNAPSHOTS

USER LOGIN:

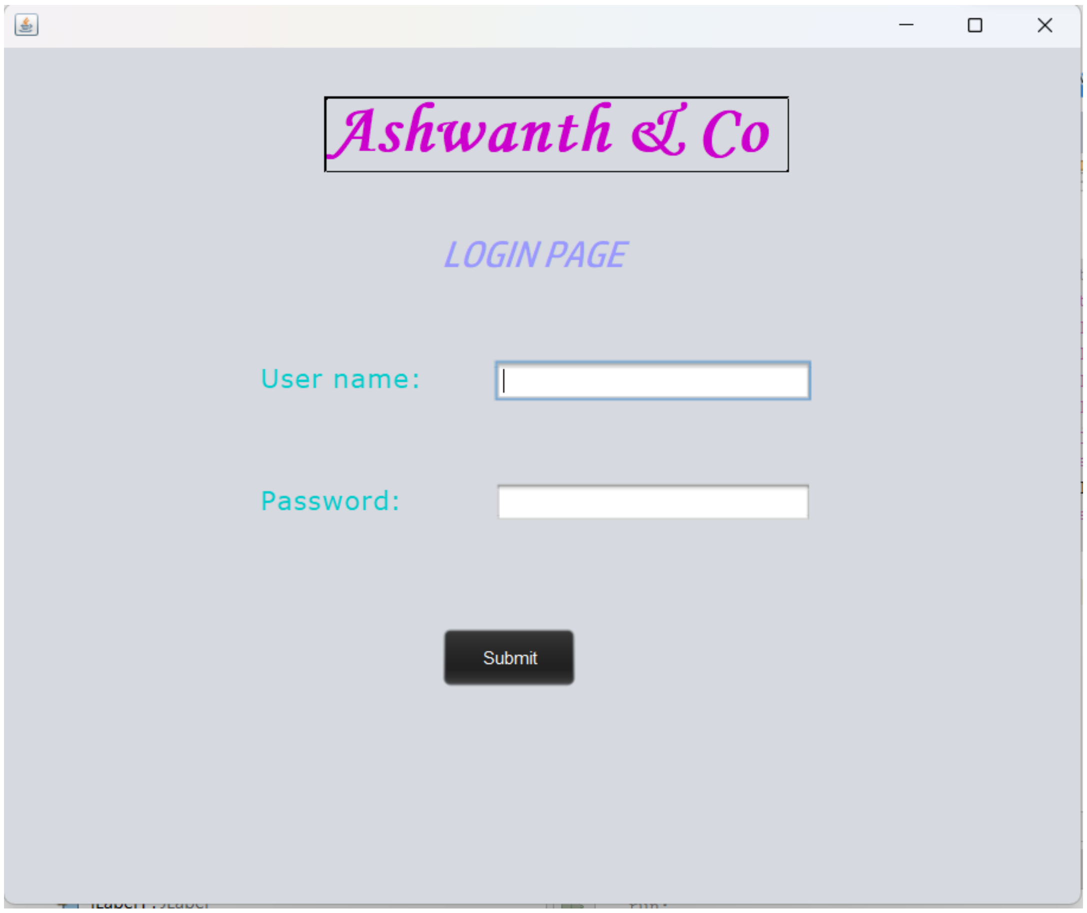


Fig 8.4.1.1

User Login Page

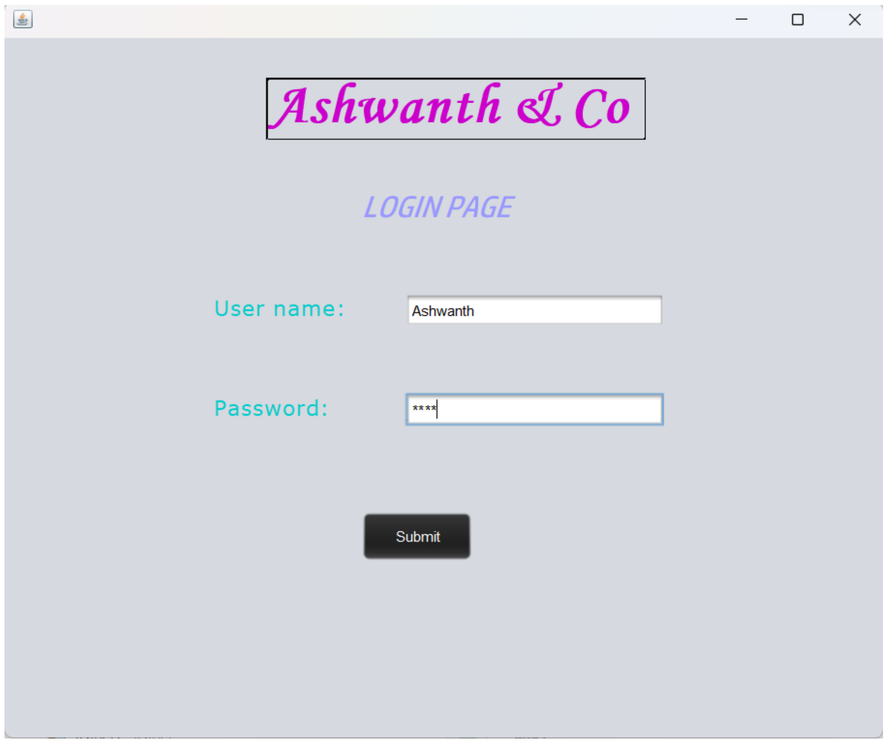
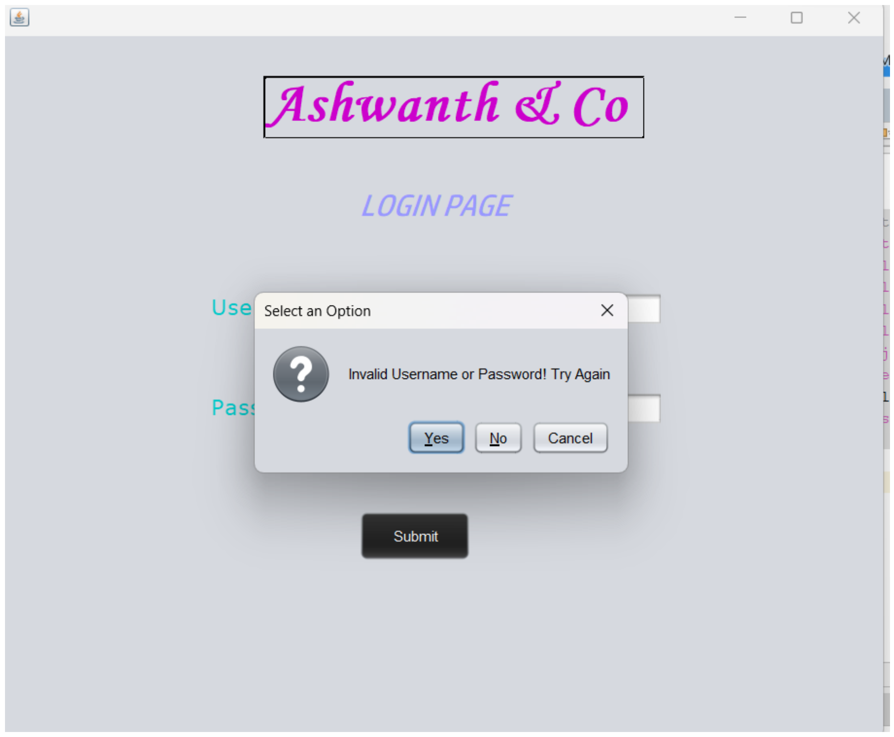


Fig8.4.1.2

Credentials

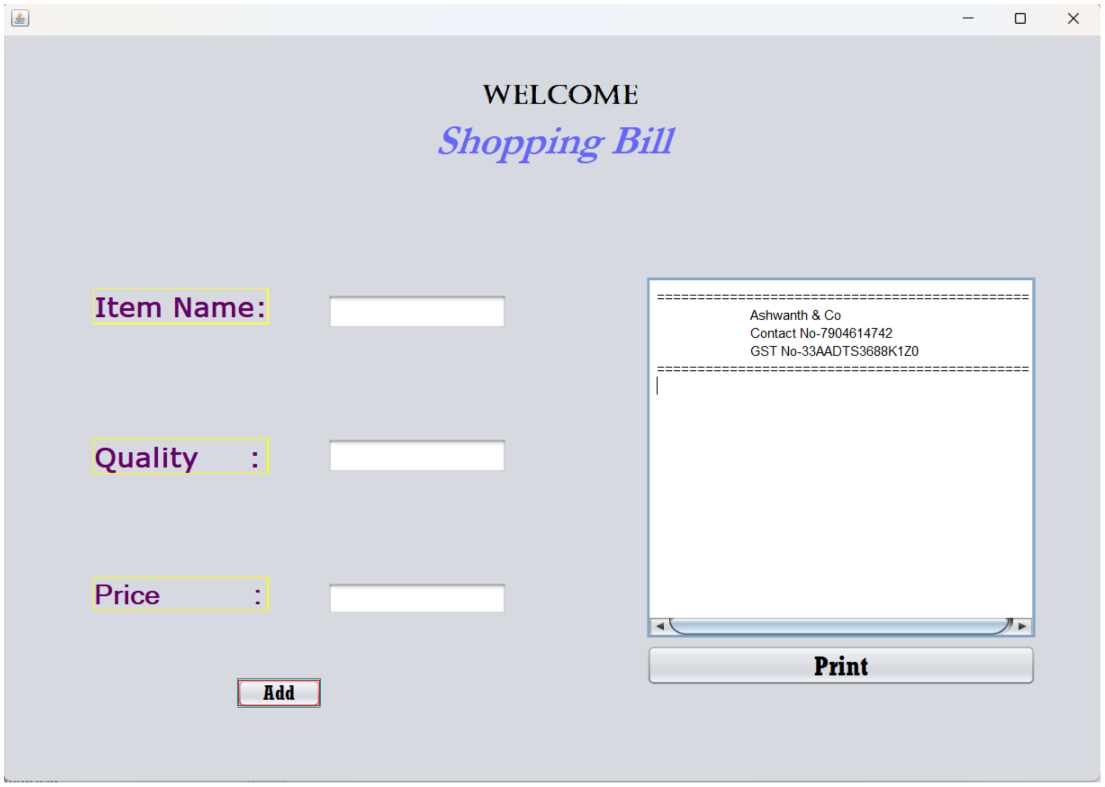


Invalid Username or Password! Try Again

Fig 8.4.1.3

Invalid Credentials

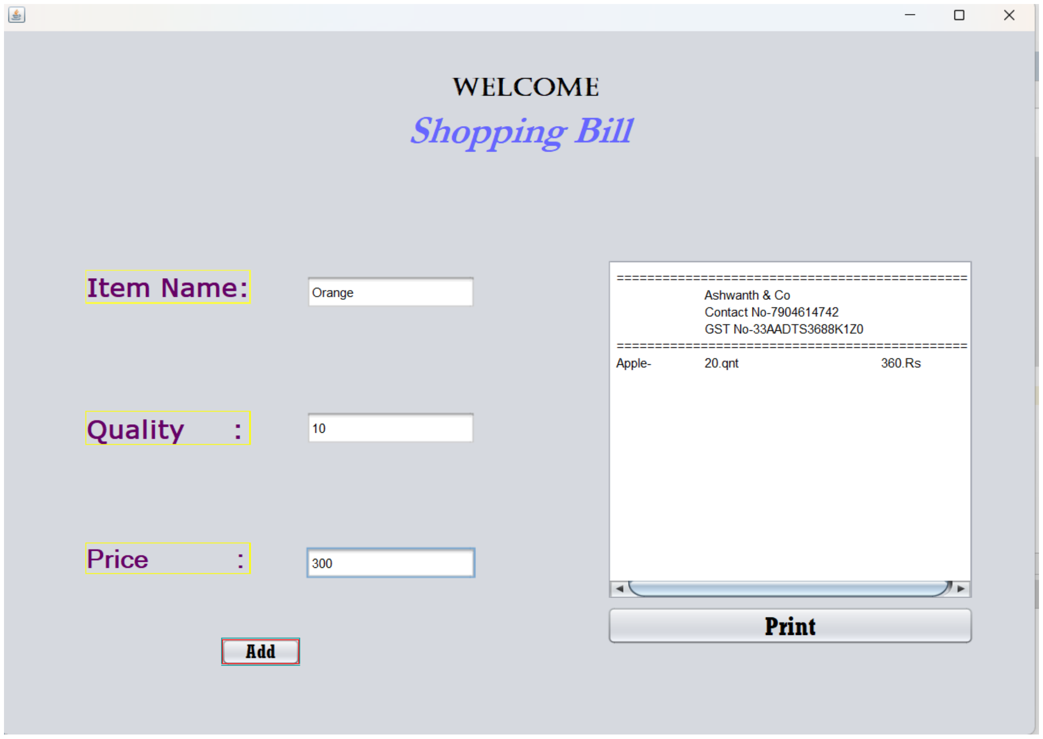
ITEM DETAIL:



```
==============================================
              Ashwanth & Co
              Contact No-7904614742
              GST No-33AADTS3688K1Z0
==============================================
```

Fig 8.4.2.1

Home Page



Orange

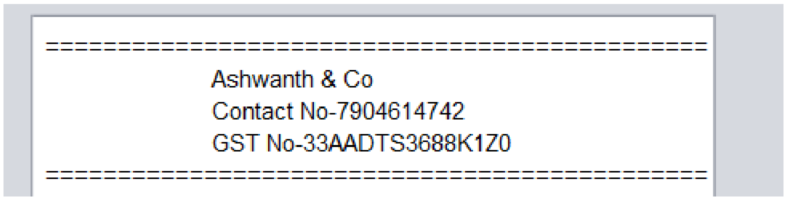
10

300

```
==============================================
            Ashwanth & Co
            Contact No-7904614742
            GST No-33AADTS3688K1Z0
==============================================
Apple-      20.qnt                  360.Rs
```

Fig 8.4.2.2

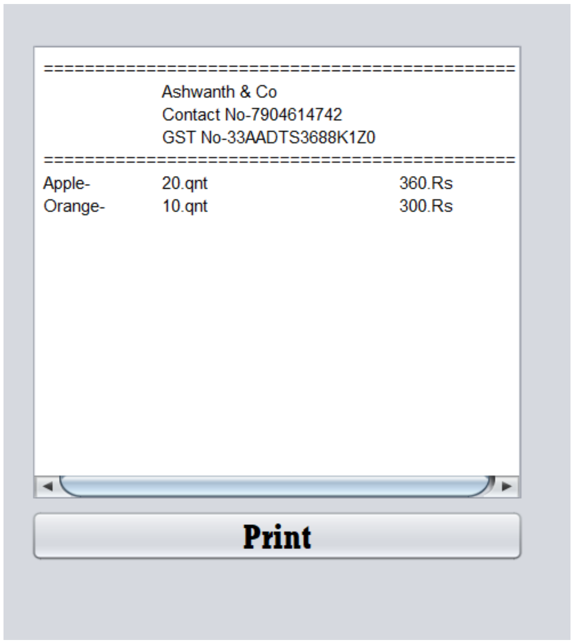
Details Added



```
==============================================
            Ashwanth & Co
            Contact No-7904614742
            GST No-33AADTS3688K1Z0
==============================================
```

Fig 8.4.2.3

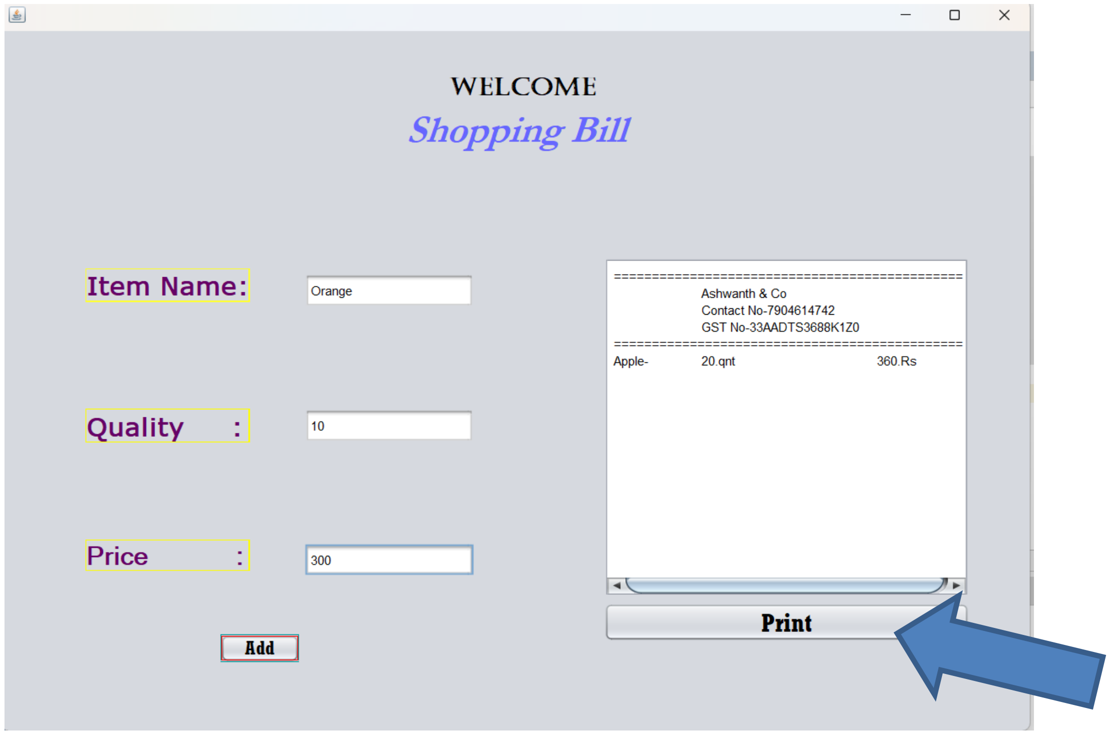
Company Details



```
===============================================
            Ashwanth & Co
            Contact No-7904614742
            GST No-33AADTS3688K1Z0
===============================================
Apple-      20.qnt                  360.Rs
Orange-     10.qnt                  300.Rs
```

Fig 8.4.2.4

Items Purchased



WELCOME

Shopping Bill

Item Name: Orange

Quality : 10

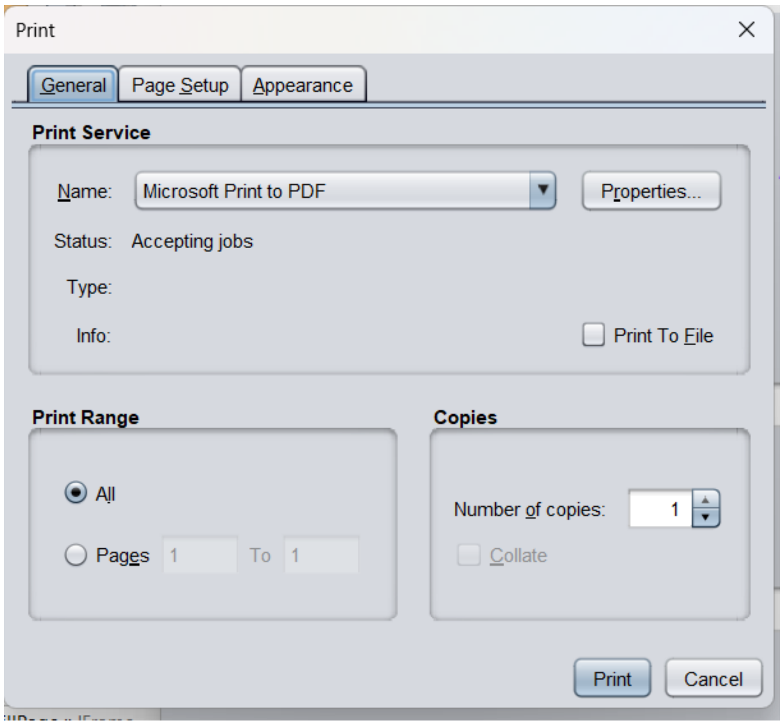
Price : 300

```
===============================================
            Ashwanth & Co
            Contact No-7904614742
            GST No-33AADTS3688K1Z0
===============================================
Apple-      20.qnt                  360.Rs
```

Fig 8.4.2.5

Final Purchase

CALCULATED PRICE:


Print

General | Page Setup | Appearance

Print Service

Name: Microsoft Print to PDF

Status: Accepting jobs

Type:

Info:

Print To File

Print Range

(● All) ○ Pages 1 To 1

Copies

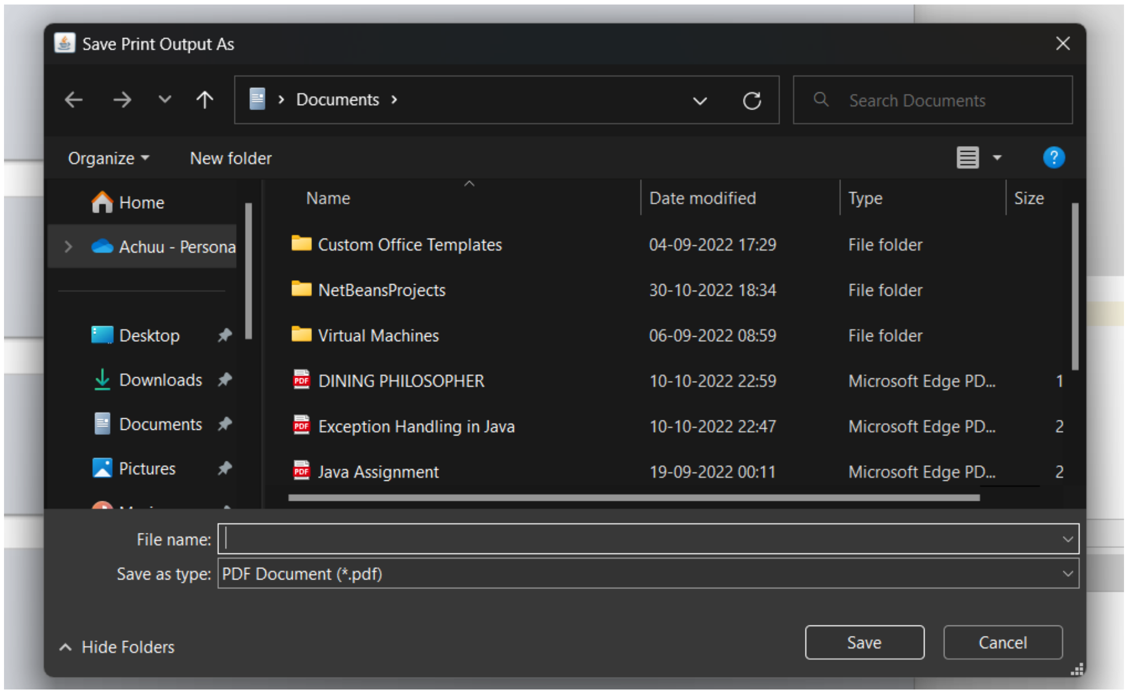
Number of copies: 1

Collate

Print | Cancel

Fig 8.4.3.1

Print Details


Save Print Output As

Documents

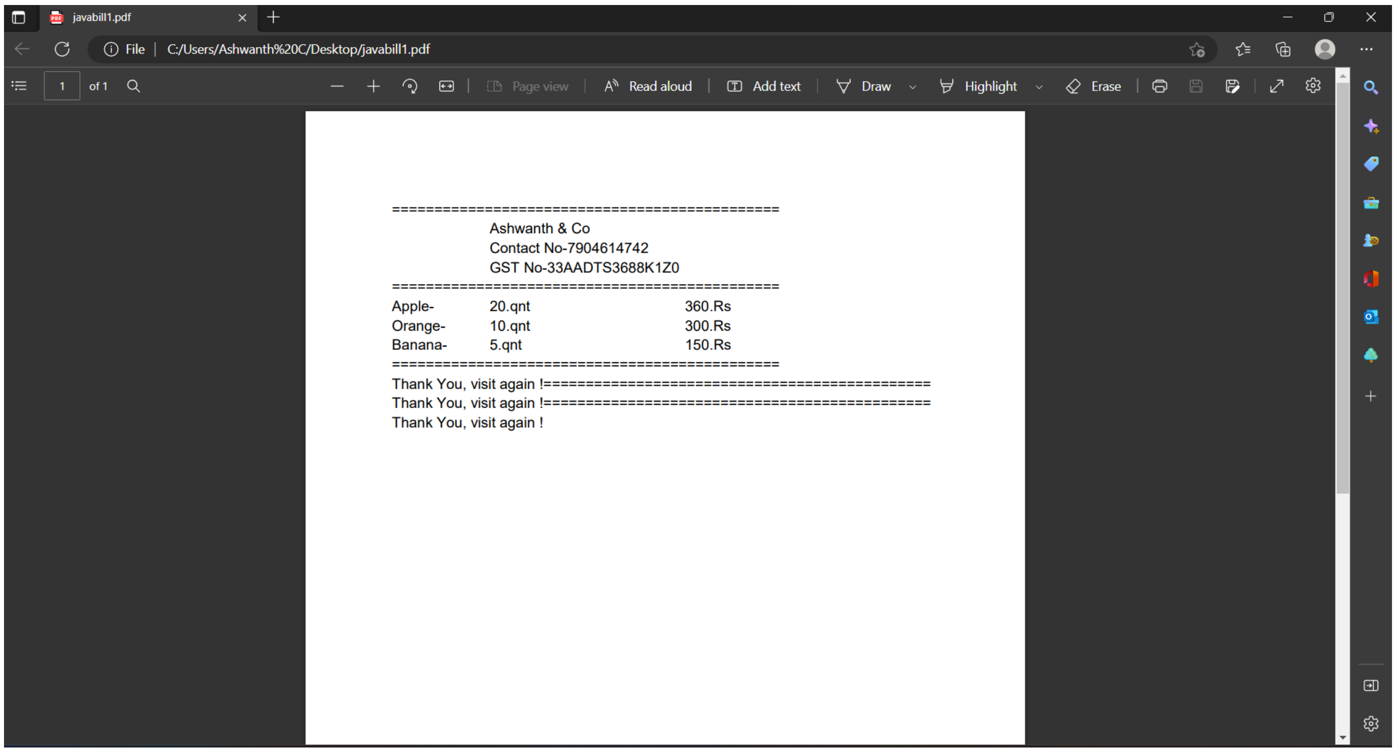
Name	Date modified	Type	Size
Custom Office Templates	04-09-2022 17:29	File folder	
NetBeansProjects	30-10-2022 18:34	File folder	
Virtual Machines	06-09-2022 08:59	File folder	
DINING PHILOSOPHER	10-10-2022 22:59	Microsoft Edge PD...	1
Exception Handling in Java	10-10-2022 22:47	Microsoft Edge PD...	2
Java Assignment	19-09-2022 00:11	Microsoft Edge PD...	2

File name:

Save as type: PDF Document (*.pdf)

Fig 8.4.3.2

Saving File



```
============================================
           Ashwanth & Co
           Contact No-7904614742
           GST No-33AADTS3688K1Z0
============================================
Apple-     20.qnt                  360.Rs
Orange-    10.qnt                  300.Rs
Banana-    5.qnt                   150.Rs
============================================
Thank You, visit again !============================================
Thank You, visit again !============================================
Thank You, visit again !
```

Fig 8.4.3.3

Final Printed Bill

REFERENCES

[1]Md. Mejbaul Haque, Md.Kamal Hossain, Md.Mortuza Ali and Md.Rafiqul Islam Sheikh, "Microcontroller Based Single Phase Digital Prepaid Energy Meter for Improved Metering and Billing system", *International of Power Electronics and Drive System (IJPEDS)*, vol. 1, no. 2, pp. 139, December 2011.

[2]Jia Huiqin and Zhang Ru, "A Novel Remote Meter-reading System based on Virtual Instrument", *Computer Science and Automation Engineering (CSAE) 2011 IEEE International Conference on*, vol. 4, 10–12 June 2011.

[3]Tariq Jamil, "Design and Implementation of a Wireless Automatic Meter Reading System", *Proceeding of the World Congress on Engineering 2008 Vol 1 WCE 2008*, July 2–4, 2008.

[4]HG Rodney Tan, CH Lee and VH Mok, "Automatic Power Meter Reading using GSM Network", *8th International Power Engineering Conference 2007*, pp. 465-469.

[5]S. Arun and Sidappa Naidu, "Hybrid Automatic Meter Reading System", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 7, July 2012, ISSN 2277 128X.

[6]Alauddin AI-Omary, Wael El-Medany and Sufyan Al-Irhayim, "Secure Low-Cost AMR System Based on GPRS Technology", *International Journal of Computer Theory and Engineering*, vol. 4, no. 1, February 2012.