A

Project Report on

Technique for an Efficient Hybrid Shopping Mall

Submitted to the partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by:
Vishal Nainwaya (RA1411003030365)
Harshit Kalra (RA1411003030348)
Karishma Mahajan (RA1411003030366)

Supervised by:
Mr. Sunil Kumar
Assistant Professor (CSE)

SRM Institute of Science and Technology
(Deemed to be University u/s of UGC Act, 1956)
NCR Campus, Modinagar

2018

DECLARATION

We, Vishal Nainwaya (RA1411003030365), Karishma Mahajan (RA1411003030366) and Harshit Kalra (RA1411003030348) hereby declare that the work which is being presented in the project report " Technique for an Efficient Hybrid Shopping Mall " is the record of authentic work carried out by us during the period from January '18 to May '18 and submitted by us in partial fulfillment for the award of the degree "Bachelor of Technology in Computer Science and Engineering" to SRM IST, NCR Campus, Ghaziabad (U.P.). This work has not been submitted to any other University or Institute for the award of any Degree/Diploma.

Vishal Nainwaya (RA1411003030365)
Karishma Mahajan (RA1411003030366)
Harshit Kalra (RA1411003030348)

BONAFIDE CERTIFICATE

This is to certify that project Report entitled "Technique for an Efficient Hybrid Shopping Mall", which is submitted by Vishal (RA1411003030365), Karishma (RA1411003030366) and Harshit (RA1411003030348) in the partial fulfillment of the requirement for the award of degree B.Tech (CSE) of SRM Institute of Science and Technology, NCR Campus, Modinagar, Ghaziabad is a record of the candidate own work carried out by them under my own supervision.

……………………................	…...
(Signature)	(Signature)
Dr. R. P. Mahapatra	Dr. Jitendra Singh
HOD (CSE)	Supervisor
	Assistant Professor (Sr.G)

INTERNAL EXAMINER   EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We would also like to express our special thanks to Mr. Sunil Kumar, Assistant Professor (CSE) and Project Supervisor of SRM Institute of Science and Technology, NCR Campus, Modinagar, for their insight and knowledge on the subject, who imbued us with the feeling to work assiduously.

We feel compelled to articulate our thankfulness to Dr. Jitendra Singh and Dr. Rajesh Kumar Tyagi, Project Coordinators, SRM Institute of Science and Technology NCR Campus, Modinagar, for his enlightening dexterity which was a perennial source of inspiration.

We would like to express our hearty gratitude to Dr. M. K. Pandey, Director, SRM Institute of Science and Technology, NCR Campus, Modinagar, under whose auspices we were able to work on our project work.

Also, we extend our sincere regards and thanks to Dr. R. P. Mahapatra, Head of the Department, Computer Science and Engineering, & Dean Admission, SRM Institute of Science and Technology, NCR Campus, Modinagar, for his suggestions.

We are also indebted to all the teaching and non-teaching staff members of our college for helping us directly or indirectly by all means throughout the course of our study and project work.

Finally, we take this opportunity to thank our parents, family members and friends for their moral support and help and extended thanks to our well wishers.

Vishal Nainwaya (RA1411003030365)
Karishma Mahajan (RA1411003030366)
Harshit Kalra (RA1411003030348)

ABSTRACT

The combined version of online and offline shopping technique is the base objective of the project. In the working application of this idea, a facility of combined shopping and pay later facility is tried to be executed. Wherein, the user will be allowed to do shopping normally as done in an offline scenario but will only scan the NFC card for each item and after the completion of the shopping payment has to be made at the payment booths. At the payment booths the customer can modify the cart by viewing the bill and edit accordingly and thus a bill will be generated. After the generation of the bill the customers can collect their purchased items for the collection counter. Thus, this idea can provide an offline and online shopping experience to its customers. Here all the product that user wants to purchase (added into cart) will be directly available at the payment counter. Also user can compare the qualities of same product of various brands, thus making it user friendly. The report describes analysis, implementation and design of SMART MALLS that uses features of both offline and online shopping system.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Officially shopping malls are defined as “One or more buildings forming a complex of shops representing merchandisers, with interconnected walkways enabling visitors to walk from unit to unit.” Un-officially, they are the heart and soul of business communities, the foundation of retail economies, and a social sanctuary for teenagers everywhere. The shopping complexes and malls are defined under the offline shopping. In the recent years, online shopping has brought a revolutionary change in for the retail economies, and social fronts for the general public.

Since both of the shopping or business foundations have brought a revolutionary change in the society and function for the betterment of the society. Thus a motivation for developing a technique for providing the facilities and the pro’s of both online and offline shopping can bring about even more of a better change for the economies and the society. A system that combines both online and offline shopping can help the customers to find the best product alternatives, with the exclusive prices in comparison to various brands and merchandise.

Various US based malls have tried the various ideas for smart mall that possess facility like providing a filter for the various types of stores in the malls along with the GPS to each kind of a store.

Thus if an overall combined billing system can be brought into account it can improve the shopping experiences for the customers on the whole.

1.2 Current Shopping System

The current smart shopping systems includes “A building one or more buildings forming a complex of shops representing merchandisers, with interconnected walkways enabling visitors to walk from unit to unit”. A shopping malls contains various stores or Brand outlets combined together to provide a choice of options and a comparison of prices for the same item under one roof. In only some parts of the world, mostly the foreign countries include a system based on GPS which finds the customer the way to a specific filtered store in the mall, that has helped the customers to have a convenient shopping experience. Mainly shopping techniques under the current system include the Online and Offline shopping methods.

1.2.1 Online Shopping

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet user a web browser. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product’s availability and pricing at different E-retailers. Customers can shop online using a range of different computer and devices, including desktop computers, laptops, tablet computers and smartphones. Online stores typically enable shoppers to use “search” features to find specific models, brands or items. Online customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Internet-enabled debit card, or a service such as PayPal. For physical products (e.g. , paperback books or clothes) , the e-retailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-retailer typically sends the file to the customer over the Internet. Some of the large online retailers include Amazon.com, Myntra.com, eBay.com. Customers are attached to online shopping not only because of high levels of convenience, but also because of broader selections, competitive pricing, and greater access to information. Business organizations seeks to offer online shopping not only because it is of much lower cost compared to bricks and mortar stores, but also because it offers access to a worldwide market, increases customer value, and builds sustainable capabilities.

Also in the online shopping the customer doesn’t need to consume his energy by going out to the stores and also it saves his time and cost of travelling.

1.2.2 Offline Shopping

A retailer or shop is a business that presents a selection of goods and offers to trade or sell them to customers for money or other goods. Shopping is an activity in which a customer browses the available goods or services presented by one or more retailers with the intent to purchase a suitable selection of them. In some context offline shopping is considered as a leisure activity as well as an economic one. The shopping experience may vary, based on a variety of factors including how the customer is treated, convenience, the type of goods being purchased, and the mood. The shopping experience can also be influenced by other shoppers. For example, research from a field experiment found that male and female shoppers who were accidentally touched from a field experiment found that male and female shoppers who were accidentally touched from behind by other shoppers left a store earlier than people who had not been touched from behind by other shoppers left a store earlier than people who had not been touched and evaluated brands more negatively, resulting in the Accidental Inter-personal Touch Effect.

1.3 Problem Description

When we see the disadvantages of offline and the online shopping combined we can find a collaborative solution as the disadvantage for one serves as the advantage for the other, thus nullifying the entire drawback for either of the two.

1.3.1 Online vs. Offline Shopping

The various drawbacks of online and the offline can be stated as

- **Disadvantages for Offline Shopping and it’s Solution In Online Shopping**

1. There is a limitation of availability of offline shops and each shop has a limitation of space so you have no such choice of wide variety of availability.

   Solution: You find less variety or options for the products or services when buying offline. Also there is a wider range of choices to choose in online shopping. One can find an endless shopping website and find a huge option for a single item you want to buy.

2. As far as money is concerned, the price of a product may be costlier in an offline shopping.

   Solution: On an online shopping portal prices are highly competitive and also one can find a variety of discount and other similar offer, the price in an online shopping will be lower one in an online shopping.

- **Disadvantages for Online Shopping and It's Solution In Offline Shopping**

1. In Online Shopping The Actual Appearance And The Visual Acceptance On An Individual Is Not Righteous. Also The Credibility For The Product Cannot Be Assured.

   Solution: A Check On The Credibility Can Easily Be Done When The Customer Is In Direct Contact With The Actual Seller.

2. Internet Can Have Various Defaulters And False Sellers, Which Seems Authentic But Is Not In Real Time.

1.4 About the Project

The combined version of online and offline shopping technique is the base objective of the project. In the working application of this idea, a facility of combined shopping and pay later facility is tried to be executed. Wherein, the user will be allowed to do shopping normally as done in an offline scenario but will only scan the NFC card for each item and after the completion of the shopping payment has to be made at the payment booths. At the payment booths the customer can modify the cart by viewing the bill and edit accordingly and thus a bill will be generated. After the generation of the bill the customers can collect their bill items for the collection counter.

Thus, this idea can provide an offline and online shopping experience to its customers.

1.5 Objective

The objective of the "Technique for an Efficient Hybrid Shopping Mall", is to make the shopping experience more reliable and convenient by giving them an comparison amongst various merchandises in a real time environment.

CHAPTER 2

INTRODUCTION TO EMBEDDED SYSTEM

2.1 What Is Embedded System

A computer system which performs a specific function is called embedded system. It can be electrical system, mechanical system or combination of both. Output of embedded system depends on the real time constraints. It is a complete device which includes both hardware and mechanical parts. Embedded systems are used in many devices today. More than Ninety percent of microcontrollers have components of embedded systems.

Nowadays embedded systems are mostly based on microcontrollers those have CPUs with integrated memory or peripheral interfaces, but ordinary microprocessors are also common, especially in complex systems. Processors can be general purpose processors or it can be processors which are specialized in certain class of computations. Even custom designed are also used for the application at hand. DSP (digital signal processor) is a common standard class of dedicated processors.

These embedded systems are designed to do a specific task only; they cannot be used for multiple tasks such as general purpose computer systems. Some systems have low or no requirements and are just made to reduce the effective cost, but some systems perform on real time constraints for reason such as safety and usability.

Inside embedded systems single or multiple processing cores are used which can be in the form of microcontroller or DSP (digital signal processors), field-programmable gate arrays (FPGA), application-specific integrated circuits (ASIC) and gate arrays. These processing components are integrated with components dedicated to handling electric and/or mechanical interfacing.

Any embedded system's key feature is to perform a specific task for which it is made and it requires a strong general purpose processor.

Router and switch for example are embedded systems but a general purpose computer uses a proper operating system for routing functionality. However, routers function more efficiently than operating system based computers for routing functionalities.

Commercial embedded systems range from digital watches and MP3 players to giant routers and switches. Complexities vary from single processor chips to advanced units with multiple processing chips.

2.2 Embedded System Design Cycle

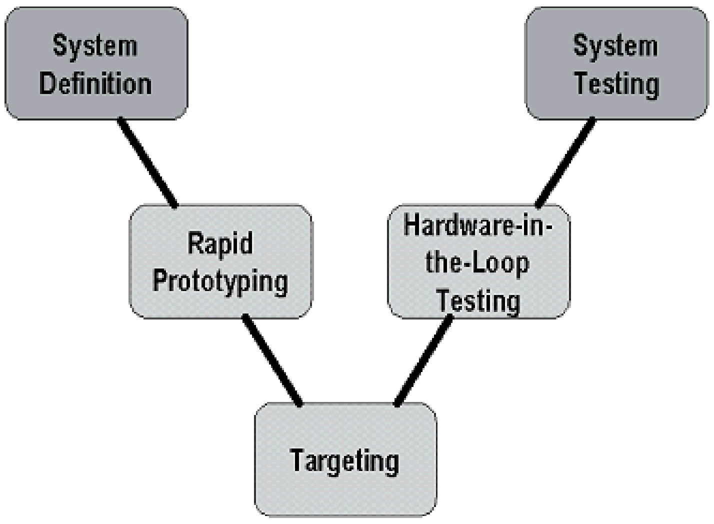


Figure 2.1: 'V' diagram embedded system

The above V diagram is same as waterfall model but the difference is that it keeps focus on testing activities upfront rather than testing later in the life cycle. In each stage of development activity test and validation are done at the same level itself. Each test phase is recognized with its matching progress phase as we see in the following figure.

In the diagram, we have:-

- Requirements ↔ System/Functional Testing
- High-Level Design ↔ Integration Testing
- Detailed Design ↔ Unit Testing

These extend from specification to customer delivery and post delivery support. If one follows the sequence down the left hand side of the drawing, one can see that the specification and design procedure utilizes a top down model whereas implementation and test process from a bottom up model as is reflected on the right hand side of the drawing.

And that each verification step tests a more complete implementation of the system against the requirements at that phase. The progress concludes the design and connected test portion of the development cycle of the system with both verification and a validation test.

2.3 Characteristics of an Embedded System

The embedded systems are designed to do a specific task only; they cannot be used for multiple tasks such as general purpose computer systems. Some systems have low or no requirements and are just made to reduce the effective cost, but some systems perform on real time constraints for reason such as safety and usability.

Embedded systems are not always standalone devices rather many large devices have small parts that serves more general purpose. For example, the Gibson Robot Guitar features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. Likewise, in an automobile an embedded system provides a specific function as a subsystem of the car itself.

The instructions written for embedded systems are called as a firmware, and are stored in ROM (read-only memory) or pendrives (flash memory chips). They need limited computer hardware resources to run such as little memory, a keyboard or screen.

a) Embedded systems perform a specific function and are single functioned; application is known apriority and the programs are executed iteratively.

b) The main importance of embedded systems is efficiency. These systems are optimized for low energy consumption, less code size, less execution time, weight & dimensions, and low cost.

c) Embedded systems are typically designed in such a way that they can perform robustly to real time constraints. A real time system act according to the previous reaction and the changes that occur due to new inputs within the time specified to it by the environment. In real time systems output are needed to appear quickly and accurate.

d) Embedded systems mostly in real time are made to interact with the external world with the help of sensors and actuators and hence are called reactive systems. A reactive system is in repetitive interaction with the real time constraint from external world and executes at a pace dedicated by that environment.

e) Embedded systems have very minimum or no user interface.

2.3.1 User Interface

There are embedded systems which have no user interface at all and theses systems are intended to do specific/dedicated task only. Also there are embedded systems which are specialized in doing certain functions that include complex user interface that resemble to today's computers. Common embedded devices use LEDs, graphic or character LCDs (HD44780_LCD for example) and buttons with a simple menu.

More complicated devices having a graphical interface with touch screens provide flexibility while minimizing space used. The functioning of the buttons can be changed with the screen. Handheld systems generally have a joystick with a screen for pointing devices.

Some systems offer user interface remotely by using USB (serial connection) or Ethernet (Network) connection. This gives several advantages such as it extends the capabilities and features of embedded system, it decreases (avoid) the cost of a display, it allows us to make (build) a more user friendly interface. A combination of embedded we server with embedded devices (IP camera or network router) is a good example of this. So a computer can be connected to the device for user interface to display web browser and need no extra software to be installed.

2.3.2 Processors in Embedded Systems

Embedded systems are categorised into two ways with reference to processors. Ordinary microprocessors (µP) which uses a separate IC (integrated circuit) for memory and peripherals. Microcontrollers (µC) that have on-chip peripherals and thus reduce power consumption, size and cost. In comparison to the personal computers, many diverse CPU architectures are used as software is custom developed for an application and is not a service product installed by the end user. RISC and non-RISC processors are found. Word lengths may vary from 4-bit to 64-bits and more, although the most typical remain 8/16-bit. Architecture is available in large number of different variants and shapes, and is manufactured by various companies.

Various microcontrollers have been developed for the use of embedded systems. Microcontrollers are very common in embedded systems. General purpose microprocessors are also worn in embedded systems but generally, require more support circuitry than microcontrollers.

2.3.3 Ready-made Computer Boards

PC/104 and PC/104+ are examples of standards for *ready-made* computer boards intended for small, low-volume embedded and ruggedized systems, mostly x86-based. These are often physically small compared to a standard PC, although still quite large compared to most simple (8/16-bit) embedded systems. They generally use DOS, Linux, NetBSD, or an embedded real-time operating system such as MicroC/OS-II, QNX or VxWorks. Sometimes these boards use non-x86 processors.

Some applications doesn't need small size or power efficiency, in such applications components compatible with those used in general purpose x86 personal computers can be used. Various board namely the VIA EPIA range helps us to bridge the gap by being compatible but highly integrated, smaller in size and other properties making them striking to embedded engineers (developers). The advantage of this approach is that low-cost components can be used all along with the identical software development tools used for common software development.

These are also called embedded system as at the end these are integrated into larger devices so that the dedicated function can be achieved and fulfil a single role. ATMs and arcade machines are good examples of such devices which have code particular to the application.

However, generally ready-made embedded systems do not use the ISA or PCI buses and their boards are not PC-centred. When a system-on-a-chip processor is concerned, there is advantage of having a standardized bus connecting separate components. The environment for both software and hardware tools may be very different.

One ordinary design method uses a small system unit, perhaps the size of a business card, holding high density BGA chips such as an ARM-based system-on-a-chip processor and peripherals, external flash memory for storage, and DRAM for runtime memory. The unit vendor will provide boot software and confirm that there is a selection of operating systems, usually including Linux and some real time choices. Such modules can be manufactured in huge volume, by organizations recognizable with their specialized testing issues, and combined with much lower volume custom mainboards with application-specific external peripherals.

The implementation of embedded systems has highly developed so that they can easily and quickly be implemented with ready-made boards that are available and accepted worldwide platforms. These platforms include Arduino and Raspberry Pi but are not limited to only this.

2.3.4 ASIC and FPGA Solutions

Generally embedded system with very-high-volume is the system on a chip (SoC) which contains the entire system consisting of numerous processors, multipliers, caches and interfaces on a single chip. SoCs can be implemented as an application-specific integrated circuit (ASIC) or using a field-programmable gate array (FPGA).

2.3.5 Peripherals

Embedded systems talk with the outside world via peripherals, such as:

1. Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485, etc.

2. Synchronous Serial Communication Interface: I2C, SPI, SSC and ESSI (Enhanced Synchronous Serial Interface)
3. Universal Serial Bus (USB)
4. Multi Media Cards (SD cards, Compact Flash, etc.)
5. Networks: Ethernet, Lan Works, etc.
6. Fieldbuses: CAN-Bus, LIN-Bus, PROFIBUS, etc.
7. Timers: PLL(s), Capture/Compare and Time Processing Units
8. Discrete IO: aka General Purpose Input/Output (GPIO)
9. Analog to Digital/Digital to Analog (ADC/DAC)
10. Debugging: JTAG, ISP, ICSP, BDM Port, BITP, and DB9 ports.

2.3.6 Tools

Embedded system designer uses compiler, assemblers, and debuggers to create system software as used by other software developers.

However, they may also use some more specific tools:

1. In circuit debuggers or emulators.
2. Utilities to add a checksum or CRC to a program, so the embedded system can check if the program is valid.
3. For systems using digital signal processing, developers may use a math workbench to simulate the mathematics.
4. System level modelling and simulation tools help designers to construct simulation models of a system with hardware components such as processors, memories, DMA, interfaces, buses and software behaviour flow as a state diagram or flow diagram using configurable library blocks. Simulation is conducted to select right components by performing power vs. performance trade-off, reliability analysis and bottleneck analysis. Typical reports that helps designer to make architecture decisions includes application latency, device throughput, device utilization, power consumption of the full system as well as device-level power consumption.
5. A model-based development tool creates and simulates graphical data flow and UML state chart diagrams of components like digital filters, motor controllers, communication protocol decoding and multi-rate tasks.
6. Custom compilers and linkers may be used to optimize specialized hardware.

7. An embedded system may have its own special language or design tool, or add enhancements to an existing language such as Forth or Basic.
8. Another alternative is to add a real-time operating system or embedded operating system
9. Modelling and code generating tools often based on state machines

Software tools can come from several sources:

1. Software companies that specialize in the embedded market
2. Ported from the GNU software development tools
3. Sometimes, development tools for a personal computer can be used if the embedded processor is a close relative to a common PC processor

As soon as the embedded system is develop each time it becomes more and more complex. And as the complexity of embedded systems increases, higher level tools and operating systems are required into machinery. For example, cell phones, personal digital assistants and other consumer computers often need significant software that is purchased or provided by a person other than the manufacturer of the electronics. In these systems, an open programming environment such as Linux, NetBSD, OSGi or Embedded Java is required so that the third-party software provider can sell to a large market.

Embedded systems are common in fields such as consumer, cooking, industrial, automotive, medical applications etc.. Some examples of embedded systems are MP3 players, mobile phones, videogame consoles, digital cameras, DVD players, and GPS. Household appliances, such as microwave ovens, washing machines and dishwashers, include embedded systems to provide flexibility and efficiency.

2.4 Classification of Embedded System

Embedded systems are classified into four categories based on their performance and functional requirements:

1. Stand-alone embedded systems
2. Real time embedded systems
3. Networked embedded systems

4. Mobile embedded systems

Embedded Systems are classified into three types based on the performance of the microcontroller such as:

1. Small scale embedded systems
2. Medium scale embedded systems
3. Sophisticated embedded systems

2.4.1 Stand Alone Embedded Systems

Stand alone embedded systems doesn’t need any host system like a computer, it works on its own. It receives input from the input ports be it analog or digital and calculates and converts the data and provide the resulting output (data) through display device or any connected devices. Mp3 players, digital cameras, video game consoles, microwave ovens and temperature measurement systems are example of standalone embedded systems.

2.4.2 Real Time Embedded Systems

A system which provides a required o/p by considering the previous outputs and present input in specific time. These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems.

2.4.3 Networked Embedded Systems

Network embedded systems are system which are related to a network to access the resources. The network can be LAN, WAN or the internet. The connection can be any wired or wireless. It is fastest growing system in embedded system applications. The embedded web server is such type of system in which all embedded devices are connected to a web server and accessed and controlled by a web browser.

2.4.4 Mobile Embedded Systems

Mobile embedded systems are mostly focused toward portable embedded devices. Limitation of memory and other resources is the basic limitation of these devices. Cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc. are example of Mobile Embedded system.

2.4.5 Small Scale Embedded Systems

These embedded systems are designed with a single 8 or 16-bit microcontroller that can be activated by a battery. For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).

2.4.6 Medium Scale Embedded Systems

These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs. These types of embedded systems have both hardware and software complexities. For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

2.4.7 Sophisticated Embedded Systems

These types of embedded systems have enormous hardware and software complexities that may need ASIPs, IPs, PLAs, scalable or configurable processors. They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

2.5 Applications

1. Military and aerospace embedded software application.
2. Communication application.
3. Industrial automation and process control software.
4. Mastering the complexity of application.
5. Reduction of product design time.
6. Real time processing of ever increasing amounts of data.
7. Intelligent, autonomous sensors.

CHAPTER 3

HARDWARE SPECIFICATION

3.1 Circuit Diagram

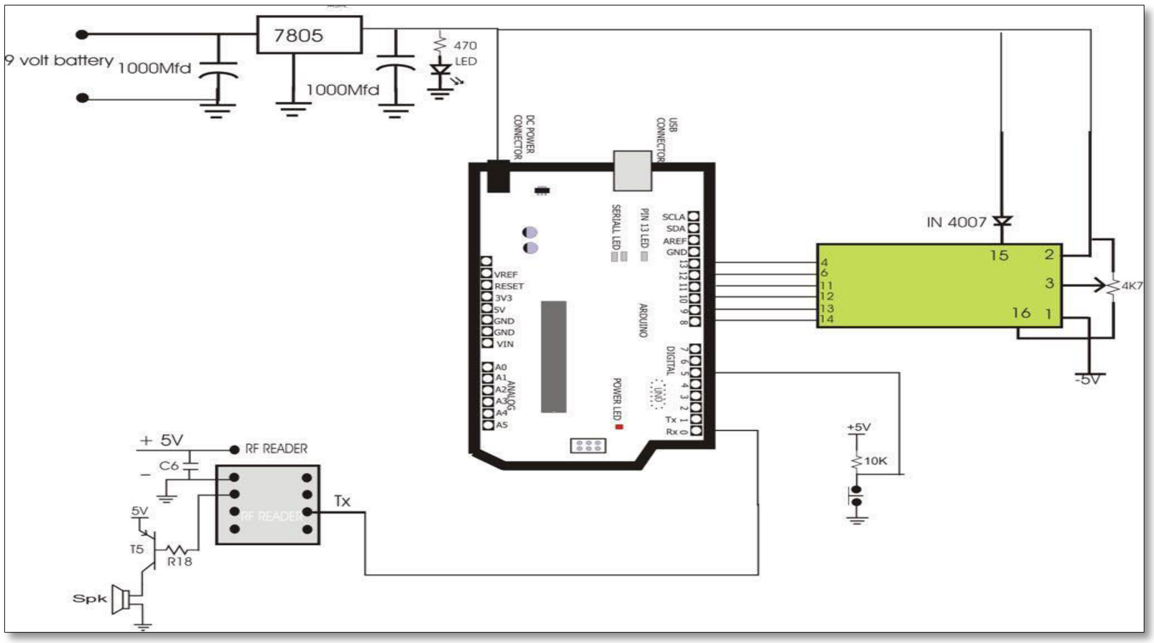


Figure 3.1: Circuit Diagram

3.2 Hardware Requirement

1. Arduino Board
2. Voltage regulator
3. Resistor
4. LCD
5. Capacitor
6. Battery
7. Buzzer
8. Reader module

3.2.1 Liquid Crystal Display

LCD display are everywhere around us such as Computers, calculators, television sets, mobile phones, digital watches use same kind of display to display the time. A liquid-crystal display (LCD) is a flat-panel display or other electronic modulated optical device that uses the light-modulating properties of liquid crystals. LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. Liquid crystal doesn’t emit light directly, instead a backlight or reflector is used to produce images in color or monochrome. Same basic technology is used here, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements. LCD screens do not use phosphors and hence they do not suffer image burn when a static image is displayed on a screen for a long time. The LCD screen uses very less energy and can be disposed of more safely than a CRT can. LCD can operate on low power which enables it to be used in battery powered electronic devices.

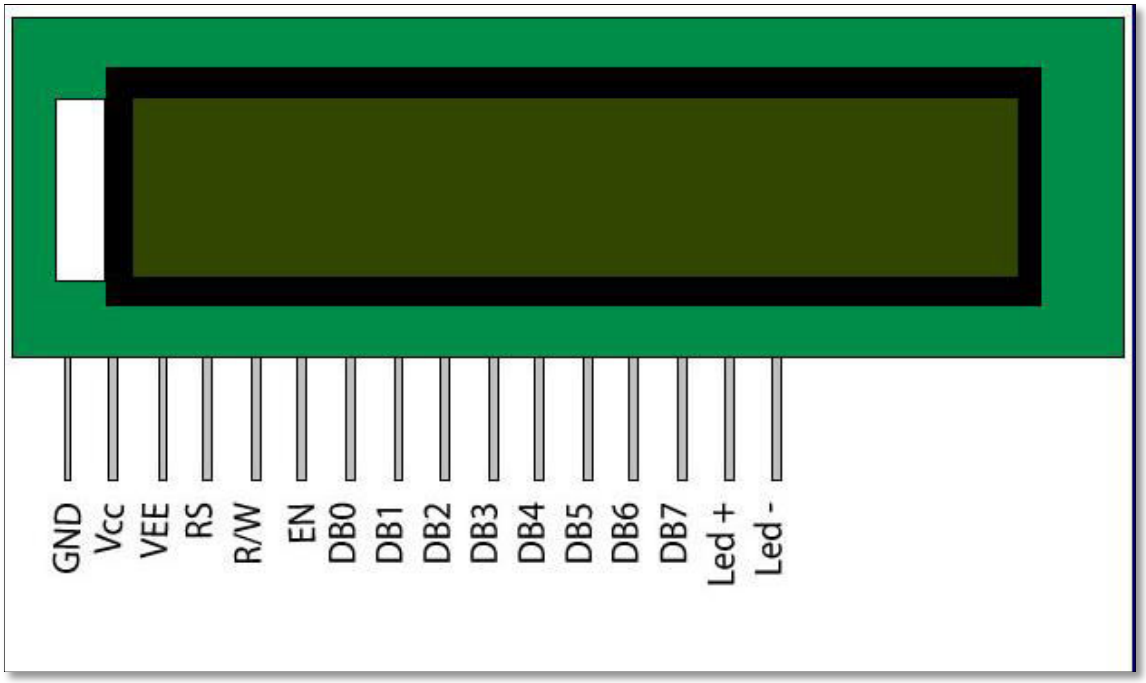


Figure 3.2: LCD Pin Diagram

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Table 3.1: LCD Pin Description

The 16×2 LCD display is a very basic module generally used in circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: it is easy to program; LCDs are economical; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

3.2.2 Voltage Regulator

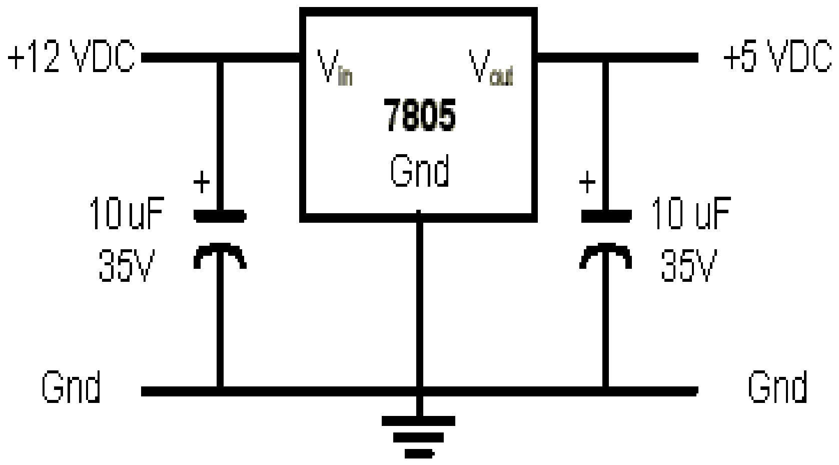


Figure 3.3: Circuit Diagram Voltage Regulator

The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. A voltage regulator is an electronic circuit that provides a stable DC voltage independent of the load current, temperature and AC line voltage variations.

A voltage regulator may use a simple feed-forward design or may include negative feedback. It may use an electromechanical mechanism, or electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages. **7805** is a **voltage regulator** integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

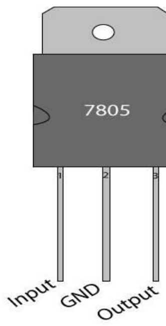


Figure 3.4: Pin Diagram Voltage Regulator

Pin No	Function	Name
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

Table 3.2: Pin Description Voltage Regulator

Features:

- Output Current up to 1A.
- Output voltage of 5V.
- Thermal overload protection.
- Short circuit protection.
- Output transistor safe operating area protection.

3.2.3 Reader Module

EM-18 RFID reader is one of the commonly used RFID reader to read 125KHz tags. It features low cost, low power consumption, small form factor and easy to use. The EM-18 RFID Reader module generates and radiates RF Carrier Signals of frequency 125 KHz through its coils. When a 125KHz Passive RFID Tag (have no battery) is brought in to this field, will get energized from it. These RFID Tags are usually made using a CMOS IC EM4102. It gets enough power and master clock for its operations from the electromagnetic fields produced by RFID Reader. When a RFID Tag is bring in to the field of EM-18 RFID Reader, it will read its tag number and give output via TX terminal. The BEEP terminal will become LOW to indicate valid tag detection.

Figure 3.5: EM-18 Reader Module

Features

1. Low cost.
2. Low power.
3. Form factor being small.
4. Usability is easy.
5. Direct interfacing with UART, PC is possible using RS232.
6. Serial and TTL output.
7. Excellent read performance without an external circuit.
8. Two RFID cards.
9. Cost-effective and compact size.

Specifications

1. Read distance 10cm
2. Current <50mA
3. Operating frequency 125khz
4. Parameter Value
5. Operating Voltage 5v

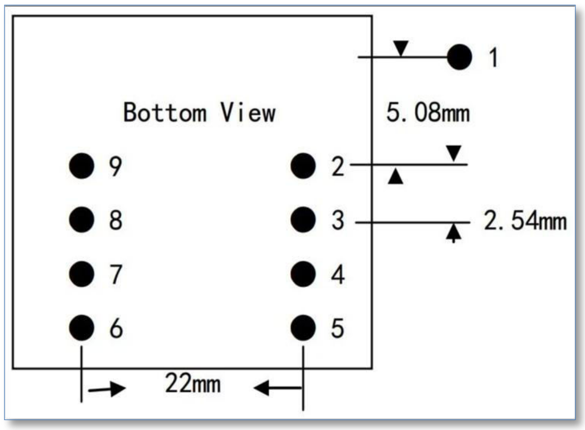


Figure 3.6: Pin Diagram Reader Module

Sr. No.	**VCC**	**5V**
1	GND	GND
2	BEEP	BEEP AND LED
3	ANT	NO USE
4	ANT	NO USE
5	SEL	HIGH IS RS232, LOW IS WEIGAND
6	RS232	RS232
7	D1	WEIGAND DATA 1
8	D0	WEIGAND DATA 0

Table 3.3: Pin Description Reader Module

RS232 interface format:

10 ASCII DATA (card no.) + 2 ASCII DATA (XOR result)
E.g. Card number is 4500C5D1E9B8 read from reader then the card number on card will be as below. Preamble 00C5D1E9 value in Hex = 12964329. B8 is XOR value for (45 XOR 00 XOR C5 XOR D1 XOR E9).
Hence number on the card is 0012964329.

3.2.4 Near Field Communication

Near-field communication (**NFC**) is a set of communication protocols that enable two electronic devices, one of which is usually a portable device such as a Smartphone, to establish communication by bringing them within 4 cm (1.6 in) of each other.

NFC devices are used in contactless payment systems, similar to those used in credit cards and electronic ticket smartcards and allow mobile payment to replace/supplement these systems. This is sometimes referred to as NFC/CTLS (Contactless) or CTLS NFC. NFC is used for social networking, for sharing contacts, photos, videos or files. NFC-enabled devices can act as electronic identity documents and keycards. NFC offers a low-speed connection with simple setup that can be used to bootstrap more capable wireless connections.

NFC-enabled portable devices can be provided with application software, for example to read electronic tags or make payments when connected to an NFC-compliant apparatus. Earlier close-range communication used technology that was proprietary to the manufacturer, for applications such as stock ticket, access control and payment readers.

Each full NFC device can work in three modes:

NFC card emulation

Enables NFC-enabled devices such as smart phones to act like smart cards, allowing users to perform transactions such as payment or ticketing.

NFC reader/writer

Enables NFC-enabled devices to read information stored on inexpensive NFC tags embedded in labels or smart posters.

NFC peer-to-peer

Enables two NFC-enabled devices to communicate with each other to exchange information in an adhoc fashion.

NFC tags are passive data stores which can be read, and under some circumstances written to, by an NFC device. They typically contain data (as of 2015 between 96 and 8,192 bytes) and are read-only in normal use, but may be rewritable. Applications include secure personal data storage (e.g. debit or credit card information, loyalty program data, personal identification numbers (PINs), contacts). NFC tags can be custom-encoded by their manufacturers or use the industry specifications.

3.2.5 Buzzer

A **buzzer** or **beeper** is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

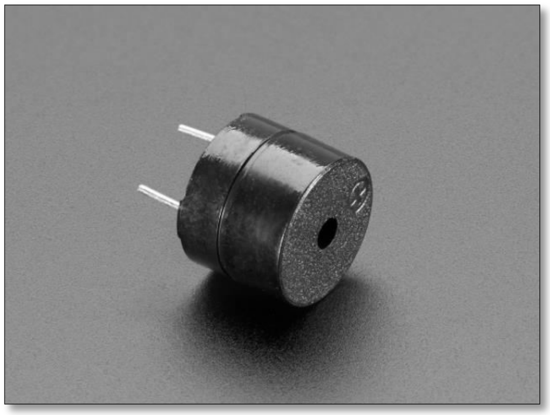


Figure 3.7: Buzzer

3.3 Arduino Board

Arduino board is an open-source platform used to make electronics projects. It consists of both a microcontroller and a part of the software or Integrated Development Environment (IDE) that runs on your PC, used to write & upload computer code to the physical board. The platform of an Arduino has become very famous with designers or students just starting out with electronics, and for an excellent cause.

Unlike most earlier programmable circuit boards, the Arduino does not require a separate part of hardware in order to program a new code onto the board you can just use a USB cable. As well, the Arduino IDE uses a basic version of C++, making it simpler to learn the program. At last, Arduino board offers a typical form factor that breaks out the functions of the microcontroller into a more available package. Arduino board has been used for making different engineering projects and different applications. The Arduino software is very simple to use for beginners, yet flexible adequate for advanced users. It runs windows, Linux and Mac. Teachers and students in the schools utilize it to design low cost scientific instruments to verify the principles of physics and chemistry. There are numerous other microcontroller platforms obtainable for physical computing.

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases.

The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Figure 3.8: Arduino Board

3.3.1 NEED FOR ARDUINO

Why is there a need to use Arduino in specific? Or What makes it different from others? Massimo Banzi, a Co-founder of Arduino mentions some very important reasons for this question.

1. **Active User Community:** A group of people using a similar product can hold posted message conversations and share their experiences or solve the problems of the other users in the communities with their own experiences.
2. **Growth of Arduino:** Arduino was developed with intent to provide an economical and trouble-free way for hobbyists, students and professionals to build devices that interact with their situation using sensors and actuators. This makes it perfect for newcomers to get started quickly.

3. **Inexpensive Hardware:** Since Arduino is an open source platform the software is not purchased and only the cost of buying the board or its parts is incurred, thus making it very cheap. The hardware designs are also available online for free from its official website.
4. **Arduino Board as a Programmer:** To make Arduino board function easy and also making it available everywhere these boards come with a USB cable for power requirements as well as functioning as a programmer.
5. **Multi-platform Environment:** The Arduino IDE is capable of running on a number of platforms including Microsoft, Linux and Mac OS X making the user community even larger.

Arduino Board	**Processor**	**Memory**	**Digital I/O**	**Analogue I/O**
Arduino Uno	16Mhz ATmega328	2KB SRAM, 32KB flash	14	6 input, 0 output
Arduino Due	84MHz AT91SAM3X8E	96KB SRAM, 512KB flash	54	12 input, 2 output
Arduino Mega	16MHz ATmega2560	8KB SRAM, 256KB flash	54	16 input, 0 output
Arduino Leonardo	16MHz ATmega32u4	2.5KB SRAM, 32KB flash	20	12 input, 0 output

Table 3.4: Types of Arduino Board

3.3.2 Hardware of Arduino Board

The Arduino Development Board consists of many components that together makes it work. Here are some of those main component blocks that help in its functioning:

1. **Microcontroller:** This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the

peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

2. **External Power Supply:** This power supply is used to power the Arduino development board with a regulated voltage ranging from 9 – 12 volts. International Journal of Control, Automation, Communication and Systems (IJCACS), Vol.1, No.2, April 2016.
3. **USB plug:** This plug is a very important port in this board. It is used to upload (burn) a program to the microcontroller using a USB cable. It also has a regulated power of 5V which also powers the Arduino board in cases when the External Power Supply is absent.
4. **Internal Programmer:** The developed software code can be uploaded to the microcontroller via USB port, without an external programmer.
5. **Reset button:** This button is present on the board and can be used to resets the Arduino microcontroller.
6. **Analog Pins:** There are some analog input pins ranging from A0 – A*7 (typical)*. These pins are used for the analog input / output. The no. of analog pins also varies from board to board.
7. **Digital I/O Pins:** There are some digital input pins also ranging from 2 to *16 (typical)*. These pins are used for the digital input / output. The no. of these digital pins also varies from board to board.
8. **Power and GND Pins:** There are pins on the development board that provide 3.3, 5 volts and ground through them

CHAPTER 4

SOFTWARE SPECIFICATION

4.1 Java

Java is a programming language and a platform. Java is a high level, robust, secured and object-oriented programming language. Any hardware or software environment in which a program runs is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plug-in for applets).

The latest version is Java 10, released on March 20, 2018, which follows Java 9 after only six months in line with the new release schedule. Java 8 is still supported but there will be no more security updates for Java 9. Versions earlier than Java 8 are supported by companies on a commercial basis; e.g. by Oracle back to Java 6 as of October 2017 (while they still "highly recommend that you uninstall" pre-Java 8 from at least Windows computers).

4.1.1 Types of Java Applications

There are mainly 4 types of applications that can be created using java programming:

1) **Standalone Application**

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Example of standalone applications are: Media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.

2) **Web Application**

An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, spring, hibernate, jsf etc. technologies are used for creating web applications in java.

3) **Enterprise Application**

An application that is distributed in nature, such as banking applications etc. is called enterprise application. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

4) **Mobile Application**

An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

4.1.2 NetBeans

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 Microsoft Windows, macOS, Linux and Solaris. In addition to Java development languages like PHP, C, C++ and HTML5, Javadoc and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers.

The NetBeans Team actively supports the product and seeks feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback.

The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signedupgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

1. User interface management (e.g. menus and toolbars)
2. User settings management
3. Storage management (saving and loading any kind of data)
4. Window management
5. Wizard framework (supports step-by-step dialogs)
6. NetBeans Visual Library
7. Integrated development tools

4.1.2.1 NetBeans IDE

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 mobileapplications) out of the box.

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: 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[15] The NetBeans Community blog has announced that Oracle is proposing to entrust the development of the NetBeans platform and IDE to the Apache Foundation to "open up the government model," reaffirming its commitment to the project. NetBeans is currently submitted as a Proposal to Apache, and it will enter incubation if accepted.

4.1.3 Jar File

A JAR (Java Archive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc.) into one file to distribute application software or libraries on the Java platform.
In simple words, a JAR file is a file that contains compressed version of .class files, audio files, image files or directories. We can imagine a .jar files as a zipped file(.zip) that is created by using WinZip software.

Even , WinZip software can be used to used to extract the contents of a .jar . So you can use them for tasks such as lossless data compression, archiving, decompression, and archive unpacking.

JAR is:

1) the only archive format that is cross-platform
2) the only format that handles audio and image files as well as class files
3) backward-compatible with existing applet code
4) an open standard, fully extendable, and written in java
5) the preferred way to bundle the pieces of a java applet.

4.2 JDBC

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java

Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

Sun Microsystems released JDBC as part of Java Development Kit (JDK) 1.1 on February 19, 1997. Since then it has been part of the Java Platform, Standard Edition (Java SE).The JDBC classes are contained in the Java package java.sql and javax.sql.

Starting with version 3.1, JDBC has been developed under the Java Community Process. JSR 54 specifies JDBC 3.0 (included in J2SE 1.4), JSR 114 specifies the JDBC Rowset additions, and JSR 221 is the specification of JDBC 4.0 (included in Java SE 6). JDBC 4.1, is specified by a maintenance release 1 of JSR 221 and is included in Java SE 7. JDBC 4.2, is specified by a maintenance release 2 of JSR 221 and is included in Java SE 8. The latest version, JDBC 4.3, is specified by a maintenance release 3 of JSR 221 and is included in Java SE 9.

4.2.1 JDBC Driver

JDBC Driver is a software component that enables java application to interact with the database. There are 4 types of JDBC drivers:-

1. JDBC-ODBC bridge driver
2. Native-API driver (partially java driver)
3. Network Protocol driver (fully java driver)
4. Thin driver (fully java driver)

4.2.1.1 JDBC-ODBC bridge driver

The JDBC-ODBC bridge driver uses ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. This is now discouraged because of thin driver.

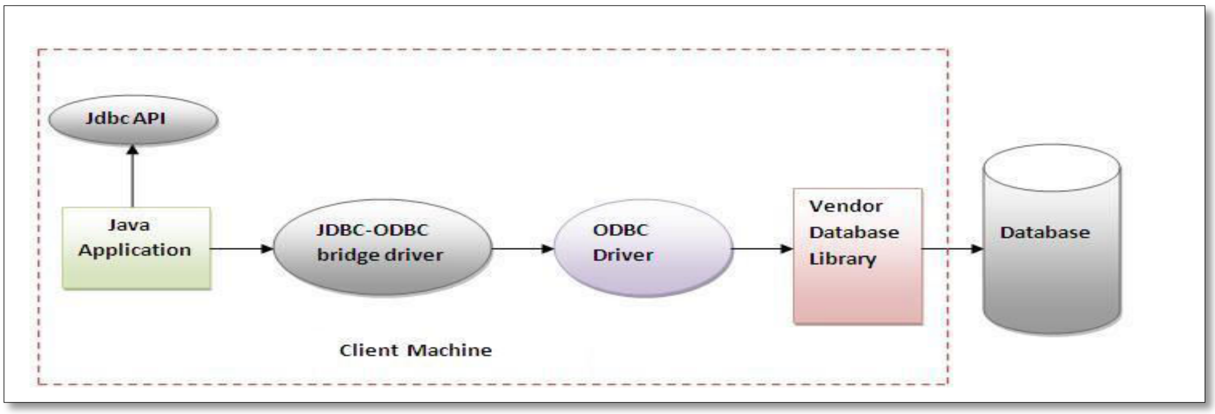


Figure 4.1: JDBC-ODBC Driver

Advantages:

1) Easy to use.
2) Can be easily connected to any database.

Disadvantages:

1) Performance degraded because JDBC method call is converted into the ODBC function calls.
2) The ODBC driver needs to be installed on the client machine.

4.2.1.2 Native-API driver

The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in java.

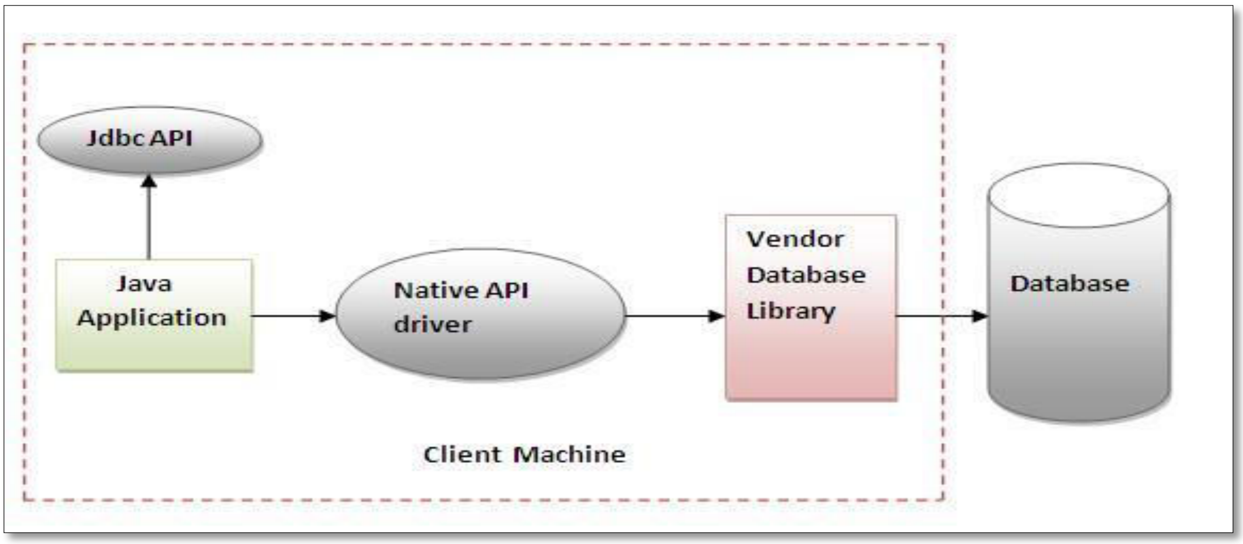


Figure 4.2: JDBC Native API

Advantage:

1) Performance upgraded than JDBC-ODBC bridge driver.

Disadvantage:

1) The Native driver needs to be installed on the each client machine.
2) The Vendor client library needs to be installed on client machine.

4.2.1.3 Network Protocol driver

The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. It is fully written in java.

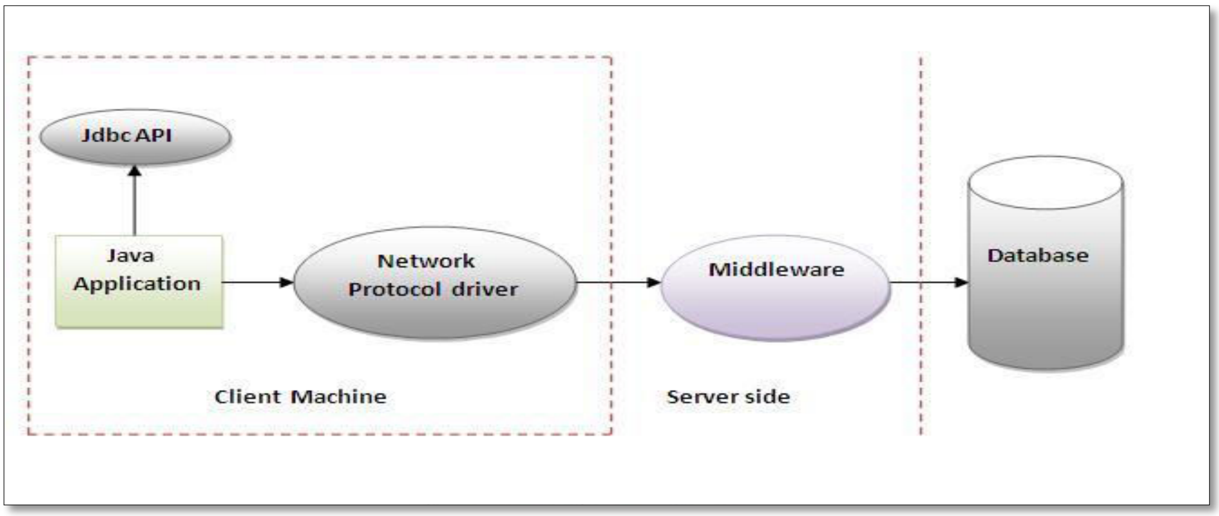


Figure 4.3: JDBC Network Protocol Driver

Advantage:

1) No client side library is required because of application server that can perform many tasks like auditing, load balancing, logging etc.

Disadvantages:

1) Network support is required on client machine.
2) Requires database-specific coding to be done in the middle tier.
3) Maintenance of Network Protocol driver becomes costly because it requires database-specific coding to be done in the middle tier.

4.2.1.4 Thin driver

The thin driver converts JDBC calls directly into the vendor-specific database protocol. That is why it is known as thin driver. It is fully written in Java language.

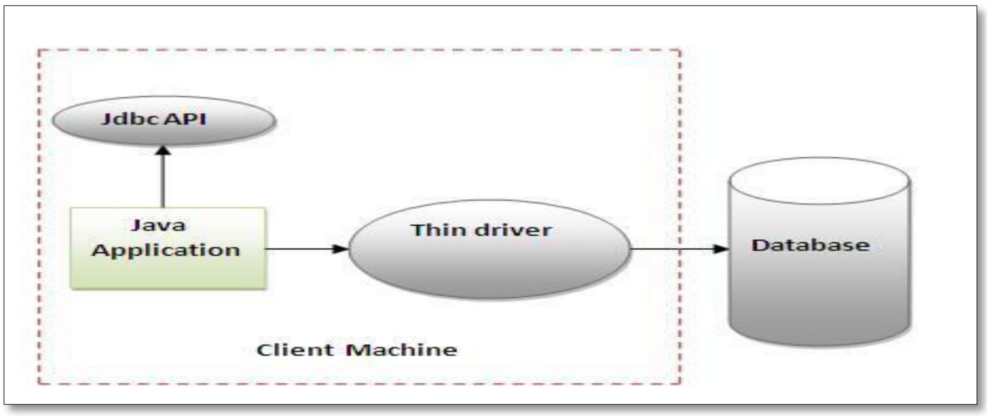


Figure 4.4: JDBC Thin Driver

Advantage:

1) Better performance than all other drivers.
2) No software is required at client side or server side.

Disadvantage:

1) Drivers depend on the Database.

4.3 Database

4.3.1 MySQL

MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. This tutorial will give you a quick start to MySQL and make you comfortable with MySQL programming.

MySQL is a fast, easy to use relational database. It is currently the most popular open-source database. It is very commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications.

MySQL is used for many small and big businesses. It is developed, marketed and supported by MySQL AB, a Swedish company. It is written in C and C++.

4.3.1.1 MySQL Features

1) **Relational Database Management System (RDBMS):** MySQL is a relational database management system.
2) **Easy to use:** MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
3) **It is secure:** MySQL consist of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
4) **Client/ Server Architecture:** MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
5) **Free to download:** MySQL is free to use and you can download it from MySQL official website.
6) **It is scalable:** MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
7) **Compatible on many operating systems:** MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).
8) **Allows roll-back:** MySQL allows transactions to be rolled back, commit and crash recovery.
9) **High Performance:** MySQL is faster, more reliable and cheaper because of its unique storage engine architecture.
10) **High Flexibility:** MySQL supports a large number of embedded applications which makes MySQL very flexible.
11) **High Productivity:** MySQL uses Triggers, Stored procedures and views which allows the developer to give a higher productivity.

4.3.1.2 Disadvantages / Drawback of MySQL

1) MySQL version less than 5.0 doesn't support ROLE, COMMIT and stored procedure.
2) MySQL does not support a very large database size as efficiently.
3) MySQL doesn't handle transactions very efficiently and it is prone to data corruption.
4) MySQL is accused that it doesn't have a good developing and debugging tool compared to paid databases.
5) MySQL doesn't support SQL check constraints.

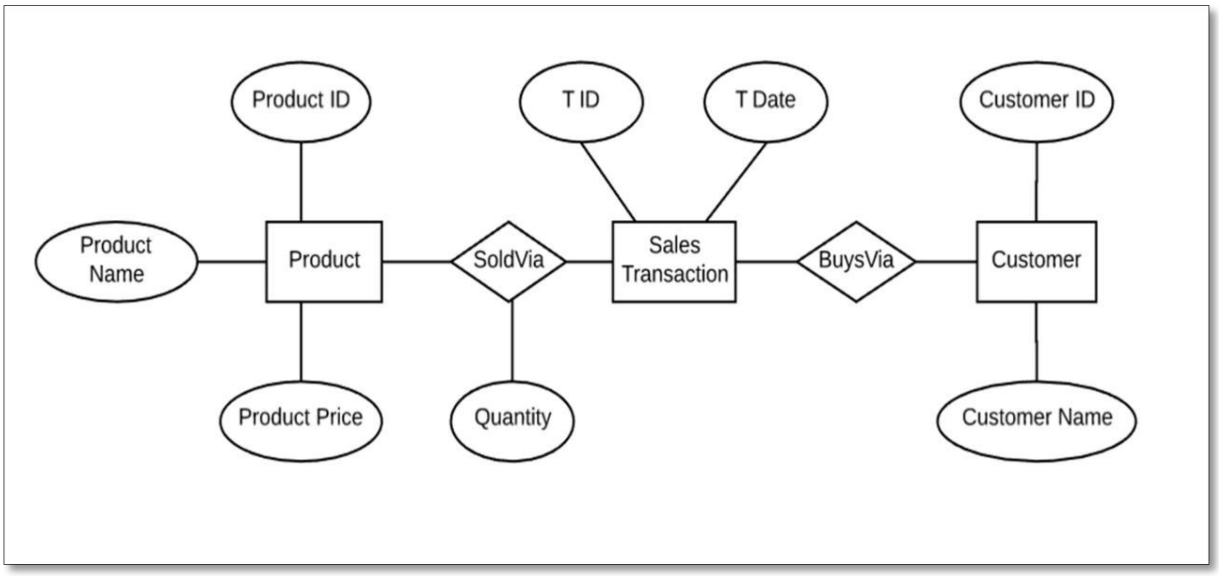


Figure 4.5: Entity Relationship Diagram

4.3.2 SQLyog

SQLyog GUI is the most powerful MySQL manager and admin tool, combining the features of MySQL Query Browser, Administrator, phpMyAdmin and various other MySQL Front Ends and MySQL clients in a single intuitive interface.

4.3.2.1 Working With SQLyog

The main window of SQLyog is divided into three panes -- Object Browser, SQL Window, Result Window.

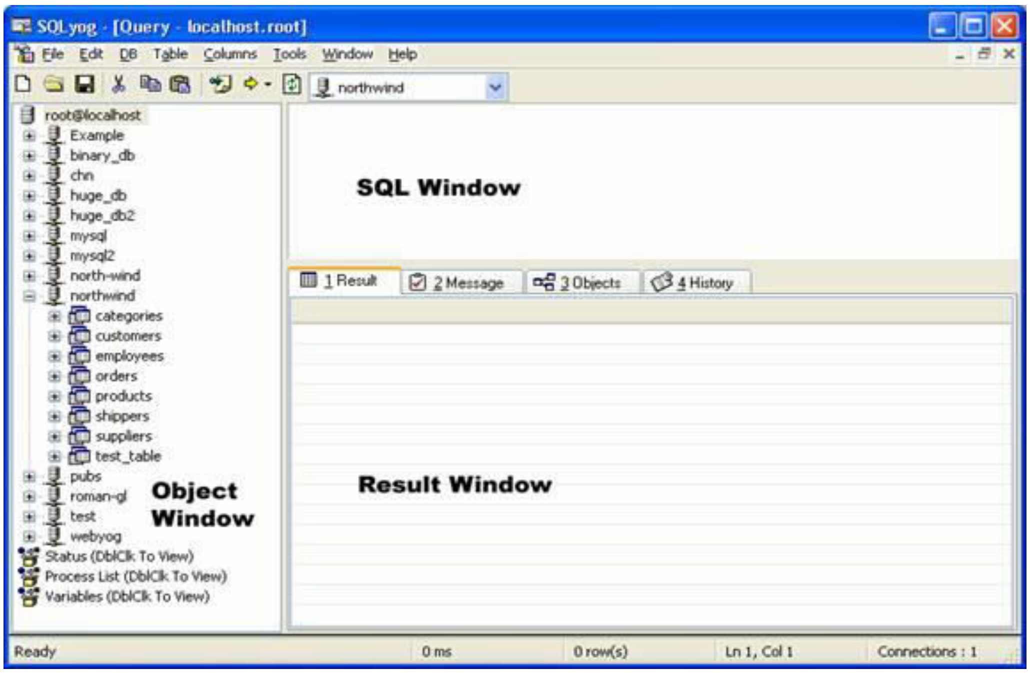


Figure 4.6: SQLyog Window

The Object Browser gives you details of all the Database/Tables/Columns/Indexes available in the MySQL server. You can Show/Hide Object Browser and Result Window by pressing Ctrl+1 and Ctrl+2 respectively.

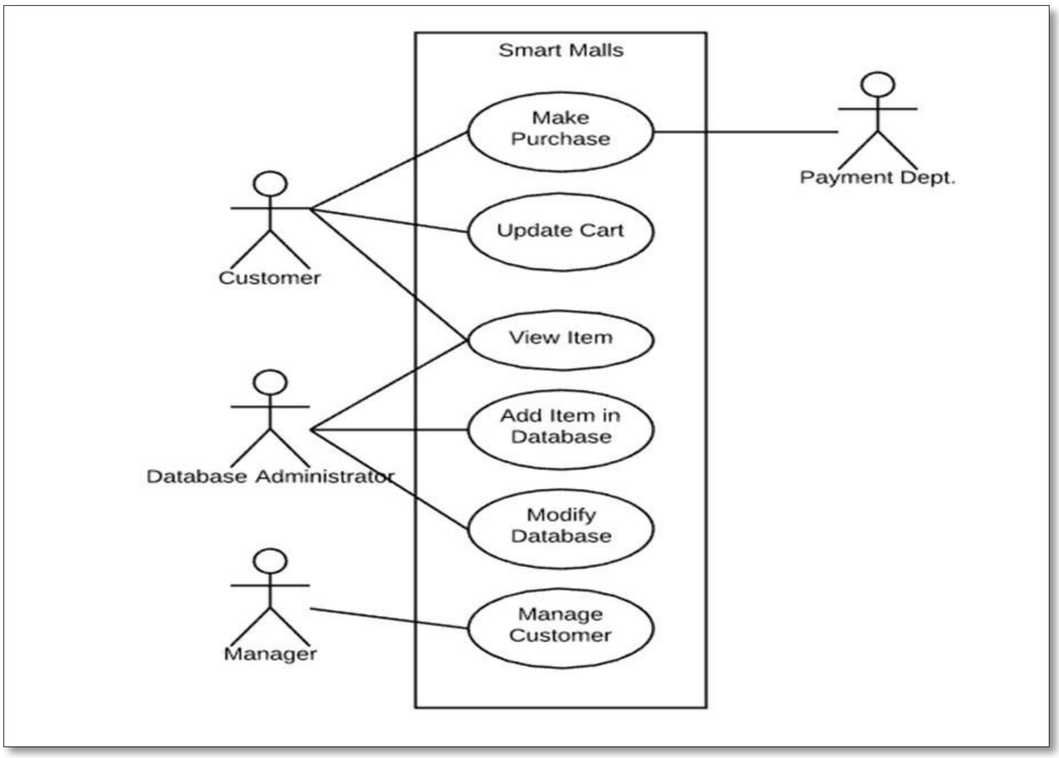


Figure 4.7: Use case Diagram

4.4 Arduino IDE

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board. Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed (referred to as a microcontroller) and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

4.4.1 Features of Arduino

1. Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
2. You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
3. Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
4. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
5. Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

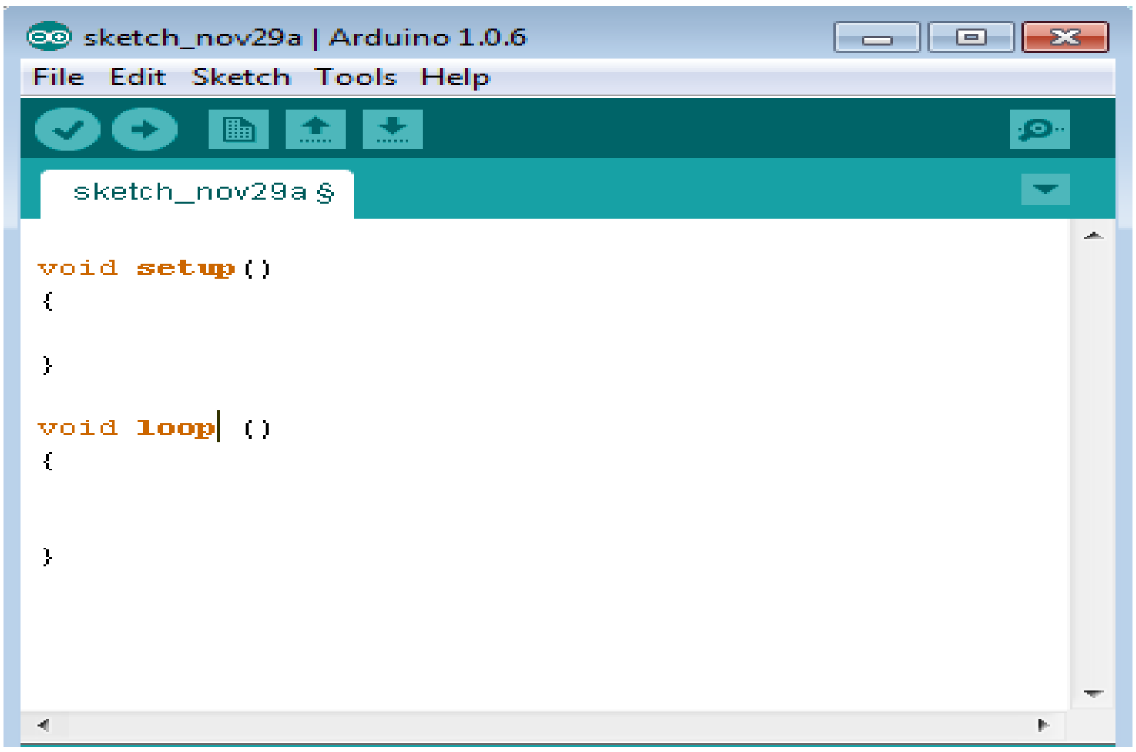
4.4.2 Arduino Program Structure

Software structure consists of two main functions −

- Setup( ) function
- Loop( ) function

```
Void setup ( ) {
}
```

PURPOSE − The **setup** () function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.



```
void setup()
{

}

void loop ()
{


}
```

Figure 4.8: Arduino IDE

```
Void Loop ( ) {
}
```

PURPOSE − After creating a **setup ()** function, which initializes and sets the initial values, the **loop ()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

4.5 Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems.

Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C uses most of the syntax and semantics of standard C, e.g., main() function, variable definition, data type declaration, conditional statements (if, switch case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, etc.

Embedded C is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependant executables. Embedded C requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications.

Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications. Embedded systems often do not have a console, which is available in case of desktop applications.

CHAPTER 5

SOFTWARE CODING

4.1 Hardware Coding

```
#include<LiquidCrystal.h>
#include<SoftwareSerial.h>
#include<EEPROM.h>
LiquidCrystal lcd(13,12,11,10,9,8);
SoftwareSerial mSerial(2,3);
unsigned int ITEMS[4]={51155,28217,44646,33496};
unsigned int QTY[4]={0,0,0,0};
char card[12];
unsigned int number;
void setup()
 {    Serial.begin(9600);
      mSerial.begin(9600);
      lcd.begin(16,2);
      lcd.setCursor(0,0);
      lcd.print("Setting Up   ");
     delay(2000);
     lcd.setCursor(0,0);
     lcd.print("Shopping Mall");
     pinMode(A0,INPUT);
    QTY[0]=EEPROM.read(0);
    QTY[1]=EEPROM.read(1);
    QTY[2]=EEPROM.read(2);
    QTY[3]=EEPROM.read(3);
}
void SHOW()
  {   int i;
     number=0;
```

```
   for(i=0;i<4;i++)
    {   if(card[6+i]>='0'&&card[6+i]<='9')
            number=(number*16)+(card[6+i]-'0');
        if(card[6+i]>='A'&&card[6+i]<='Z')
            number=(number*16)+(card[6+i]-'A'+10);
    }
lcd.clear();
lcd.setCursor(0,0);
lcd.print("ID:");
for(i=0;i<12;i++)
  lcd.write(card[i]);
lcd.setCursor(0,1);
lcd.print(number);
delay(3000);
lcd.clear();
lcd.setCursor(0,0);
if(number==ITEMS[0])
    { lcd.print("ITEM:1"); i=0; }
if(number==ITEMS[1])
    { lcd.print("ITEM:2"); i=1; }
if(number==ITEMS[2])
    { lcd.print("ITEM:3"); i=2; }
if(number==ITEMS[3])
    { lcd.print("ITEM:4"); i=3; }
QTY[i]++;
lcd.setCursor(0,1);
lcd.print("Quantity:");

lcd.print(QTY[i]);
EEPROM.write(0,QTY[0]);
EEPROM.write(1,QTY[1]);
EEPROM.write(2,QTY[2]);
EEPROM.write(3,QTY[3]);
delay(2000);
```

```
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Shopping Mall");
}
int i;
void loop()
{   if(mSerial.available()>0)
    {    delay(700);
          i=0;
         while(mSerial.available()>0&&i<12)
          {   card[i]=(char)mSerial.read();
              i++;
          }
    SHOW();
     }
  if(digitalRead(A0)==LOW)
    {  lcd.clear();
       lcd.setCursor(0,0);
       lcd.print("Sending Data");
       delay(2000);
       for(i=0;i<4;i++)
         {   Serial.print(i+1);
             Serial.print(",");
             Serial.print(QTY[i]/10);


              Serial.print(QTY[i]%10);
              if(i!=3)
             Serial.print("#");
             QTY[i]=0;
             EEPROM.write(i,0);
         }
      lcd.setCursor(0,0);
      lcd.print("Thank you for");
      lcd.setCursor(0,1);
      lcd.print("  Shopping   ");
      delay(3000);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Shopping Mall");
    }
}
```

4.2 Software Coding

4.2.1 Index.java

```
import javax.swing.JOptionPane;
public class index extends javax.swing.JFrame
{   /* Creates new form index  */
        public index() {
        initComponents();
        try
        {       connn obj=new connn();
                obj.connn1();
                 int i = 0;
                obj.st.executeUpdate("delete from  customer");
        }
        catch(Exception et)
        {
                et.printStackTrace();
        }
        ReadingDatainsert obj=new ReadingDatainsert();
        obj.reddt();
        JOptionPane.showMessageDialog(null,"Process Completed");
  }
        /* * 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() {
          jPanel1 = new javax.swing.JPanel();
           jLabel1 = new javax.swing.JLabel();
          jButton1 = new javax.swing.JButton();
     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
      setTitle("Shopping");
      jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,
153), 2, true));
      jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/1.jpg"))); // NOI18N
      jButton1.setBackground(new java.awt.Color(255, 255, 255));
     jButton1.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
     jButton1.setText("SHOPPING SYSTEM");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
             jButton1ActionPerformed(evt);
          }
       });
       javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
       jPanel1.setLayout(jPanel1Layout);
       jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jLabel1, javax.swing.GroupLayout.Alignment.TRAILING)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,jPanel1Layout.createSe
quentialGroup().addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 374,
javax.swing.GroupLayout.PREFERRED_SIZE)
             .addContainerGap())
       );
```

```
        jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
 .addGroup(jPanel1Layout.createSequentialGroup().addComponent(jLabel1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
55,javax.swing.GroupLayout.PREFERRED_SIZE).addContainerGap(23,
Short.MAX_VALUE))
        );
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        pack();
    }// </editor-fold>//GEN-END:initComponents
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed
        this.setVisible(false);
        new viewrecord().setVisible(true);        // TODO add your handling code here:
    }//GEN-LAST:event_jButton1ActionPerformed
    /* @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.        */
        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(index.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(index.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(index.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(index.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 index().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration//GEN-END:variables
}
```

4.2.2 index.form

```
<?xml version="1.0" encoding="UTF-8" ?>
<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
    <Property name="defaultCloseOperation" type="int" value="3"/>
    <Property name="title" type="java.lang.String" value="Shopping"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle"
type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
```

```
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
          <Component id="jPanel1" min="-2" max="-2" attributes="0"/>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="0" attributes="0">
          <Component id="jPanel1" alignment="0" max="32767" attributes="0"/>
      </Group>
    </DimensionLayout>
  </Layout>
  <SubComponents>
    <Container class="javax.swing.JPanel" name="jPanel1">
      <Properties>
        <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
          <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
            <LineBorder roundedCorners="true" thickness="2">
              <Color PropertyName="color" blue="99" green="0" red="0" type="rgb"/>
            </LineBorder>
          </Border>
        </Property>
      </Properties>
      <Layout>
        <DimensionLayout dim="0">
          <Group type="103" groupAlignment="0" attributes="0">
              <Component id="jLabel1" alignment="1" min="-2" max="-2"
attributes="0"/>
              <Group type="102" alignment="1" attributes="0">
```

```
                <Component id="jButton1" min="-2" pref="374" max="-2"
attributes="0"/>
                <EmptySpace max="-2" attributes="0"/>
            </Group>
          </Group>
        </DimensionLayout>
        <DimensionLayout dim="1">
          <Group type="103" groupAlignment="0" attributes="0">
            <Group type="102" alignment="0" attributes="0">
                <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
                <EmptySpace max="-2" attributes="0"/>
                <Component id="jButton1" min="-2" pref="55" max="-2"
attributes="0"/>
                <EmptySpace pref="23" max="32767" attributes="0"/>
            </Group>
          </Group>
        </DimensionLayout>
      </Layout>
      <SubComponents>
        <Component class="javax.swing.JLabel" name="jLabel1">
          <Properties>
            <Property name="icon" type="javax.swing.Icon"
editor="org.netbeans.modules.form.editors2.IconEditor">
              <Image iconType="3" name="/img/1.jpg"/>
            </Property>
          </Properties>
        </Component>
        <Component class="javax.swing.JButton" name="jButton1">
          <Properties>
            <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
              <Color blue="ff" green="ff" red="ff" type="rgb"/>
            </Property>
            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
              <Font name="Tahoma" size="24" style="1"/>
            </Property>
            <Property name="text" type="java.lang.String" value="SHOPPING
SYSTEM"/>
          </Properties>
          <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton1ActionPerformed"/>
          </Events>
        </Component>
      </SubComponents>
    </Container>
  </SubComponents>
</Form>
```

4.2.3 bookingproduct.java

```
import javax.swing.JOptionPane;
public class Bookingproduct extends javax.swing.JFrame {
    /*  Creates new form Bookingdomflight */
    public Bookingproduct()
    {
        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">//GEN-
BEGIN:initComponents
```

```
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jButton2 = new javax.swing.JButton();
        jButton1 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
jPanel1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 102, 0),
2, true));
jPanel1.setDebugGraphicsOptions(javax.swing.DebugGraphics.NONE_OPTION);
jLabel2.setText("Name");
jLabel3.setText("Mobile");
jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
jLabel1.setText("Payment Gateway");
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
 jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addComponent(jLabel2)
                .addGap(64, 64, 64)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 49,
Short.MAX_VALUE) .addComponent(jLabel3).addGap(18, 18, 18)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 213,
javax.swing.GroupLayout.PREFERRED_SIZE).addContainerGap())
            .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(207, 207, 207)
            .addComponent(jLabel1)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
      );
      jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addGap(12, 12,
12).addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.BASELINE).addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jLabel3)
               .addComponent(jLabel2))
            .addContainerGap(30, Short.MAX_VALUE))
      );
      jPanel2.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,
0), 1, true));
 jButton2.setFont(new java.awt.Font("Times New Roman", 1, 36)); // NOI18N
jButton2.setText("CARD");
 jButton2.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
              jButton2ActionPerformed(evt);
         }
      });
jButton1.setFont(new java.awt.Font("Times New Roman", 1, 36)); // NOI18N
      jButton1.setText("CASH");
      jButton1.addActionListener(new java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt) {
             jButton1ActionPerformed(evt);
          }
       });
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
       jPanel2.setLayout(jPanel2Layout);
       jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup().addGap(42, 42, 42)
             .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
163, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
             .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
196, javax.swing.GroupLayout.PREFERRED_SIZE)
             .addGap(50, 50, 50)));
       jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup().addContainerGap(66, Short.MAX_VALUE)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE).addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 91,
javax.swing.GroupLayout.PREFERRED_SIZE)).addContainerGap())   );
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
       getContentPane().setLayout(layout);
       layout.setHorizontalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
          .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
          layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE).addPreferredGap(javax.swing.Layout
Style.ComponentPlacement.RELATED).addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE).addContainerGap(javax.swing.Group
Layout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed
       if(jTextField1.getText().equals("") || jTextField2.getText().equals("") ||
jTextField2.getText().length()!=10)
        {
          JOptionPane.showMessageDialog(null,"Filed Should not Blank or Mobt
Valid");
        }
        else
        {
        this.setVisible(false);
        new paymentgateway().setVisible(true);
        }
    }//GEN-LAST:event_jButton2ActionPerformed
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed
```

```
      if(jTextField1.getText().equals("") || jTextField2.getText().equals("") ||
jTextField2.getText().length()!=10)
        {
            JOptionPane.showMessageDialog(null,"Filed Should not Blank or Mobt
Valid");
         }
        else
        {
            this.setVisible(false);
           new viewrecord().setVisible(true);
           JOptionPane.showMessageDialog(null,"Thanks");
         }
    }//GEN-LAST:event_jButton1ActionPerformed
    /* @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.
         */
        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(Bookingproduct.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(Bookingproduct.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
```

```
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(Bookingproduct.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(Bookingproduct.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Bookingproduct().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    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.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration//GEN-END:variables
}
```

4.2.4 bookingproduct.form

```
<?xml version="1.0" encoding="UTF-8" ?>
<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
```

```
    <Property name="defaultCloseOperation" type="int" value="2"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle"
type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>
  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
          <Component id="jPanel1" max="32767" attributes="0"/>
          <Component id="jPanel2" max="32767" attributes="0"/>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="0" attributes="0">
              <Component id="jPanel1" min="-2" max="-2" attributes="0"/>
              <EmptySpace max="-2" attributes="0"/>
              <Component id="jPanel2" min="-2" max="-2" attributes="0"/>
              <EmptySpace max="32767" attributes="0"/>
          </Group>
      </Group>
    </DimensionLayout>
  </Layout>
  <SubComponents>
    <Container class="javax.swing.JPanel" name="jPanel1">
      <Properties>
        <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
          <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
            <LineBorder roundedCorners="true" thickness="2">
              <Color PropertyName="color" blue="0" green="66" red="0" type="rgb"/>
            </LineBorder>
          </Border>
        </Property>
        <Property name="debugGraphicsOptions" type="int" value="-1"/>
      </Properties>
      <Layout>
        <DimensionLayout dim="0">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" alignment="0" attributes="0">
                  <EmptySpace min="-2" pref="20" max="-2" attributes="0"/>
                  <Component id="jLabel2" min="-2" max="-2" attributes="0"/>
                  <EmptySpace min="-2" pref="64" max="-2" attributes="0"/>
                  <Component id="jTextField1" min="-2" pref="213" max="-2"
attributes="0"/>
                  <EmptySpace pref="49" max="32767" attributes="0"/>
                  <Component id="jLabel3" min="-2" max="-2" attributes="0"/>
```

```
         <EmptySpace type="separate" max="-2" attributes="0"/>
         <Component id="jTextField2" min="-2" pref="213" max="-2"
attributes="0"/>
         <EmptySpace max="-2" attributes="0"/>
       </Group>
       <Group type="102" alignment="0" attributes="0">
         <EmptySpace min="-2" pref="207" max="-2" attributes="0"/>
         <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
         <EmptySpace max="32767" attributes="0"/>
       </Group>
     </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
     <Group type="103" groupAlignment="0" attributes="0">
       <Group type="102" alignment="0" attributes="0">
         <EmptySpace max="-2" attributes="0"/>
         <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
         <EmptySpace min="-2" pref="12" max="-2" attributes="0"/>
         <Group type="103" groupAlignment="3" attributes="0">
           <Component id="jTextField1" alignment="3" min="-2" max="-2"
attributes="0"/>
           <Component id="jTextField2" alignment="3" min="-2" max="-2"
attributes="0"/>
           <Component id="jLabel3" alignment="3" min="-2" max="-2"
attributes="0"/>
           <Component id="jLabel2" alignment="3" min="-2" max="-2"
attributes="0"/>
         </Group>
         <EmptySpace pref="30" max="32767" attributes="0"/>
       </Group>
     </Group>
    </DimensionLayout>
   </Layout>
   <SubComponents>
```

```
        <Component class="javax.swing.JLabel" name="jLabel2">
          <Properties>
            <Property name="text" type="java.lang.String" value="Name"/>
          </Properties>
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel3">
          <Properties>
            <Property name="text" type="java.lang.String" value="Mobile"/>
          </Properties>
        </Component>
        <Component class="javax.swing.JTextField" name="jTextField1">
        </Component>
        <Component class="javax.swing.JTextField" name="jTextField2">
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel1">
          <Properties>
            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
              <Font name="Times New Roman" size="24" style="1"/>
            </Property>
            <Property name="text" type="java.lang.String" value="Payment Gateway"/>
          </Properties>
        </Component>
      </SubComponents>
    </Container>
    <Container class="javax.swing.JPanel" name="jPanel2">
      <Properties>
        <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
          <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
            <LineBorder roundedCorners="true"/>
          </Border>
        </Property>
      </Properties>
      <Layout>
        <DimensionLayout dim="0">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" alignment="1" attributes="0">
                  <EmptySpace min="-2" pref="42" max="-2" attributes="0"/>
                  <Component id="jButton1" min="-2" pref="163" max="-2"
attributes="0"/>
                  <EmptySpace max="32767" attributes="0"/>
                  <Component id="jButton2" min="-2" pref="196" max="-2"
attributes="0"/>
                  <EmptySpace min="-2" pref="50" max="-2" attributes="0"/>
              </Group>
          </Group>
        </DimensionLayout>
        <DimensionLayout dim="1">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" alignment="1" attributes="0">
                  <EmptySpace pref="66" max="32767" attributes="0"/>
                  <Group type="103" groupAlignment="3" attributes="0">
                      <Component id="jButton1" alignment="3" min="-2" pref="91" max="-
2" attributes="0"/>
                      <Component id="jButton2" alignment="3" min="-2" pref="91" max="-
2" attributes="0"/>
                  </Group>
                  <EmptySpace max="-2" attributes="0"/>
              </Group>
          </Group>
        </DimensionLayout>
      </Layout>
      <SubComponents>
        <Component class="javax.swing.JButton" name="jButton2">
          <Properties>
            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
```

```
            <Font name="Times New Roman" size="36" style="1"/>
          </Property>
          <Property name="text" type="java.lang.String" value="CARD"/>
        </Properties>
        <Events>
          <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton2ActionPerformed"/>
        </Events>
      </Component>
      <Component class="javax.swing.JButton" name="jButton1">
        <Properties>
          <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
            <Font name="Times New Roman" size="36" style="1"/>
          </Property>
          <Property name="text" type="java.lang.String" value="CASH"/>
        </Properties>
        <Events>
          <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton1ActionPerformed"/>
        </Events>
      </Component>
    </SubComponents>
  </Container>
 </SubComponents>
</Form>
```

4.2.5 paymentgateway.java

```
import javax.swing.JOptionPane;

/* To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

public class paymentgateway extends javax.swing.JFrame {

   /* Creates new form Bookingdomflight     */
   public paymentgateway() {
      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">//GEN-
BEGIN:initComponents
   private void initComponents() {

      jPanel2 = new javax.swing.JPanel();
      jLabel14 = new javax.swing.JLabel();
      jLabel15 = new javax.swing.JLabel();
      jLabel16 = new javax.swing.JLabel();
      jLabel17 = new javax.swing.JLabel();
      jComboBox1 = new javax.swing.JComboBox();
      jTextField12 = new javax.swing.JTextField();
      jTextField13 = new javax.swing.JTextField();
      jTextField14 = new javax.swing.JTextField();
      jButton2 = new javax.swing.JButton();
      jLabel18 = new javax.swing.JLabel();
      jTextField15 = new javax.swing.JTextField();
      jLabel1 = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();


setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

      jPanel2.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 0,
0), 1, true));

      jLabel14.setText("Card Type");

      jLabel15.setText("Card Number");

      jLabel16.setText("Security Code");

      jLabel17.setText("Expiry Date");
      jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "Master Card", "Debit Card", "Credit Card", "Fleet card", "ATM card", "Debit
card", "Charge card" }));

      jButton2.setText("Payment");
      jButton2.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
         }
      });
      jLabel18.setText("Card Holder Name");
      jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 24)); // NOI18N
      jLabel1.setText("Payment gateway");
      javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
      jPanel2.setLayout(jPanel2Layout);
      jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(221, 221, 221)
            .addComponent(jLabel1))
         .addGroup(jPanel2Layout.createSequentialGroup()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING).addGroup(jPanel2Layout.createSequentialGroup()
.addGap(18, 18, 18)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING, false).addComponent(jLabel16,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                     .addComponent(jLabel14,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
               .addGroup(jPanel2Layout.createSequentialGroup()
                  .addContainerGap()
                  .addComponent(jLabel18)))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)
               .addComponent(jTextField15)
               .addComponent(jComboBox1, 0, 131, Short.MAX_VALUE)
               .addComponent(jTextField13))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 15,
Short.MAX_VALUE)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)
               .addComponent(jLabel15, javax.swing.GroupLayout.DEFAULT_SIZE,
87, Short.MAX_VALUE)
```

```
                    .addComponent(jLabel17, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addGap(18, 18, 18)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING, false)
                    .addComponent(jTextField12)
                    .addComponent(jTextField14)
                    .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
196, Short.MAX_VALUE)))
        );
        jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(6, 6, 6)
                .addComponent(jLabel1)
                .addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(jLabel14)
                    .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField12,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel15))
                .addGap(18, 18, 18)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(jLabel16)
                    .addComponent(jTextField13,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField14,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel17))
                .addGap(18, 18, 18)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
                    .addComponent(jTextField15,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel18)
                .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

        jLabel2.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/img/py.png"))); // NOI18N
        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()
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
442, Short.MAX_VALUE)
  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(14, 14, 14))        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
                .addComponent(jLabel2)
                .addGap(0, 11, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(110, 110, 110))        );
        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed
if(jTextField12.getText().equals("") || jTextField13.getText().equals("") ||
jTextField14.getText().equals("") || jTextField15.getText().equals(""))
        {
       JOptionPane.showMessageDialog(null,"Filed Should not Blank");
        }
        else        {

        JOptionPane.showMessageDialog(null,"Thanks");
         this.setVisible(false);
```

```
      new viewrecord().setVisible(true);
      }
    }//GEN-LAST:event_jButton2ActionPerformed

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
             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(paymentgateway.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(paymentgateway.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(paymentgateway.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(paymentgateway.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new paymentgateway().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton jButton2;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel15;
    private javax.swing.JLabel jLabel16;
    private javax.swing.JLabel jLabel17;
    private javax.swing.JLabel jLabel18;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField jTextField12;
    private javax.swing.JTextField jTextField13;
    private javax.swing.JTextField jTextField14;
    private javax.swing.JTextField jTextField15;
    // End of variables declaration//GEN-END:variables
}
```

4.2.6 paymentgateway.form

```
<?xml version="1.0" encoding="UTF-8" ?>
<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <Properties>
    <Property name="defaultCloseOperation" type="int" value="2"/>
  </Properties>
  <SyntheticProperties>
    <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
    <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
  </SyntheticProperties>
  <AuxValues>
    <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_autoSetComponentName"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
    <AuxValue name="FormSettings_generateMnemonicsCode"
type="java.lang.Boolean" value="false"/>
    <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle"
type="java.lang.Integer" value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
```

```
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>
  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="1" attributes="0">
              <Component id="jLabel2" pref="442" max="32767" attributes="0"/>
              <EmptySpace max="-2" attributes="0"/>
              <Component id="jPanel2" min="-2" max="-2" attributes="0"/>
              <EmptySpace min="-2" pref="14" max="-2" attributes="0"/>
          </Group>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="1" attributes="0">
          <Group type="102" alignment="0" attributes="0">
              <Component id="jLabel2" min="-2" max="-2" attributes="0"/>
              <EmptySpace min="0" pref="11" max="32767" attributes="0"/>
          </Group>
          <Group type="102" alignment="1" attributes="0">
              <EmptySpace max="32767" attributes="0"/>
              <Component id="jPanel2" min="-2" max="-2" attributes="0"/>
              <EmptySpace min="-2" pref="110" max="-2" attributes="0"/>
          </Group>
      </Group>
    </DimensionLayout>
  </Layout>
  <SubComponents>
    <Container class="javax.swing.JPanel" name="jPanel2">
      <Properties>
        <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
          <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
```

```
            <LineBorder roundedCorners="true"/>
          </Border>
        </Property>
      </Properties>
      <Layout>
        <DimensionLayout dim="0">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" attributes="0">
                  <EmptySpace min="-2" pref="221" max="-2" attributes="0"/>
                  <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
              </Group>
              <Group type="102" alignment="0" attributes="0">
                  <Group type="103" groupAlignment="0" attributes="0">
                      <Group type="102" attributes="0">
                          <EmptySpace min="-2" pref="18" max="-2" attributes="0"/>
                          <Group type="103" groupAlignment="1" max="-2" attributes="0">
                              <Component id="jLabel16" alignment="0" max="32767"
attributes="0"/>
                              <Component id="jLabel14" alignment="0" max="32767"
attributes="0"/>
                          </Group>
                      </Group>
                      <Group type="102" attributes="0">
                          <EmptySpace max="-2" attributes="0"/>
                          <Component id="jLabel18" min="-2" max="-2" attributes="0"/>
                      </Group>
                  </Group>
                  <EmptySpace max="-2" attributes="0"/>
                  <Group type="103" groupAlignment="0" max="-2" attributes="0">
                      <Component id="jTextField15" max="32767" attributes="0"/>
                      <Component id="jComboBox1" pref="131" max="32767"
attributes="0"/>
                      <Component id="jTextField13" max="32767" attributes="0"/>
                  </Group>
```

```
                <EmptySpace pref="15" max="32767" attributes="0"/>
                <Group type="103" groupAlignment="0" max="-2" attributes="0">
                    <Component id="jLabel15" pref="87" max="32767" attributes="0"/>
                    <Component id="jLabel17" max="32767" attributes="0"/>
                </Group>
                <EmptySpace type="separate" max="-2" attributes="0"/>
                <Group type="103" groupAlignment="1" max="-2" attributes="0">
                    <Component id="jTextField12" max="32767" attributes="0"/>
                    <Component id="jTextField14" max="32767" attributes="0"/>
                    <Component id="jButton2" pref="196" max="32767" attributes="0"/>
                </Group>
            </Group>
        </Group>
      </DimensionLayout>
      <DimensionLayout dim="1">
        <Group type="103" groupAlignment="0" attributes="0">
            <Group type="102" alignment="0" attributes="0">
                <EmptySpace min="-2" pref="6" max="-2" attributes="0"/>
                <Component id="jLabel1" min="-2" max="-2" attributes="0"/>
                <EmptySpace type="separate" max="-2" attributes="0"/>
                <Group type="103" groupAlignment="3" attributes="0">
                    <Component id="jLabel14" alignment="3" min="-2" max="-2"
attributes="0"/>
                    <Component id="jComboBox1" alignment="3" min="-2" max="-2"
attributes="0"/>
                    <Component id="jTextField12" alignment="3" min="-2" max="-2"
attributes="0"/>
                    <Component id="jLabel15" alignment="3" min="-2" max="-2"
attributes="0"/>
                </Group>
                <EmptySpace type="separate" max="-2" attributes="0"/>
                <Group type="103" groupAlignment="3" attributes="0">
                    <Component id="jLabel16" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
                <Component id="jTextField13" alignment="3" min="-2" max="-2"
attributes="0"/>
                <Component id="jTextField14" alignment="3" min="-2" max="-2"
attributes="0"/>
                <Component id="jLabel17" alignment="3" min="-2" max="-2"
attributes="0"/>
              </Group>
              <EmptySpace type="separate" max="-2" attributes="0"/>
              <Group type="103" groupAlignment="3" attributes="0">
                <Component id="jTextField15" alignment="3" min="-2" max="-2"
attributes="0"/>
                <Component id="jLabel18" alignment="3" min="-2" max="-2"
attributes="0"/>
                <Component id="jButton2" alignment="3" min="-2" pref="41" max="-
2" attributes="0"/>
              </Group>
              <EmptySpace max="32767" attributes="0"/>
            </Group>
          </Group>
        </DimensionLayout>
      </Layout>
      <SubComponents>
        <Component class="javax.swing.JLabel" name="jLabel14">
          <Properties>
            <Property name="text" type="java.lang.String" value="Card Type"/>
          </Properties>
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel15">
          <Properties>
            <Property name="text" type="java.lang.String" value="Card Number"/>
          </Properties>
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel16">
          <Properties>
```

```
    <Property name="text" type="java.lang.String" value="Security Code"/>
  </Properties>
</Component>
<Component class="javax.swing.JLabel" name="jLabel17">
  <Properties>
    <Property name="text" type="java.lang.String" value="Expiry Date"/>
  </Properties>
</Component>
<Component class="javax.swing.JComboBox" name="jComboBox1">
  <Properties>
    <Property name="model" type="javax.swing.ComboBoxModel"
editor="org.netbeans.modules.form.editors2.ComboBoxModelEditor">
      <StringArray count="7">
        <StringItem index="0" value="Master Card"/>
        <StringItem index="1" value="Debit Card"/>
        <StringItem index="2" value="Credit Card"/>
        <StringItem index="3" value="Fleet card"/>
        <StringItem index="4" value="ATM card"/>
        <StringItem index="5" value="Debit card"/>
        <StringItem index="6" value="Charge card"/>
      </StringArray>
    </Property>
  </Properties>
</Component>
<Component class="javax.swing.JTextField" name="jTextField12">
</Component>
<Component class="javax.swing.JTextField" name="jTextField13">
</Component>
<Component class="javax.swing.JTextField" name="jTextField14">
</Component>
<Component class="javax.swing.JButton" name="jButton2">
  <Properties>
    <Property name="text" type="java.lang.String" value="Payment"/>
  </Properties>
```

```
          <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton2ActionPerformed"/>
          </Events>
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel18">
          <Properties>
            <Property name="text" type="java.lang.String" value="Card Holder Name"/>
          </Properties>
        </Component>
        <Component class="javax.swing.JTextField" name="jTextField15">
        </Component>
        <Component class="javax.swing.JLabel" name="jLabel1">
          <Properties>
            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
              <Font name="Times New Roman" size="24" style="1"/>
            </Property>
            <Property name="text" type="java.lang.String" value="Payment gateway"/>
          </Properties>
        </Component>
      </SubComponents>
    </Container>
    <Component class="javax.swing.JLabel" name="jLabel2">
      <Properties>
        <Property name="icon" type="javax.swing.Icon"
editor="org.netbeans.modules.form.editors2.IconEditor">
          <Image iconType="3" name="/img/py.png"/>
        </Property>
      </Properties>
    </Component>
  </SubComponents>
</Form>
```

CHAPTER 6

WORKING OF PROJECT

6.1 Working of Hardware

The working of SMART MALLS starts by distribution of scanning devices to the user (customer). As the customer enters the mall each customer is provided with a device which consists of Arduino Board, Reader module and a display screen (LCD).

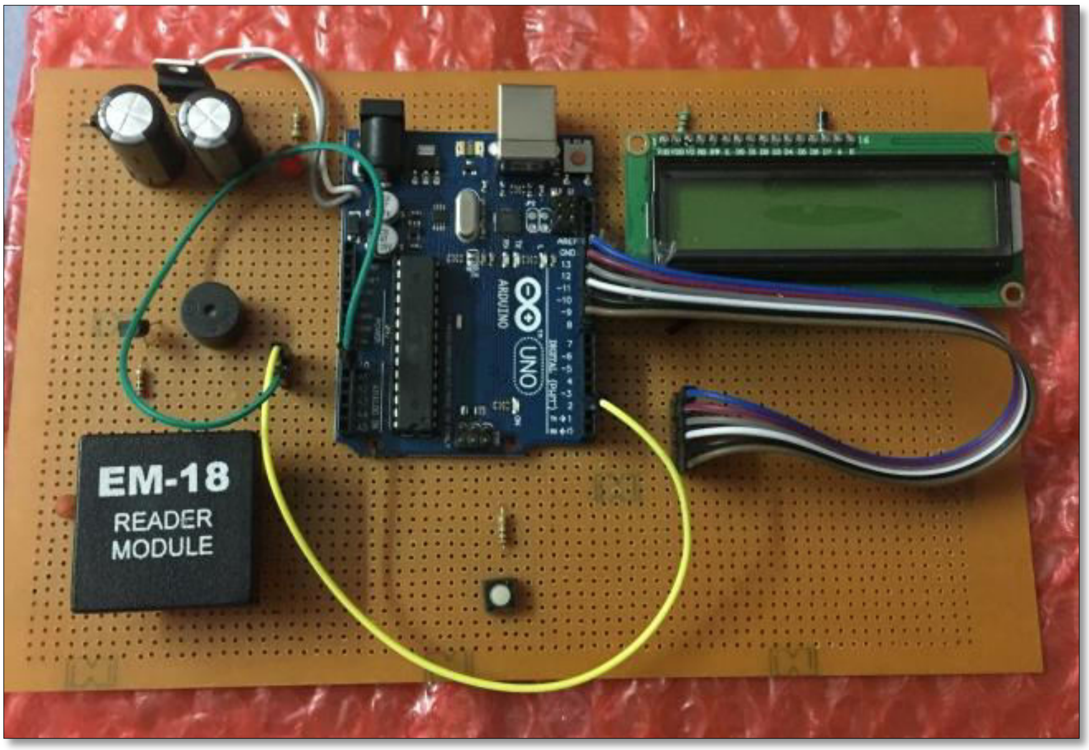


Figure 6.1: Device (Smart Malls)

Now each customer can visit to any shop for shopping, where each product has a magnetic card attached to it as a unique id. These magnetic cards are called NFC (Near Field Communication) card. Now to add any item to the cart the customer (user) has to scan the NFC card attached to that product using the device provided by the mall staff. A user (customer) should scan the NFC card only once if the quantity he/she wants to buy one, but if the quantity of a product is more than one than the user should scan the NFC card as many time as he/she wants to purchase. The quantities can be modified once the user reaches the payment gateway.

All the NFC cards have some pre-stored data such as the unique id of the product etc. Now as soon as the NFC card is scanned, the id of the card is stored in the microcontroller where the whole code of hardware is stored. Each time the NFC is scanned, a small sound comes out from the buzzer, this is only to indicate that the card is successfully scanned and the data is stored in the memory. As soon as the card is scanned there is a LCD screen attached to it which shows the unique id of the NFC card and the quantity of the product (NFC card) scanned. As each NFC card has 12 digit hexadecimal number (ID of NFC card). The code present in microcontroller scans that 12 digit number and converts this code into a unique 5 digit decimal number. So using this 5 digit the further processing is done.

```
for(i=0;i<4;i++)
            {
                  if(card[6+i]>='0'&&card[6+i]<='9')
                        number=(number*16)+(card[6+i]-'0');
                   if(card[6+i]>='A'&&card[6+i]<='Z')
                        number=(number*16)+(card[6+i]-'A'+10);
            }
```

The above code converts the 12 digit hexadecimal number to 5 digit decimal number for easier processing. This 5 digit number is matched to database to check whether this NFC card is registered or not.

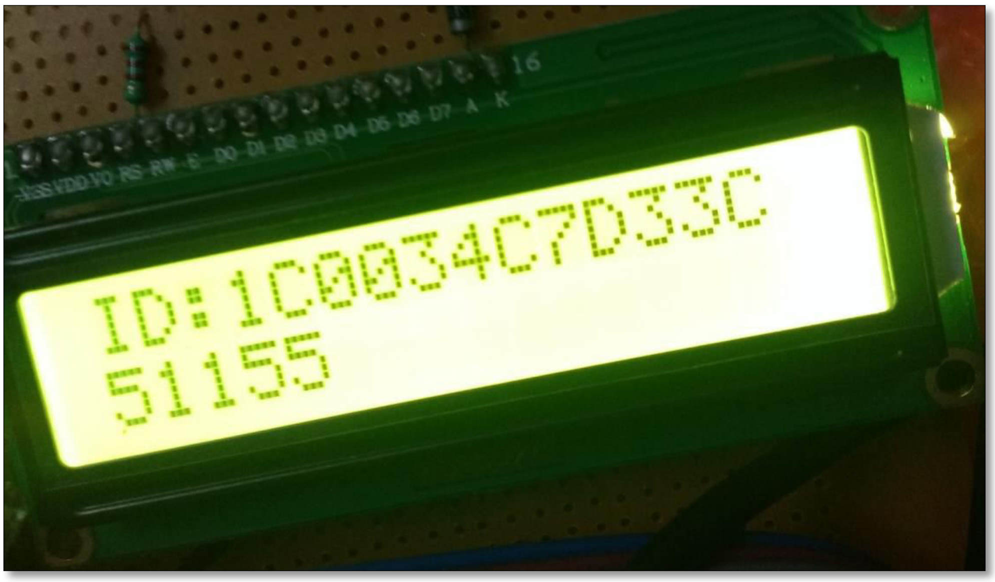


Figure 6.2: NFC card ID

Now if any unregistered NFC card is scanned, the display will show its ID but the data for that NFC card will not be sent to the software as it is not registered in the software code. Each device has a button which can be used to transfer data to the software at the payment counter. This button when pressed gives a “SENDING DATA” message on the display screen. Now as the user has completed the scanning of products and finally wants to pay amount and complete the shopping, one has to connect the device to the machine like a computer at the payment counters. And after connecting the device to the computer, a send button is to be pressed to transferring the data to the software.

6.2 Working of Software

The software code is responsible for providing the user interface to the customer, this user interface helps the customer to view their respective cart and make the further processing. Now the device connected to the computer and the sending data button is pressed. All the IDs of all scanned NFC are sent to the software. Any unregistered NFC card is rejected. The data sent from the hardware is sent in the form of 1,02#2,01#3,00#4,01# .

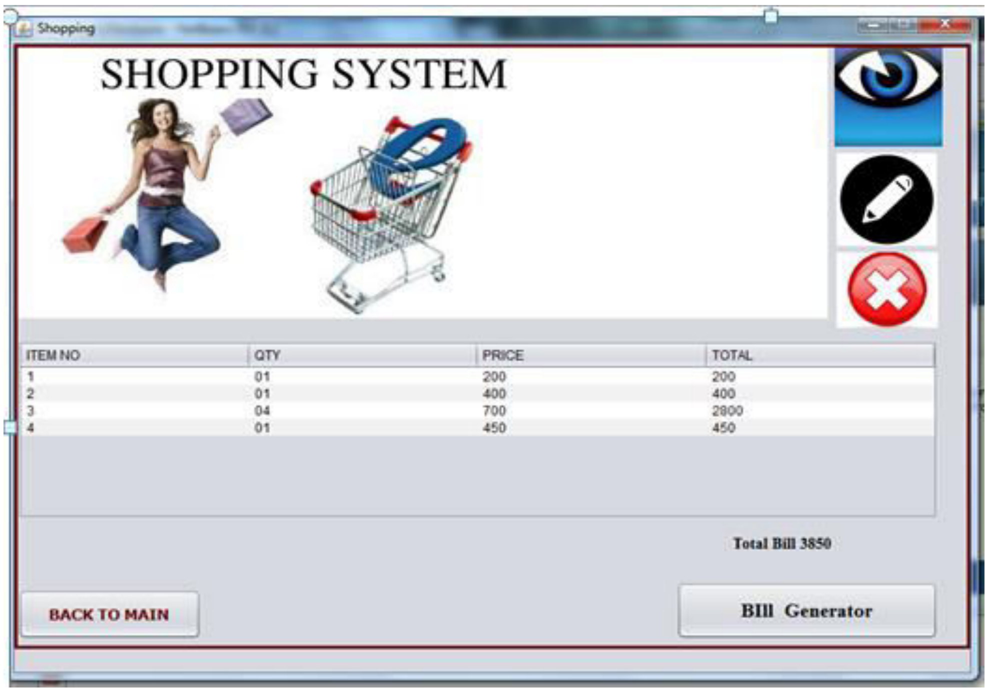
Now here each ‘#’ sign is used to display the termination of a particular NFC card. In above data there are four NFC registered such as:

NFC	Data
NFC 1	1,02#
NFC 2	2,01#
NFC 3	3,00#
NFC 4	4,01#

Table 6.1: NFC card data

Each NFC card data has two fields separated by ‘,’ (comma). The entry before comma is the number of NFC card and the entry after each comma is the quantity of that particular NFC. Each NFC’s quantity is represented by two bits ie. 01, 00, 02 etc. So for data such as “ x,m# “ the NFC card ‘x’ has ‘m’ number of quantities. Now this data after sorting is stored into data base.

Each product present in the mall is already added to the data base with their total quantity and the price for any particular product. When the Send Data button is pressed on the device the user interface is opened and the message pops up "Process Completed" which means the data is successful transferred to the software and stored in the database. After which a new window namely "view record" as in figure 6.3 opens and the whole cart is displayed. This interface has some buttons in it such as Update, Delete and Refresh.



ITEM NO	QTY	PRICE	TOTAL
1	01	200	200
2	01	400	400
3	04	700	2800
4	01	450	450

Total Bill 3850

Figure 6.3: View Record Window

Each time any change is made the user is needed to refresh the view record window. Suppose the user wants to update the cart that is increase or decrease the quantities of the product. The user is needed to provide the item number as displayed in the cart in view record window and the quantity for that particular item (product). Now if the item number is correct, the quantity for that particular item will be update in the database. Now suppose the user want to completely delete an item from the cart, the user needs to press the delete item button in the view record window. Doing so will generate an alert message as "Item Deleted Successfully". Each time whether an item is updated or deleted the total is needed to be changed. As the price of the product is specified in the database the total is calculated by fetching the price from database. So by pressing refresh button total bill is changed and displayed in view record window.

After this if the user is satisfied, the bill generation button is to be pressed. Now the control will be taken to the payment gateway portal. Here a user can opt between cash or card. If a user opts for card, necessary card details are needed to be provided and after successful payment shopping is completed and an alert message is popped up "Thank you for Shopping".

Once the payment is done the entire purchased product, be it anything from any shop in the mall it will be delivered at the payment counter after successful payment.

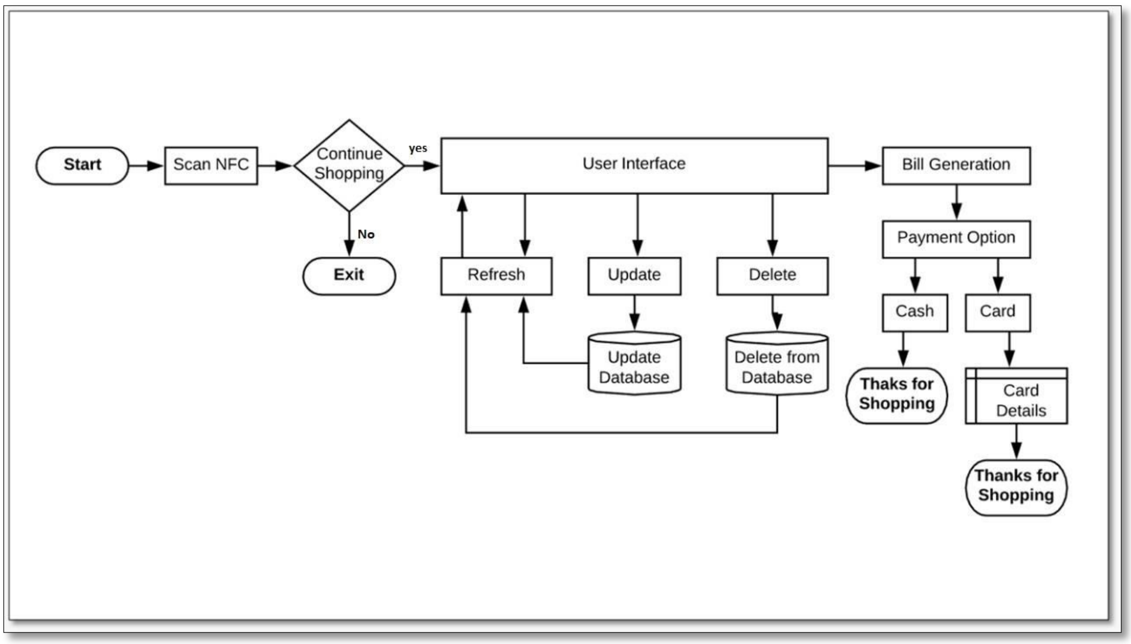


Figure 6.4: Flowchart Diagram (Working of Smart Malls)

CHAPTER 7

OUTPUT AND RESULT ANALYSIS

7.1 Output and screenshots

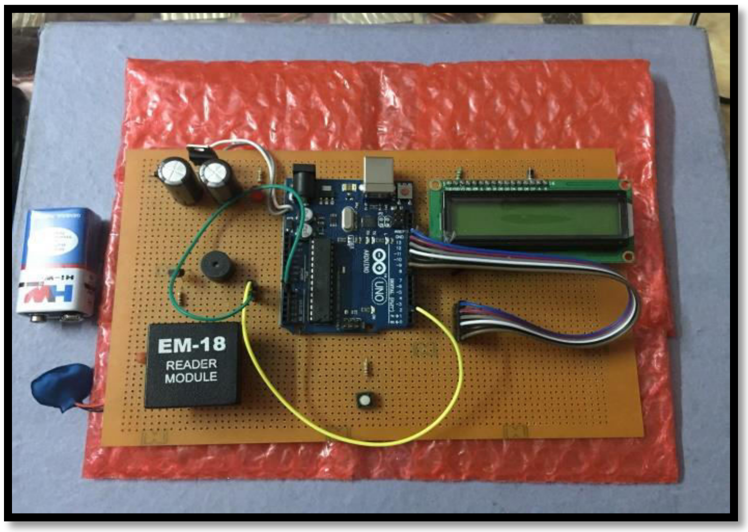


Figure 7.1: Device for the scanning of products

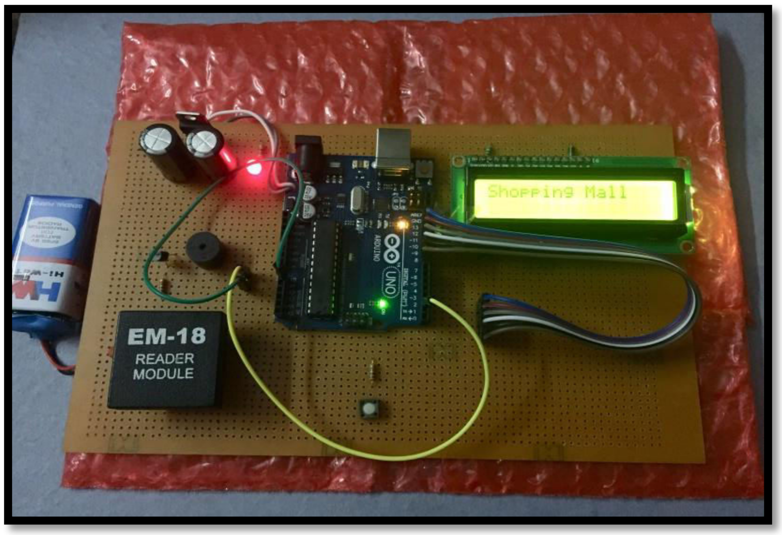
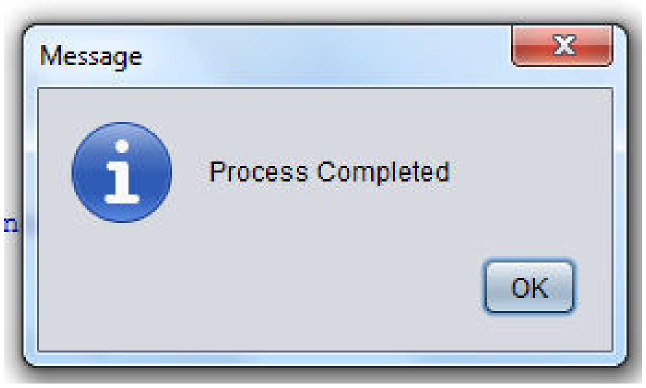


Figure 7.2: A working module of the Scanner Device

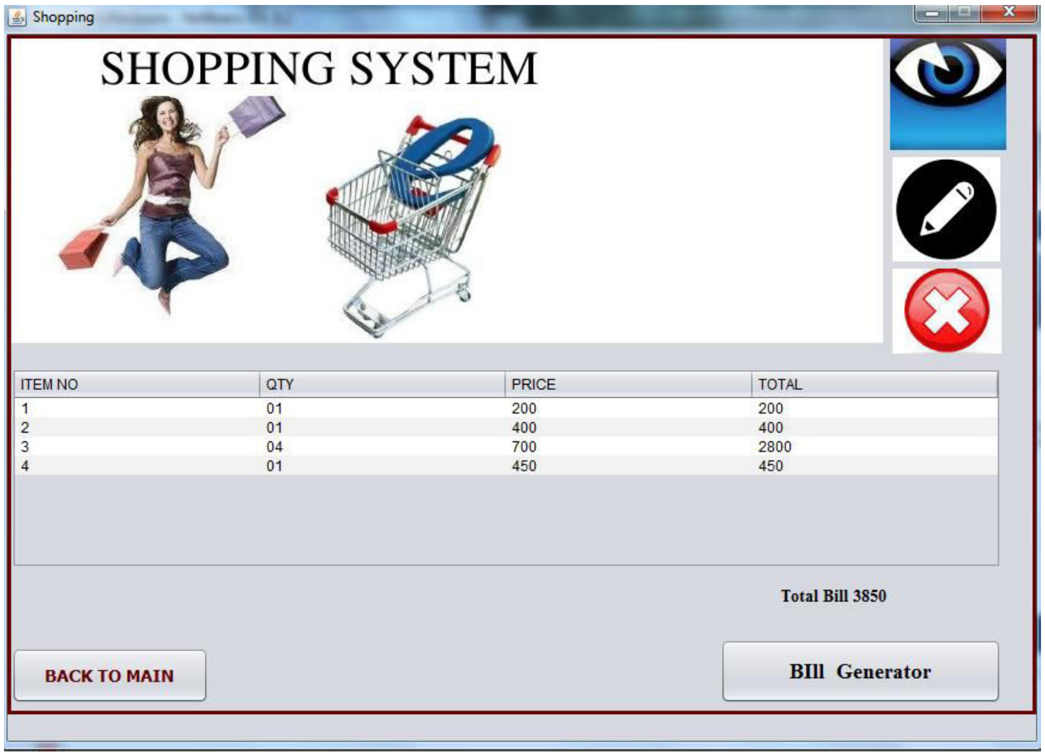
As the SEND DATA button is pressed on the device, the data is sent to the software and a message is shown (figure 7.3) as “Process Completed”.


Message

Process Completed

Figure 7.3: Message for process completion

The output declared that the data or the shopping cart items are transferred to the software for further processing.


Shopping

SHOPPING SYSTEM

ITEM NO	QTY	PRICE	TOTAL
1	01	200	200
2	01	400	400
3	04	700	2800
4	01	450	450

Total Bill 3850

Figure 7.4: View Record Window

The above figure (Figure 7.4) displays the main page of the user interface. In this page all the items in the cart are shown.

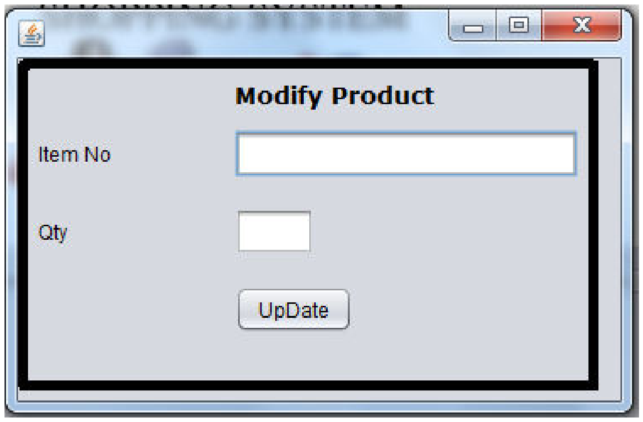


Figure 7.5: Modify the product in the cart

The above figure (Figure 7.3) displays the product modification window, i.e for any modifications to be made in the existing cart. Providing item number and its quantity modifications can be done.

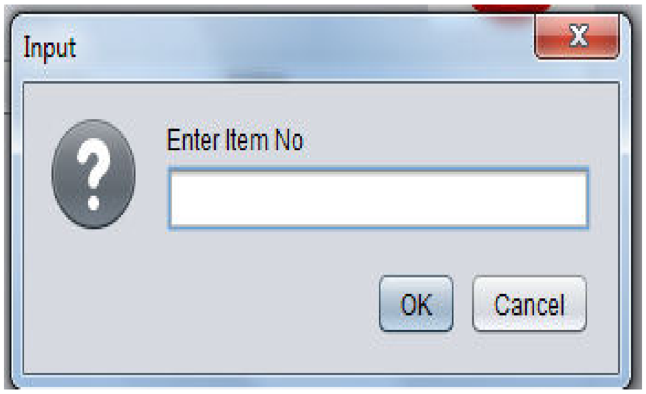


Figure 7.6: Deletion from the existing cart

This above figure (Figure 7.6) gives the interface the deletion of the item from the cart.

Providing item number a user can delete any item from cart.

By pressing the BILL GENERATION button the control moves to Payment Gateway window where user chooses between given payment modes.

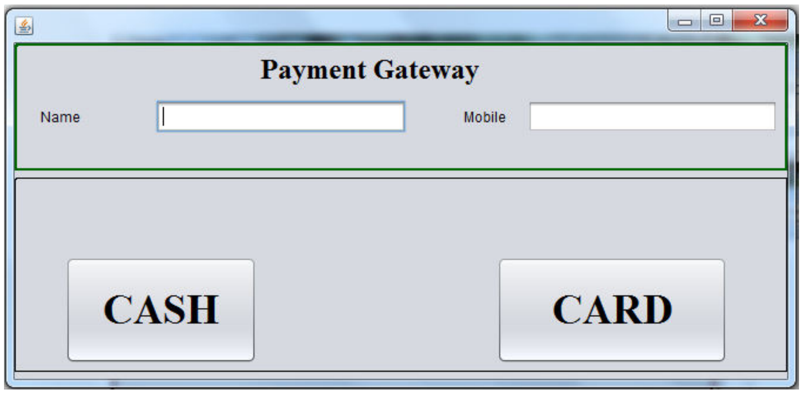


Figure 7.7: Payment Gateway

From above figure (Figure 7.7) if a user opts for card then the necessary card details are needed to be given for successful payment.

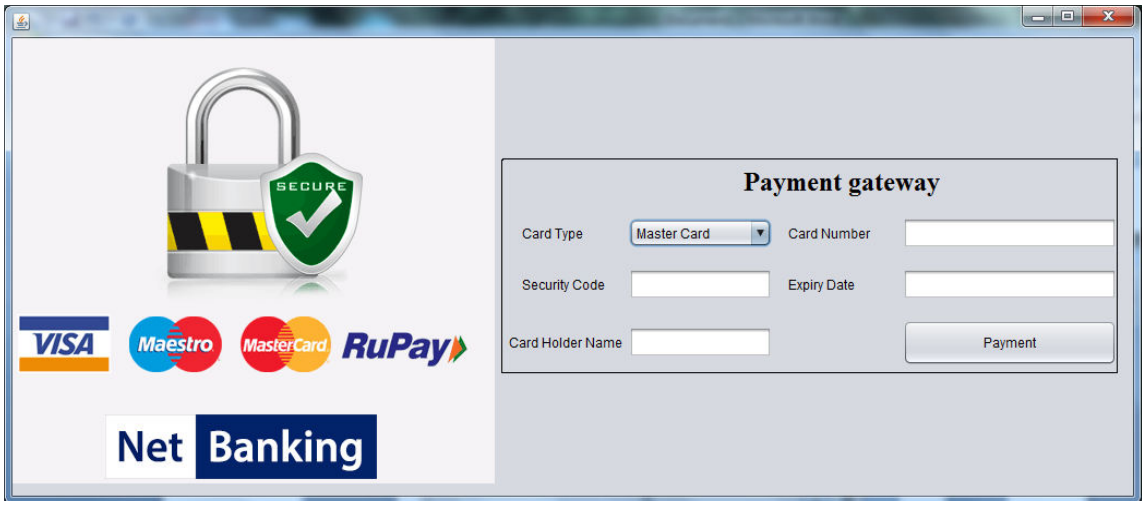
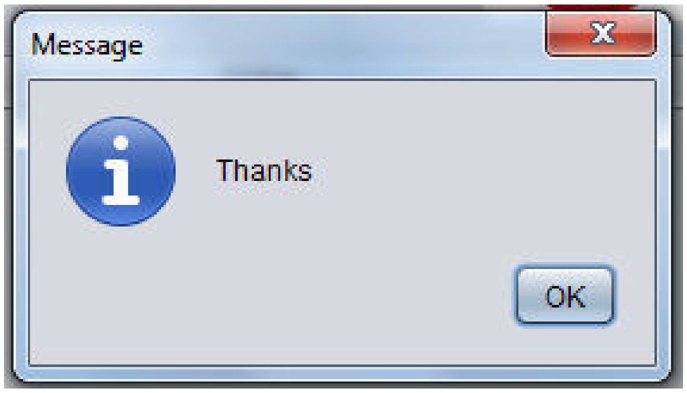


Figure 7.8: Payment through card

This above figure is for the card details for successful payment.



Thanks

Figure 7.9: Completion of a shopping

The above figure (Figure 7.9) displays the successful completion of shopping.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The project titled “Technique for an Efficient Hybrid Shopping Mall”, in the working application of this idea, a facility of combined shopping and pay later facility is tried to be executed. Wherein, the user will be allowed to do shopping normally as done in an offline scenario but will only scan the NFC card for each item and after the completion of the shopping payment has to be made at the payment booths. At the payment booths the customer can modify the cart by viewing the bill and edit accordingly and thus a bill will be generated. After the generation of the bill the customers can collect their bill items for the collection counter.

Modern communication systems and smart phones have tremendously produced positive impact on the life of human beings. These systems make man’s life easier and allow them to perform daily-life activities in a comfortable way. Through our project, we introduce an innovative hybrid model for the shopping malls and propose an advanced purchasing system for the customer. The proposed system reduces the limitations of both online and offline shopping systems and provides an efficient procedure for shopping.

The idea resulted in more customer satisfaction as a customer can view various products and can then purchase the best one. It is customer friendly then the current offline shopping system where customer has to carry shopping bags from one place to another. Here all the purchased product will be billed at a single counter thus reduce the man power in carrying bags. It will also gain attention of more and more customer due to its unique feature.

We have completed the project as per the requirements of our project. Finally we achieved the aim to combine the features of both online and offline shopping. Which will add more comfort to customer and will reduce manpower too?

8.2 Future Scope

1. In terms of future scope, a variety of data mining techniques can be used by researchers to simplify customer perceptions and attitudes.
2. A Bluetooth device can be used to transfer data from device to the software.
3. More smart NFC cards can be used, that can store more information.
4. Using Arduino Boards will reduce the cost of prototyping, allowing companies to iterate more during development, leading to better, more functional products.
5. Interfacing the system with a GSM so that data can be transmitted through messages.

REFERENCES

- X.O. Yang, Z. Xu and A. Gu, “The effects of shopping mall environment on shopping outcomes” 2011 international conference on business management and electronic information (BMEI).vol.4 pp. 110-113, 2011.
- Y. Ha and B.Y. Kim “Shopping mall system with image retrieval based on UML” 2011First ACIS International Symposium on Software and Network Engineering,pp.103-106, 2011.
- J. Swartz “E-commerce and mega machines: identification connectivity, and inference engines” Elsevier Journal on Technology in Society. vol. 23, pp. 159-175, 2001.
- F. Susan, "Consumer and their brands: Developing relationship theory in consumer research," Journal of Consumer Research, 343-373, March 1998.
- Price, Linda and E. J. Arnould, "Commercial Friendships: Service provider-client relationships in context," Journal of Marketing, 63 (October), 38-56, 1999.
- McAlexander, James H., J. W. Scholuten and H. F. Koenig, "Building Brand Community," Journal of Marketing, Vol.66, pp. 38-54, 2002.
- M. B. Akiva and S.R. Lerman, Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press, Cambridge, MA. 1985.
- Leo Louis. Working principle of Arduino and using it As a tool for study and research.[Online] 2016;10.5121/ijcacs.2016.1203
- Arduino Forum [Online]. Available from: http://arduino.cc/forum ; https://store.arduino.cc/usa/arduino-uno-rev3
- NFC Card [Online]. Available from: http://www.smartcardalliance.org/smart-cards-applications-nfc/ ; https://en.wikipedia.org/wiki/Near-field_communication ; https://electronics.howstuffworks.com/nfc-tag.htm;
- Online vs Offline [Online].Available from: http://www.sparkyhub.com/online-shopping-vs-offline-shopping-infographic/
- Reader Module [Online]. Available from: http://www.nskelectronics.com/em-18_rfid_reader.html ; https://electrosome.com/em-18-rfid-reader-arduino-uno/ ; https://electrosome.com/em-18-rfid-reader-raspberry-pi/

- Online Journals [Online]. Available from: https://scholar.google.co.in/ ; https://ieeexplore.ieee.org/Xplore/home.jsp
- Images [Online]. Available from: https://images.google.com/