

## **Abstract**

This report outlines the contents of a minimal set of Store bill Generator software development documents, tailored for use by students in software engineering projects, and firmly based on Java standards. The document set is designed to support software development activities. A supplementary report describes the content of each document in more detail. I would like to express our appreciations to all individuals who assisted us in any way and we sincerely of that no one feel left out.

I also wish to express our indebtedness to those who helped me .This would not have been made successful without his help and precious suggestions .I mostly thankful to Sofiqul Islam, Debashis Barman, Bashar Sardar, Tanvir Alom, Chanchal Kumar sir who, guide every time when we required with Programming and other aspect.

## Objective

Store Bill Generators a software that can make bill of customer and make pdf file of it automatically. It used to make Bill of customer.

The software will help to shopkeeper, to make bill of customer, and make documentation of it.

My software has been designed a general way. It contains ...

- Name of Customer
- Date
- Name of Product
- Received Money
- Back Money
- Save as PDF

# Table of Contents

<b>Chapter1.....</b>	<b>5-6</b>
----------------------	------------

## **Introduction**

- 1.1 Introduction
- 1.2 Background
- 1.3 Project description
- 1.4 Which technology used to make

<b>Chapter2 .....</b>	<b>7-11</b>
-----------------------	-------------

## **Literature Review**

- 2.1 Introduction
- 2.2 Object-Oriented Programming Concept
  - 2.2.1 Classes
  - 2.2.2 Objects
- 2.3 Characteristics of Object Oriented System
  - 2.3.1 Encapsulation
  - 2.3.2 Inheritance
  - 2.3.3 Polymorphism
- 2.4 Object oriented design
- 2.5 About Programming Language “Java”

**Chapter 3.....12-13**

**Project Detail**

3.1 Palette and Method Information

3.1.1 Palette

3.1.2 Method

3.2 Coding Information

3.2.1 Packages

3.2.2 Button input operation

3.3 Instruction

3.4 Conclusion

**Chapter-4 .....14-15**

**Interfaces of the application**

4.1 Bill Make Form

4.2 PDF File

**Chapter-5.....16**

**Conclusion and Future Work**

5.1 Limitation

5.2 Future plan

5.3 Suggestions

5.4 Conclusion

# Chapter 1

## Introduction

### 1.1 Introduction

#### Meaning of Project

The meaning of project is to give physical existence to the various brain ideas and thoughts. Project is a great source to develop technical skills in the Technical student.

During the programming of the project student goes through different problems and experience. A student gets a change to climb on the roof of practical knowledge to economize to go at the fields. A very important aim of this world “PROJECT” consist of seven alphabets each of them has separate meaning.

- “P” for Planning – Planning is good and impressive half work is done.
- “R” for RELIABLE SOURCE – Practice and theoretical material and able guidance and assistance is achieved from different sources to promote the function of planned area.
- “O” for OVERALL EXPENSES – Overall expenses to provide final design to any project is also considerable. Their expenses are analyzed for each concept and method of working on the computer.
- “J” for JOINT EFFORTS – Joint efforts are much necessary for programming and result of any project. Many times some problems as books etc., are solved for external sources.
- “E” for ECONOMIC TIME FEASIBILITY – It is important that the planned scheme is how much economic time feasible. It is must to analyze estimate for completion of the project.
- “C” for CONSTRUCTION – After analyzing the project work is done to give logical existence of planned scheme.
- “T” for TESTING – Testing of program is done before submitting the project after testing the gadget is ok for use.

Java is the most widely used programming language in the world of its Object Oriented Properties. Although it's widely used for system programming, everything can be done by using Java. From system to application programming it has a supreme power. Java has strong graphics library. It contains thousands of methods.

### 1.2 Background

#### Definition of the Store Bill Generator

Store Bill Generator is that software which makes Bill report of product. It is used to make Bill of sell product.

#### Target Learner

This project can be used by shopkeeper. They can make Bill of sell product and make documentation of it.

### **1.3 Project description**

The purpose is for Store Bill Generator. There a shopkeeper simply make Bill of sell product and save it as pdf. I am trying to implement a simple Store Bill Generator in java.

### **1.4 Which technology used to make**

- Java SE
- AWT (Abstract Window Toolkit)
- Swing
- iText

#### **Java SE**

We used Java SE to make the application.

#### **AWT**

We used AWT framework of Java to make GUI Application.

#### **Swing**

Swing is update version of AWT. We can make GUI application and Handle any Event used it.

#### **iText**

We can make pdf to used iText.

## **Chapter 2**

### **Literature Review**

#### **2.1 Introduction**

It is designed to make Bill of sell product for shopkeeper. This projects developed in a JAVA platform is a replica I am familiar with and which I use quite often on a daily basis. The only difference being that, this editor has been created using JAVA for the front-end interface.

#### **2.2 Object-Oriented Programming Concepts**

OOP is used for a large class of common software applications. It's likely that the object-oriented approach will undergo further evolution and that a new, improved software-development paradigm will take its place. But for right now, and the foreseeable future, it is recognized as the best approach for the majority of software that we develop today.

##### **2.2.1 Classes**

A class is a template or prototype that defines a type of object. A class is to an object what a blueprint is to a house. Many houses may be built from a single blueprint; the blueprint outlines the makeup of the houses. Classes work exactly the same way, except that they outline the makeup of objects.

##### **2.2.2 Objects**

Objects are software bundles of data and the procedures that act of that data. The procedures are also known as methods. The merger of data and methods provides a means of more accurately representing real-world objects in software. Without objects, modeling a real-world problem in software requires a significant logical leap. Objects, on the other hand, enable programmers to solve real-world problems in the software domain much easier and more logically.

#### **2.3 Characteristics of Object Oriented System**

There are three characteristics that make object oriented systems unique. They are-

- (1) Encapsulation,
- (2) Inheritance,
- (3) Polymorphism.

##### **2.3.1 Encapsulation**

Encapsulation is the process of packaging an object's data together with its methods. A powerful benefit of encapsulation is the hiding of implementation details from other objects.

##### **2.3.2 Inheritance**

When there is an object that is very similar to one that already exists, but that has a few extra characteristics then we should inherit a new class based on the class of the similar object. Inheritance is the process of creating a new class with the characteristics of an existing class.

##### **2.3.3 Polymorphism**

Polymorphism is the ability to assume different forms. In object-oriented programming, this refers to the ability of objects to have many methods of the same name, but with different forms. The compiler and runtime system support polymorphism by matching each method invocation to the correct method, class, and object.

#### **2.4 Object oriented design**

Object oriented design transforms the analysis model created using object-oriented analysis into a design model that serves as a blueprint for software construction. Unlike conventional software design methods, OOD results in a design that achieves a number of different levels of modularity.

1. Abstraction
2. Information hiding
3. Functional independence

4. Modularity Only OOD provides a mechanism that enables the designer to achieve all four with less complexity and compromise.

## **2.5 About Programming Language “Java”**

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. This language was initially called “oak” but was renamed “Java” in 1995, between the initial implementation of oak in the fall of 1992 and public announcement of Java in the spring of 1995, many more people contributed to the design and evolution of the language. Bill Joy, Arthur van Hoff, Jonathan Payne, Frank Yellin, and Tim Lindholm were key contributors to the maturing of the original prototype.

Somewhat surprisingly, the original impetus for Java was not the INTERNET! Prototype. The primary motivation was the need for a platform-independent (that is architecture –neutral) language that could be used to create software to be embedded in various consumer electronic devices such as microwave ovens and remote controls. As you can probably guess many different types of CPU are used as controllers. The trouble with C and C++ (and most other languages) is that they are designed to be compiled for a specific target. Although it is possible to compile a C++ program for just about any type of CPU, to do so requires a full compiler targeted for that CPU.

Java is a programming language and environment invented by James Gosling in 1994. Gosling was the first designer of the Java programming language and implemented its original compiler and virtual machine.

Java is the first and foremost programming language. Creation of Java was driven by both elements in nearly equal measures which are:

1. To adapt to changing environments and uses.
2. To implement refinements and improvements in the art of programming.

"Java: A simple, object-oriented, distributed, interpreted, robust, secure, portable, high-performance, multithreaded, and dynamic language" –Java Soft

A Java Program: From Birth to Execution

1. Coding: Human-readable Java code is produced by the programmer
2. Building: A Java Development Tool “build”s the Java program into byte-code, which is saved as a “. class” file.
3. Loading: Via the web or command line, the class file is sent to the Java Virtual Machine (VM) with an attached digital signature. The Java VM is simply an interpreter.
4. Byte code Verification: The Java VM verifies the digital signature. When downloaded remotely, the Java VM isolates the Java program in a restricted part of memory. The Java program is not allowed to access local Hard drives and System resources.



5. Internal Integrity: Verification checks are made to insure that the loaded Java program is well formed. Data types are verified along with other syntax structure

6. Execution: Program execution begins

### **Java Project**

In 1990, Sun Microsystems began a project called Green to develop software for consumer electronics. Gosling began writing software in C++ for embedding into such items as toasters, VCR's, and Personal Digital Assistants (PDA's). The embedded software makes more appliances more intelligent. Gosling's solution to the problem of C++ was a new language called Oak. Finally in 1995, Oak was renamed Java. Since then, Java is rising in popularity. Java Soft also sued Microsoft, for violating its Java license agreement. Microsoft wants to add Windows specific alterations to the Java language, which would blunt the "run anywhere" goal of Java.

Java Soft, which presents compatibility problems with existing web browsers and Virtual Machines are currently expanding Java. Its syntax is similar to C and C++, but it omits many of the features that make C and C++ complex, confusing, and unsafe. The Java platform was initially developed to address the problems of building software for networked consumer devices. It was designed to support multiple host architectures and to allow secure delivery of software components. To meet these requirements, compiled code had to survive transport across networks, operate on any client, and assure the client that it was safe to run.

The popularization of the World Wide Web made these attributes much more interesting. The Internet demonstrated how media-rich content could be made accessible in simple ways. Web browsers enabled millions of people to roam the Net and made Web surfing part of popular culture. At last there was a medium where what you saw and heard was essentially the same whether you were using a Mac, PC, or UNIX machine, and whether you were connected to a high-speed network or a slow modem.

Web enthusiasts soon discovered that the content supported by the Web's HTML document format was too limited. HTML extensions, such as forms, only highlighted those limitations, while making it clear that no browser could include all the features users wanted. Extensibility was the answer.

Sun's Hot Java browser showcases the interesting properties of the Java programming language and platform by making it possible to embed programs inside HTML pages.

These programs are transparently downloaded into the Hot Java browser along with the HTML pages in which they appear. Before being accepted by the browser, the programs are carefully checked to make sure they are safe. Like HTML pages, compiled programs are network- and host-independent. The programs behave the same way regardless of where they come from or what kind of machine they are being loaded into and run on.

Visitors to Web pages incorporating dynamic content can be assured that their machines cannot be damaged by that content. Programmers can write a program once, and it will run on any machine supplying a Java or Java 2 run time environment.

The Java language is very secure and platform independent when compared to alternative languages. Java's secret is the tightly integrated language model.

## **Java features**

### **Simple**

Java was designed to be easy for the professional programmer. It is easy to learn and can be used effectively. If you are an experienced C++ programmer, moving to Java will require very little effort.

### **Secure**

There is a concept of applets in Java which can be downloaded without fear of virus or malicious content, because the Java programs are confined to the Java execution environment and are not allowed to access other parts of the CPU.

### **Portable**

The Java programs called Applets run in the JVM (Java virtual machine) environment that is in every browser; therefore, the programs can run anywhere.

### **Object Oriented**

Java classes follow the OOPS concept of encapsulation, inheritance, and polymorphism.

### **Robust**

Garbage collection and exception handling make Java a robust language. In garbage collection, the user doesn't have to bother about memory allocation as, when the object is no longer in use, it is automatically deleted to release memory space.

### **Multithreaded**

A single-threaded application has one thread of execution running at all times, and such programs can do only one task at a time.

A multi-threaded application can have several threads of execution running independently and simultaneously. These threads may communicate and cooperate and will appear to be a single program to the user.

### **Interpreted**

The Java code is compiled into the byte code, which is the class file. The byte code is then interpreted to the machine language by the JVM environment.

### **Distributed**

Java handles the TCP/IP protocols, which makes it easier to use in the Internet. Some other features of Java programming

### **Encapsulation**

Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from interference and misuse. It is a protective wrapper that prevents the code and data from being arbitrarily accessed by other code defined outside the wrapper.

Encapsulation is the capability to represent, denote, and handle information at a higher level that is inherent to a computer or base language. Variables and methods are formerly known as instance variables and instance methods to distinguish from class variables and class methods.

### **Inheritance**

Inheritance is the process by which one object acquires the properties of another object.

Classes inherit state and behavior from their superclass. A class is a blueprint or prototype that defines the variables and methods common to all objects of a certain kind.

Object oriented systems allow classes to be defined in terms of other classes. For example, mountain bikes, racing bikes and tandems are all subclasses of the bicycle class. Similarly, the bicycle class is the superclass of mountain bikes, racing bikes and tandems.

Each subclass inherits state (in the form of variable declarations) from the superclass. Mountain bikes, racing bikes and tandems share some states: Cadence, speed, and the like. Also each subclass inherits methods from the superclass.

### **Benefits of Inheritance**

Subclasses provide specialized behaviors on the basis of common elements provided by the superclass. Through the use of inheritance programmers can reuse this code in the superclass many times.

### **Abstraction**

Abstraction is this process of categorizing data. Consider that a person is looking for a frame in an optician's shop. To be able to choose a frame from amongst the various types of frames available, he has to first identify the attributes he is looking for. Once he has identified the attributes, he has with him a category or class of frames. Similarly, to model any real life objects in OOPS an "object" has to be instantiated from a specific "class". This basic process of forming a class is known as "abstraction".

## **Chapter-3**

### **Project Description**

#### **3.1 Palette and Method Information**

##### **3.1.1 Palette**

###### **❖ Swing controls:-**

Text field- to create a display field

Button- to create a button

Label to display text

Panel to create division of container

##### **3.1.2 Method**

We have mainly used in our application two method under the java class library.

i.     getText() method

ii.    getSet() method

In the action Performed method, the input string is obtained using the getText() method of the JTextField class. Then the input text is converted to a numeric value and the text of the output label is set.

In the getSet() method set the value and perform the operation then show the display. If the operation is null then catch the hidden operation.

#### **3.2 Coding Information**

##### **3.2.1 Packages**

The packages we used in this project are-

```
import java.awt.Color;
import java.awt.Container;
import java.awt.Cursor;
import java.awt.Font;
import java.awt.Image;
import java.awt.Toolkit;
import java.io.FileOutputStream;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
```

##### **3.2.2 Button input operation**

How to get the button input by following way-

```

Btn.addActionListener(new ActionListener()
{
    Public void actionPerformed(ActionEvent e)
    {
        String name=nameT.getText();
        String roll=rollT.getText();
    }
});

```

### 3.2.3 Operator input operation

We have used to get sign input.

```

nameT.addActionListener(new ActionListener()
{
    Public void actionPerformed(ActionEvent e)
    {
        String name=nameT.getText();
        System.out.println(name);
    }
});

```

### 3.3 Instruction

Typing instruction to achieve the projects objectives-

- (a) At first show a make bill screen.
- (b) Then click the save as pdf button to save it pdf.

### 3.4 Conclusions

All the main themes of the project like coding information, instructions are added in this chapter.

## Chapter 4

### Interfaces of the application

#### 4.0 Interfaces

The interfaces is GUI form of the application.

#### 4.1 Bill Make Form

Here any shopkeeper make Bill of any sell product.

### Digital Shop

Name of Customer :  Save As PDF

Date : 2024-07-03

Name of Product	Received Money	Price	Back Money
Keyboard	1000	320	680

## 4.2 PDF file

### Digital Shop

---

Date :2024-07-03

**Name of Customer : Ray**

Shop Memo			
Name of Product	Receive Money	Price	Back Money
Keyboard	1000	320	680

## **Chapter 5**

### **Conclusion and Future Work**

#### **5.1 Limitation**

(a) The Application is Java based so, it will run on those machine where have installed Java.

#### **5.2 Future plan**

We have some future plans with our Projects. Now we will have to try to add print button.

#### **5.3 Suggestions**

- (a) Give users chance to add information and activities.
- (b) Use activate colors
- (c) Make it possible to maximize the screen
- (d) Provide help button in each section

#### **5.3 Conclusion**

We have implemented our project using object oriented concepts of Java language. Our honorable course teacher helped us to implement the project. We have learned about built in methods of java and NETBEANS (IDE) tools

# **End**