



JAVA lab report assignment

Java Programming
B.P.Koirala Institute of Health Sciences
74 pag.

Programming in Java

Reeky Tandukar(HND/Fourth Semester)
Arjun Rokaya

Contents

Task 1	4
Introduction	4
Principles of Java.....	4
Characteristics and features of programming in Java	5
Conclusion.....	7
Task 2	8
Introduction	8
Memory Management	8
Primitives vs Objects / auto-boxing	8
Non-Virtual methods	9
Single Paradigm.....	9
Exception Handling in Java	10
Closure.....	10
Floating point arithmetic	10
Look and Feel.....	11
Performance.....	11
Conclusion.....	11
Task 3	12
Introduction	12
Pseudo Code	12
Algorithm	13
Flowchart.....	13
Context Diagram	14
Data Flow Diagram	14
Conclusion.....	14
Task 4	15

Introduction	15
Data and File Structure	15
Classes and Object used	21
Conclusion.....	23
Task 5	24
Introduction	24
Tables created in the database	28
Control Structure	28
Event Handler	33
Integrated Development Environment	35
Database Management System.....	39
Conclusion.....	40
Task 6	41
Introduction	41
Unit Testing	42
Integrated Testing	43
Compatibility Testing	44
Stress Testing.....	45
Critical Review	45
Conclusion.....	47
Task 7	48
Introduction	48
Independent Feedback	48
Conclusion.....	51
Task 8	52
Introduction	52
User Documentation.....	52

Conclusion.....	61
Task 9	62
Introduction	62
Technical Documentation.....	62
Choice of Tools	66
Task Breakdown Approach	69
Data and information collection mechanism	70
Product Deployment.....	71
Maintenance Techniques	72
Conclusion.....	73

Task 1

Discuss the principles, characteristics and features of programming in Java. [1.1, M1]

Introduction

Java is a computer programming language. It enables programmers to write computer instructions using English based commands, instead of having to write in numeric codes. From mobile phones to handheld devices, games and navigation systems to e-business solutions, Java is everywhere (**Oracle.com, 2014**). Once a program has been written, the high-level instructions are translated into numeric codes that computers can understand and execute.

Java is a simple and yet powerful object-oriented programming language and it is in many respects similar to C++. Java originated at Sun Microsystems, Inc. in 1991. It was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. It was developed to provide a platform-independent programming language. Java was designed with a concept of 'write once and run everywhere'. It is a software that is implemented on top of real hardware and operating system. When the source code (.java files) is compiled, it is translated into byte codes and then placed into (.class) files. The JVM executes these bytecodes. So, Java byte codes can be thought of as the machine language of the JVM.

Principles of Java

The four principles of programming in JAVA are inheritance, polymorphism and encapsulation. All the concepts of JAVA are based on classes and its objects. An object is a real-world entity that has some attributes or properties.

1. Inheritance

One of the most important concepts in object-oriented programming is that of inheritance. It allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the base class, and the new class is referred to as the derived class (**Tutorialspoint.com, 2015**).

2. Encapsulation

The whole idea behind encapsulation is to hide the implementation details from users. If a data member is private it means it can only be accessed within the same class. No outside class can access private data member (variable) of other class (**Beginnersbook.com, 2015**). However, if we setup public getter and setter methods to update and read the private data fields then the outside class can access those private data fields via public methods. This way data can only be accessed by public methods thus making the private fields and their implementation hidden for outside classes. That's why encapsulation is also known as data hiding. Object-oriented programming languages rely heavily on encapsulation to create high-level objects. It is also closely related to abstraction.

3. Polymorphism

Polymorphism is generally the ability to appear in many forms. In object-oriented programming, polymorphism refers to a programming language's ability to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes (**Webopedia.com, 2015**). For example, given a base class shape, polymorphism enables the programmer to define different area methods for any number of derived classes, such as circles, rectangles and triangles. No matter what shape an object is, applying the area method to it will return the correct results. Polymorphism is considered to be a requirement of any true object-oriented programming language (OOPL).

4. Abstraction

Abstraction is one of the most important principles in object-oriented software engineering and is closely related to several other important concepts, including encapsulation, inheritance and polymorphism. Abstraction is applied in the process of identifying software artefacts (objects) to model the problem domain. It is the process of reducing these objects to their essence such that only the necessary elements are represented. Abstraction defines an object in terms of its properties, functionality, and interface (**Techopedia.com, 2015**).

Characteristics and features of programming in Java

1. Simple

Java is quite easier than the other popular object-oriented programming languages such as C++ which was the dominant software-development language before Java. Java is designed

with a small number of language constructs so that programmers could learn it quickly. It eliminates several language features available in C/C++ that are associated with poor programming practices or rarely used: go to statements, header files, structures, operator overloading, multiple inheritance and pointers.

2. Object-oriented

Java is an object-oriented language, which means that we focus on the data in our application and methods that manipulate that data, rather than thinking strictly in terms of procedures. In an object-oriented system, a class is a collection of data and methods that operate on that data. Taken together, the data and methods describe the state and behavior of an object. Classes are arranged in a hierarchy, so that a subclass can inherit behavior from its superclass. Java comes with an extensive set of classes, arranged in packages that we can use in our programs (**Web.cs.wpi.edu, 2015**).

3. Secure

Java is secure and never uses memory pointers. All applications in Java are run under the sand box. Security administrator establishes the accessibility alternatives of a class such as reading and writing a file to the local disk. Java utilizes the public key encryption process to enable the Java programs to transfer over the web in a protected and encrypted form. Java program does not harm other system thus making it secure. It provides a secure means of creating Internet applications and provides secure way to access web applications. The security is based on the premise that nothing should be trusted.

4. Portable

The Java programs can be executed in any environment for which there is a Java run-time system (JVM). It can be run on any platform such as Linux, Windows or Mac. And it also can be transferred over the world wide web. It ensures that other implementation-dependent aspects of language specification are eliminated. For example; Java specifies the sizes of primitive data types and their arithmetic behavior (**Borysowich, 2015**).

5. Robust

Robust refers to reliability. Java encourages error-free programming by being strictly typed and performing run-time checks. Java puts a lot of emphasis on early checking for possible errors, because Java compilers can detect many problems that would first show up at the

execution time in other languages. Java has eliminated certain types of error-prone programming constructs found in other languages. It does not support pointers, for example, thereby eliminating the possibility of overwriting memory and corrupting data. Java has a runtime exception-handling feature to provide programming support for robustness. Java forces the programmer to write the code to deal with exceptions. Java can catch and respond to an exceptional situation so that the program can continue its normal execution and terminate gracefully when a runtime error occurs.

6. Dynamic

Java is a dynamic language. It is capable of linking dynamic new classes, methods and objects. It supports functions written in C and C++ also. These functions are called native methods. During Run-Time Native, methods can be linked dynamically. Java programs carry with them substantial amounts of run-time type information that is used to verify and resolve accesses to objects at run time.

7. High performance

Java programs provides the high performance. The bytecodes are highly optimized and JVM can be executed much faster. The software programs designed in Java are well performed and are highly overrated.

Conclusion

Java is a powerful language with the strong features like security, dynamics, robust, etc. Java is an OOP language considering the OOP principles like encapsulation, inheritance, polymorphism and data abstraction. It is one of the most secured and popular programming languages that develops the application for the platforms of both web-based and desktop-based. In this task, I have discussed the principles, characteristics and features of programming in Java. For this task to be completed, I have used various resources and gathered various information regarding Java and the programming language that has been cited in this task.

Task 2

Critically evaluate the environment flexibility of programming in java. [1.2]

Introduction

The Departmental Store has appointed me as a system developer and designer. They want me to develop them a system which should facilitate them according to their requirements. Before the development phase, they want to know on what environment I will be developing the system and the flexibility of language that I am going to use. Considering the facts, in this task, I will be evaluating the environment flexibility of programming in java to provide the Departmental Store an assurance that their required system is on the right track.

Memory Management

Memory management is the process of allocating new objects and removing unused objects to make space for those new object allocations. Java objects reside in an area called the heap. The heap is created when the JVM starts up and may increase or decrease in size while the application runs. When the heap becomes full, garbage is collected. During the garbage collection objects that are no longer used are cleared, thus making space for new objects. JVM uses more memory than just the heap. For example, Java methods, thread stacks and native handles are allocated in memory separate from the heap, as well as JVM internal data structures (Docs.oracle.com, 2015).

An alternate approach to memory management that is now commonly utilized, especially by most Java, is automatic management by a program called a garbage collector. Automatic memory management enables increased abstraction of interfaces and more reliable code. Garbage collection avoids the dangling reference problem, because an object that is still referenced somewhere will never be garbage collected and so will not be considered free.

Primitives vs Objects / auto-boxing

Autoboxing is the automatic conversion that the Java compiler makes between the primitive types and their corresponding object wrapper classes. For example, converting an **int** to and **Integer**, a **double** to a **Double**, and so on.

Data stored in memory is a string of bits (0 or 1). In Java, we must make the context explicit by specifying the type of the data. Java has two categories of data:

- Primitive data i.e. numbers, characters
- Object data i.e. programmer defined types

According to **Programmerinterview.com (2015)**, “There are 8 primitive data types such as byte, short, int, long, double, char and Boolean. Primitive data are only single values; they have no special capabilities. These types serve as the building blocks of data manipulation in Java. Such types serve only one purpose of containing pure and simple values of a kind. Because these data types are defined into the Java type system by default, they come with a number of operations predefined. We cannot define a new operation for such primitive types.”

Type	Size	Range	Default
boolean	1 bit	true or false	false
byte	8 bits	[-128, 127]	0
short	16 bits	[-32,768, 32,767]	0
char	16 bits	['\u0000', '\uffff'] or [0, 65535]	'\u0000'
int	32 bits	[-2,147,483,648 to 2,147,483,647]	0
long	64 bits	$[-2^{63}, 2^{63}-1]$	0
float	32 bits	32-bit IEEE 754 floating-point	0.0
double	64 bits	64-bit IEEE 754 floating-point	0.0

Fig: Primitive data types in Java

An object is a large chunk of memory that can potentially contain a great deal of data along with methods (little programs) to process that data. There are thousands of object classes that come standard with Java, and a programmer can easily create additional classes.

Non-Virtual methods

In Java, all the methods are the virtual methods. That means, the most derived method is always called. Non-virtual methods are resolved statically i.e. the member function is selected statically based on the type of the pointer to the object. In contrast, virtual methods are resolved dynamically i.e. the member function is selected dynamically based on the type of the object., not the type of the pointer/reference to that object. The non-virtual methods are called directly using its address.

Single Paradigm

Java is a high-level, general-purpose, mostly single-paradigm, statically typed programming language. **Weblogs.java.net (2015)**, defines programming paradigm is the manner in which programming elements such as functions, objects and variables are exploited to produce the desired output. Java is predominantly a single-paradigm language. The addition of static imports in Java 5.0 accommodates the procedural paradigm better than earlier versions of Java. It follows Object Oriented Programming paradigm.

Exception Handling in Java

Exception is an error event that can occur during the execution of a program and disrupts its normal flow. Java provides a robust and object-oriented way to handle exception scenarios known as Java Exception Handling. Exceptions in java can arise from different kind of situations such as wrong data entered by user, hardware failure, network connection failure, Database server down etc. The term exception means exceptional condition, it is a problem that may arise during the execution of program. A bunch of things can lead to exceptions, including programmer error, hardware failures, files that need to be opened cannot be found, resource exhaustion etc. (Studytonight.com, 2015). The exceptions are divided into 3 categories:

- **Checked exceptions**
- **Unchecked exceptions**
- **Errors**

Closure

A closure is a first-class function with bound variables. It means that we can pass the closure as a parameter to other functions. It stores the value of some variables from the lexical scope that existed at the time that is created. As defined by Christopher Strachey in 'Understanding Programming Languages', a first-class object can be stored in a data structure, passed as a parameter, can be returned from a function, can be constructed at runtime and independent of any identity (Javac.info, 2015).

Floating point arithmetic

Floating-point arithmetic is the standard way to represent and work with non-integer numbers in a digital computer. It is designed to create the illusion of working with real numbers in a machine that can only strictly work with a finite set of numbers. In many cases, calculations performed with floating-point numbers will produce results which are not significantly different from those obtained using real numbers. However, in many other situations thinking of floating-point numbers as real numbers can be highly misleading (Macaulay.ac.uk, 2015).

The JVM's floating-point support follows to the IEEE-754 1985 floating-point standard. This standard defines the format of 32-bit and 64-bit floating-point numbers and defines the operations upon those numbers. In the JVM, floating-point arithmetic is performed on 32-bit

floats and 64-bit doubles. For each byte code that performs arithmetic on floats, there is a corresponding byte code that performs the same operation on doubles.

Look and Feel

Look and Feel in Java is upgrading slowly with each new version. The Java Look and Feel creates an instance of `MetalTabbedPaneUI` to provide the L&F for `JTabbedPane`. The actual creation of the UI delegate is handled by Swing for us—for the most part we never need to interact directly with the UI delegate.

Performance

Java performance is a matter of concern because lots of business software has been written in Java after the language quickly became popular in the late 1990s and early 2000s, and concerns over its performance led to the development of specialized hardware able to run Java directly, dubbed Java processors. The performance of a compiled Java program depends on how optimally its particular tasks are managed by the host Java Virtual Machine (JVM), and how well the JVM takes advantage of the features of the hardware and OS in doing so. Thus, any Java performance test or comparison has to always report the version, vendor, OS and hardware architecture of the used JVM. In a similar manner, the performance of the equivalent natively compiled program will depend on the quality of its generated machine code, so the test or comparison also has to report the name, version and vendor of the used compiler and its activated optimization directives (**Freejavaguide.com, 2015**).

Conclusion

Java programming language is the built-on C++ for a simple-to-use language. It enables programmers to write programs on simple and flexible language. In this task, I had evaluated the environmental flexibility of programming in Java. The Departmental Store will surely be able to identify and know the programming language that is used for developing their application is flexible. Hence, the conclusion is that the Java programming language could provide the flexible environment and satisfy the requirement of the Departmental Store.

Task 3

Design a Java programming solution to the problem given in the scenario. [2.1 M2]

Introduction

Departmental Store wants the system to operate even in no-lights condition. For the Departmental Store, the Inventory Management System is to be made using the Java Programming language. Before developing the program, it is very essential to sketch everything from the scratch in order to avoid the problem that might face in the future. In this task, I will be considering the Pseudo-code/algorithm/flowchart, Context diagram and 0-level DFD diagram according to the scenario of the Departmental Store.

Pseudo Code

Pseudo code can be defined as the notation that resembles the programming language into the simplified form. **Bennett (2018)** says that, “Pseudocode is an informal way of programming description that does not require any strict programming language syntax or underlying technology considerations.” Pseudo code is not an actual programming language but it is a guide or a set of syntaxes of codes that are written by the programmer to understand the programming data flow. With the help of pseudo code, one can understand the program in less time span. It enables the programmer to concentrate only on the algorithm part of the code development. The pseudo code has also been written for the web application of the leads management system.

Login

1. OPEN Login form
2. GET username and password
3. CHECK username and password in database
4. IF username and password is incorrect
 - a. PRINT “login failed”
5. IF username and password is correct
 - a. Start session and open “Homepage”
6. END

Add Cashier (Only from Administrator account)

1. Click on Cashier
2. SET the credentials.
3. Click on Add New

Update Cashier (Only from Administrator account)

1. On Cashier details, click on the cashier that you want to update.
2. Edit the credentials
3. Click on Update.

Delete Cashier (Only from Administrator account)

1. On Cashier details, click on the cashier that you want to delete.
2. Click on Delete.






Algorithm

An algorithm is a specific set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminate at some point (Mathworld.wolfram.com, 2014). A computer program can be viewed as an elaborate algorithm. In computer science, an algorithm usually means a small procedure that solves a recurrent problem. The algorithm for Departmental Store is given below:

Login	
Step 1	Start
Step 2	Enter Username and password
Step 3	Check username and password
Step 4	If username and password is correct then Display Homepage
Step 5	If username and password is incorrect then Display Error message
Step 6	End

Flowchart

A flowchart visually represents and organizes the steps used to write the program—it is a diagram of the “flow” of the process. When programmers write code, they need to give the robot instructions that are both sequential and specific. Flowcharts enable programmers to work these steps out before needing to translate their behaviors into code.

Flow Chart Symbol	Meaning	Explanation
	Start and End	The Symbol denoting the beginning and end of the flow chart.
	Process	This symbol shows that the user performs/process a task.
	Decision	This symbol represents a point where a decision is made.
	Action	This symbol means that the user performs an action.
	Flow Line	A line that connects the various symbols in an ordered way.

Context Diagram

Context diagram is the graphical representation of the system that shows the relationship of the system with the external bodies such as systems, organizational groups, external data stores, etc. The context diagram provides information of the interfaces and boundaries of the project or the process at hand. It also shows the interactions of the system with the other systems and users. The elements used in the context diagram are ovals, rectangle and arrows. The context diagram has many benefits, such as;

- Easily understood and can easily be designed.
- Easy to expand by adding other levels.
- Shows the scope and boundaries of a system at a glance including the other systems that interface with it
- No technical knowledge is assumed or required to understand the diagram

Data Flow Diagram

Data flow diagram (DFD) is one of the system investigation technique that shows how the data is processes by the system relating to the inputs and outputs. It represents the flow of information, data travelling and collection and storage of the data. In simple terms, it maps out the flow of data in the system. The DFD is drawn using the defined symbols like rectangles, circles and arrows, etc. The DFD has been used as a system analysis tool in order to investigate the system called Academic Portal in the Falcon college. The data flow diagram has many benefits, such as;

- Easily recognizable.
- More information can be gathered.
- Provides ease in communicating with the system to gain knowledge.
- Provides top leveled system overview.

Conclusion

This was the task where I had to design the Java programming solution for the Departmental Store. The solution is named as Inventory Management System. The solution has been initialized with different diagrams, algorithms and flowcharts and also pseudo code. The design that I have implemented in this task will help me while coding and developing the solution. Hence, I do not have to keep every functional step in my mind, and can look at this task as soon as I get confused in developing, without missing any of the requirements. In this way the design of Departmental Store is completed.

Task 4

Explain the components and data and file structures required to implement the given design.

Introduction

For the solution of the Departmental Store to be completed, the various components and the data and file structures are required. The layout must be clear and understandable. For the Java programming solution of the Departmental Store, I will be using eclipse as an integrated development environment (IDE). Referring to the task 3, I will be explaining the components and data file structures required to implement the given design.

Data and File Structure

A **data structure** is a specialized format for organizing and storing data. Data structure refers to methods of organizing units of data within larger data sets. Achieving and maintaining specific data structures help improve data access and value. Data structures also help programmers implement various programming tasks (**Rouse, 2015**). General data structure types include the array, the file, the record, the table, the tree, and so on. Any data structure is designed to organize data to suit a specific purpose so that it can be accessed and worked with in appropriate ways. In computer programming, a data structure may be selected or designed to store data for the purpose of working on it with various algorithms.

A **file structure** is the format in the program that contains everything that defines the workplace from source code and assists to test code and build configurations. While developing the solution for the Departmental Store, the necessary file structure has been created. The file structure of the Inventory Management System is shown below:

<p>Package Explorer</p> <ul style="list-style-type: none"> coreproject QBill QuickBill <ul style="list-style-type: none"> JRE System Library [JavaSE-1.8] src <ul style="list-style-type: none"> Com.ismt.qbill.client <ul style="list-style-type: none"> AddCashier.java AddNewBill.java AdminDashBoard.java cashierSalesDetails.java CashierHomePage.java CashierLogin.java HomePage.java IndexPage.java InvoiceForm.java login.java LoginForm.java ProductForm.java ProductInfo.java RegisterForm.java SalesDetails.java SearchBill.java StockForm.java TransactionListForm.java Com.ismt.qbill.db Com.ismt.qbill.model Com.ismt.qbill.service image Referenced Libraries <ul style="list-style-type: none"> jcalendar-1.4.jar 	<h3>Add Cashier</h3> <ul style="list-style-type: none"> AddCashier.java <ul style="list-style-type: none"> AddCashier <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> address contentPane email mobilenno name password table uid username AddCashier() displayData() : void
<h3>Add New Bill</h3> <ul style="list-style-type: none"> AddNewBill.java <ul style="list-style-type: none"> AddNewBill <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane panel <ul style="list-style-type: none"> panel_1 <ul style="list-style-type: none"> table textField textField_1 textField_2 textField_3 textField_4 textField_5 textField_6 textField_7 textField_8 AddNewBill() 	<h3>Admin Dashboard</h3> <ul style="list-style-type: none"> AdminDashBoard.java <ul style="list-style-type: none"> AdminDashBoard <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane AdminDashBoard()

Fig: File Structure of
Inventory Management System

<p>Cashier Sale Detail</p> <ul style="list-style-type: none"> cashierSalesDetails.java <ul style="list-style-type: none"> cashierSalesDetails <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane lblStockDetails scrollPane table cashierSalesDetails() displayData() : void getLblStockDetails() : JLabel 	<p>Cashier Home page</p> <ul style="list-style-type: none"> CashierHomePage.java <ul style="list-style-type: none"> CashierHomePage <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane panel CashierHomePage()
<p>Cashier Login Page</p> <ul style="list-style-type: none"> CashierLogin.java <ul style="list-style-type: none"> CashierLogin <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane label_1 lblBack lblForgotPassword lblLoggingIn lblLogin lblNewLabel lblNewLabel_1 lblPassword loader login loginbtn panel panel_1 panel_2 passwordtxt usernametxt CashierLogin() <ul style="list-style-type: none"> getLabel_1() : JLabel getLblBack() : JLabel getLblForgotPassword() : JLabel getLblLoggingIn() : JLabel getLblLogin() : JLabel getLblNewLabel() : JLabel getLblNewLabel_1() : JLabel getLblPassword() : JLabel getLogin() : JPanel getLoginbtn() : Button getPanel() : JPanel getPanel_1() : JPanel 	<p>Home page</p> <ul style="list-style-type: none"> HomePage.java <ul style="list-style-type: none"> HomePage <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane HomePage()

Invoice page	
<ul style="list-style-type: none"> InvoiceForm.java <ul style="list-style-type: none"> InvoiceForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> availableTxt btnPring btnSubmit comboBox contentPane CustomernameTxt dateChooser InvoiceCtxt lblCustomerName lblDate lblInvoiceNo lblPrice lblSelectedItem lblTotal lblUnits lblUnitsLeft panel pdao pricetxt returnbtn scrollPane separator separator_1 table totalamountTxt unitstxt InvoiceForm() <ul style="list-style-type: none"> displayData() : void getAvailableTxt() : JLabel 	<ul style="list-style-type: none"> InvoiceForm() <ul style="list-style-type: none"> displayData() : void getAvailableTxt() : JLabel getBtnPring() : JButton getBtnSubmit() : JButton getComboBox() : JComboBox getCustomernameTxt() : JTextField getDateChooser() : JDateChooser getInvoiceCtxt() : JTextField getLblCustomerName() : JLabel getLblDate() : JLabel getLblInvoiceNo() : JLabel getLblPrice() : JLabel getLblSelectedItem() : JLabel getLblTotal() : JLabel getLblUnits() : JLabel getLblUnitsLeft() : JLabel getPanel() : JPanel getPricetxt() : JTextField getReturnbtn() : Button getScrollPane() : JScrollPane getSeparator() : JSeparator getSeparator_1() : JSeparator getTable() : JTable getTotalamountTxt() : JTextField getUnitstxt() : JTextField loadProductInCmb() : void

Login Page	
<ul style="list-style-type: none"> login.java <ul style="list-style-type: none"> login <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane password username login() <ul style="list-style-type: none"> 	

Admin Login

<ul style="list-style-type: none"> LoginForm.java <ul style="list-style-type: none"> LoginForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane label_1 lblBack lblForgetPassword lblLoggingIn lblLogin lblNewLabel lblNewLabel_1 lblPassword lblRegisterHere loader login loginbtn panel panel_1 panel_2 passwordtxt usernametxt LoginForm() <ul style="list-style-type: none"> getLabel_1() : JLabel getLblBack() : JLabel getLblForgetPassword() : JLabel getLblLoggingIn() : JLabel getLblLogin() : JLabel getLblNewLabel() : JLabel getLblNewLabel_1() : JLabel getLblPassword() : JLabel getLblRegisterHere() : JLabel getLogin() : JPanel 	<ul style="list-style-type: none"> LoginForm() <ul style="list-style-type: none"> getLabel_1() : JLabel getLblBack() : JLabel getLblForgetPassword() : JLabel getLblLoggingIn() : JLabel getLblLogin() : JLabel getLblNewLabel() : JLabel getLblNewLabel_1() : JLabel getLblPassword() : JLabel getLblRegisterHere() : JLabel getLogin() : JPanel getLoginbtn() : Button getPanel() : JPanel getPanel_1() : JPanel getPanel_2() : JPanel getPanel_2_1() : JPanel getPasswordField_1() : JPasswordField getUsernametxt() : JPasswordField
---	---

Product form

<ul style="list-style-type: none"> ProductForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane lblCustomerName lblName lblNewBill lblQuantity mrptxt newbtn panel panel_1 productname.txt qtyavailable.txt updatebtn ProductForm() <ul style="list-style-type: none"> getLblCustomerName() : JLabel getLblName() : JLabel getLblNewBill() : JLabel getLblQuantity() : JLabel getNewbtn() : Button getPanel() : JPanel getPanel_1() : JPanel getProductname.txt() : JPasswordField getQtyavailable.txt() : JPasswordField getTextField_5() : JPasswordField getUpdatebtn() : Button

Product Information

<ul style="list-style-type: none"> ProductInfo.java <ul style="list-style-type: none"> ProductInfo <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane table textField ProductInfo() <ul style="list-style-type: none">
--

Registration form

<ul style="list-style-type: none"> RegisterForm.java <ul style="list-style-type: none"> RegisterForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane <ul style="list-style-type: none"> fname.txt label label_1 lblFirstName lblLastName lblNewLabel lblPassword lblRegisterToBilling lblRetypePass lblSignUp lblTheMagical lblUsername lname.txt panel passwordField password.txt registerbtn separator separator_1 separator_9 Username.txt 	<ul style="list-style-type: none"> Username.txt RegisterForm() <ul style="list-style-type: none"> getFname.txt() : JTextField getLabel() : JLabel getLabel_1() : JLabel getLblFirstName() : JLabel getLblLastName() : JLabel getLblNewLabel() : JLabel getLblPassword() : JLabel getLblRegisterToBilling() : JLabel getLblRetypePass() : JLabel getLblSignUp() : JLabel getLblTheMagical() : JLabel getLblUsername() : JLabel getLname.txt() : JTextField getPanel() : JPanel getPasswordField() : JPasswordField getPassword.txt() : JPasswordField getRegisterbtn() : JButton getSeparator() : JSeparator getSeparator_1() : JSeparator getSeparator_9() : JSeparator getUsername.txt() : JTextField
--	--

Sales Detail

<ul style="list-style-type: none"> SalesDetails.java <ul style="list-style-type: none"> SalesDetails <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> btnAddNewProduct contentPane lblStockDetails scrollPane table SalesDetails() <ul style="list-style-type: none"> displayData() : void getBtnAddNewProduct() : JButton getLblStockDetails() : JLabel

Search Bill

<ul style="list-style-type: none"> SearchBill.java <ul style="list-style-type: none"> SearchBill <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> contentPane table textField SearchBill() <ul style="list-style-type: none"> displayData(int) : void
--

Stock Details

<ul style="list-style-type: none"> StockForm.java <ul style="list-style-type: none"> StockForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> btnAddNewProduct contentPane lblStockDetails scrollPane table StockForm() <ul style="list-style-type: none"> displayData() : void getBtnAddNewProduct() : JButton getLblStockDetails() : JLabel
--

Transaction Details

<ul style="list-style-type: none"> TransactionListForm.java <ul style="list-style-type: none"> TransactionListForm <ul style="list-style-type: none"> main(String[]) : void <ul style="list-style-type: none"> btnPring contentPane lblTransactionDetails panel pdao returnbtn scrollPane table TransactionListForm() <ul style="list-style-type: none"> displayData() : void getBtnPring() : JButton getLblTransactionDetails() : JLabel getPanel() : JPanel getReturnbtn() : JButton getScrollPane() : JScrollPane getTable() : JTable

Classes and Object used

Following the OOP principle various class and objects are used in the application. Class and objects used and their use in the application is given below:

Inbuilt Classes

Class	Object	Work
Intent	Intent intent = new Intent(context,classname)	Used to open one activity from another activity
Bundle	Bundle b = new Bundle()	Used to pass a string or integer value from one activity to another
Alert Dialog. Builder	AlertDialog.Builder alertDialog = new AlertDialog.Builder(context());	Used to create an alert dialog
Cursor	Cursor c= SQLiteDatabase.rawQuery(string, null)	Used to display specific message for a limited period of time
Toast	Toast toast = Toast.makeText(context,duration).show();	Used to display specific message for a limited period of time
LayoutInflater	LayoutInflater inflater = getLayoutInflater();	Used to inflate a custom layout for a dialog, list view, menu etc.
ArrayAdapter	ArrayAdapter<> arrayadapter= new ArrayAdapter<>(Context context, int) Resource, List<T> (objects)	Used to fill data in ListView.
Fragment Transaction	FragmentTrasaction fragmentTransaction = getSupportFragmentManager().beginTransaction();	Used for the transaction from one fragment to another.

User created class

Similarly, there are various user created class to the convenience and to handle various activities and requests from the user and meet the requirements of the user. User created classes their objects and their use are given below

class	object	Description
Dbconnection.java	Dbconnection db = new dbconnection(context);	Created to connect database and application.
AddBill.java	Addproduct frame= new Addproduct()	Created for adding products

searchBillFrom.java	SearchBill frame = new serachProduct();	Created in order to search for bill
InvoiceFrom.java	InvoiceFrom frame= new Invoicefrom();	Created to add customer details.
SalesDetails.java	SalesDetails frame = new SalesDetails();	Created to display sales details
HomePage.java	HomePage frame = new HomePage();	Created to display a home page and navigate through other pages
IndexPage	IndexPage frame = new IndexPage	Created as the login page
ProductInfo	ProductInfo frame = new ProductInfo	Created to search for product information
Login.java	Login frame = new Login	Created to login to the application
LoginFrom.java	LoginFrom = new LoginForm	Create to provide and use login form
RegisterFrom.java	RegisterFrom frame = new RegisterFrom	Created to register admin
StockFrom.java	stockFrom frame = new StockFrom	Created to view stock information.
Addcashier	Addcashier frame new Addcashier	Created to display cashier details
AddNewBill	AddNewBill frame =new AddNewBill	Created to add new bill and details
cashierSaleDetail.java	cashierSaleDetail frame = new cashierSaleDetail	Created to provide sale details
CashierHomePage.java	CashierHomePage frmae = new CashierHomePage	Created as the cashier home page and to help cashier navigate through pages
AdminDashBoard.java	AdminDashBoard frame = new AdminDashBoard	Created as dashboard and allows the user to navigate different activities in the application
CashierLoginPage.java	CashierLoginPage frame = new CashierLoginPage	Created as a cashier login page to login to the cashier
ProductFrom.java	ProductFrom frame new ProductFrom	Created to display product details

TransactionListForm.java	TransactionListForm frame =new TransactionListForm	Created to display transaction details.
--------------------------	---	---

Conclusion

In this task, I have explained the components, data and file structures required to implement the programming solution for the Departmental Store. The solution has both feasible GUI and Database system. Moreover, different user has been created and inbuilt class and objects are used which even provide future reference to add another page with very few lines of codes saving time.

Task 5

Consider: Event handling, control structures, conditionals, loops, exception handling and error reporting mechanisms, effective use of IDE using code and screen templates.

Introduction

The Java programming solution is developed in Eclipse Integrated Development Environment. The solution has been completely designed and developed. In this task, I will be implementing a Java programming solution based on the prepared design and define the relationships with the objects. I will be considering event handling, control structures, conditionals, loops, exception handling and error reporting mechanisms.

Design of Inventory Management System and its codes

Index Page



Login Page

Back

Admin Login

Sanjil

Forget Password?

Login

Register Here

```

public LoginForm() {
    setForeground(Color.DARK_GRAY);
    setTitle("Admin Login");
    setType(Type.UTILITY);
    setResizable(false);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 842, 591);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    contentPane.add(getPanel());
}

```

Register Cashier

Register to Billing System

"Life is for Service"
- Fred Rogers

5

Sign Up

First Name

Last Name

Username

Password

Retype Pass

Register

```

public RegisterForm() {
    setForeground(Color.GRAY);
    setBackground(Color.WHITE);
    setTitle("Register");
    setResizable(false);
    setType(Type.UTILITY);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 1049, 669);
    contentPane = new JPanel();
    contentPane.setBackground(Color.WHITE);
    contentPane.setBorder(null);
    setContentPane(contentPane);
    contentPane.setLayout(null);
    contentPane.add(getPanel());
    contentPane.add(getLblSignUp());
    contentPane.add(getLblFirstName());
    contentPane.add(getSeparator_2());
    contentPane.add(getFnameetxt());
    contentPane.add(getLblLastName());
    contentPane.add(getLnameetxt());
    contentPane.add(getSeparator_3());
}

```

Product Form

Product Details

Product Name

Quantity Available

MRP

```

public ProductForm() {
    setResizable(false);
    setType(Type.UTILITY);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(350, 100, 715, 569);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    contentPane.add(getPanel());
}

private JPanel getPanel() {
    if (panel == null) {

```

Invoice form

Customer Name Date

Invoice No. Units Left:

Select Item Price Units Total

CustomerName	Item	Unit	TotalPrice
Roman Poudel	Dell Laptop	30	2100000.0
Hari	Printer	11	88000.0
Gopal	Samsung TV	6	180000.0
Gopal	Washing Machine	8	480000.0
Ram	Vacuum Cleaner	2	40000.0
Alisa	Samsung TV	10	300000.0
Kishor	AC	5	400000.0
pawan	Washing Machine	5	300000.0
Ruesh	acer Laptop	30	1500000.0

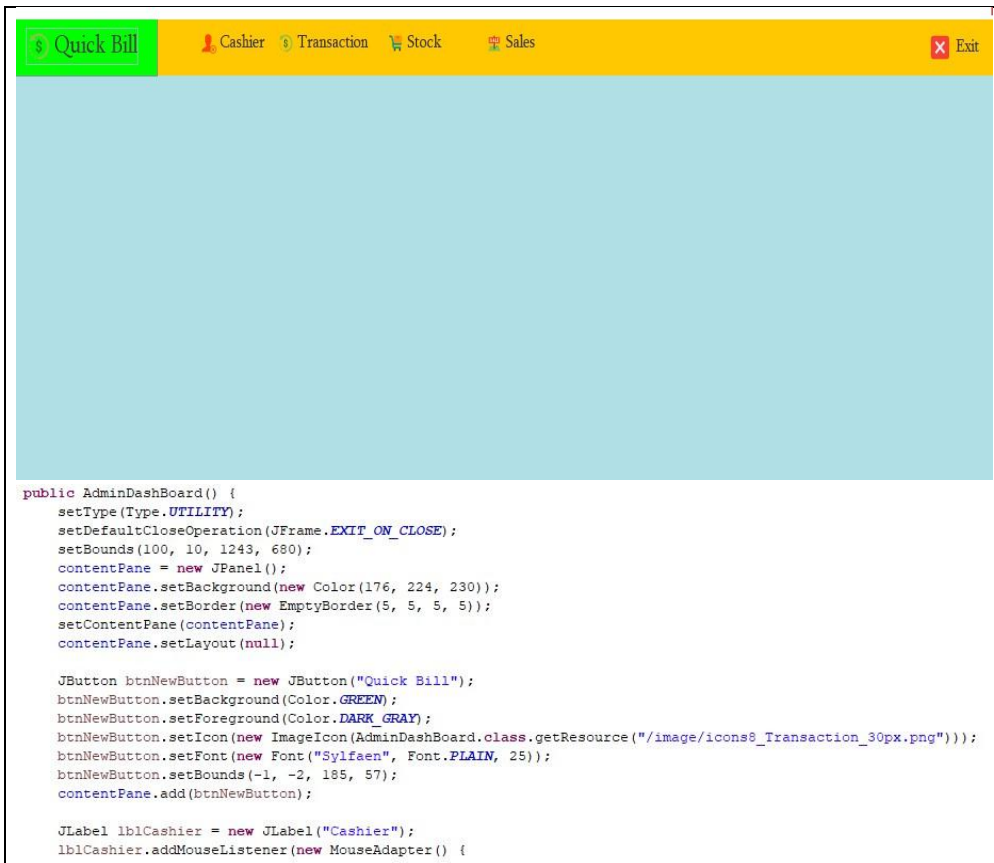
```

public InvoiceForm() {
    setResizable(false);
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(180, 100, 1000, 548);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);
    contentPane.add(getPanel());

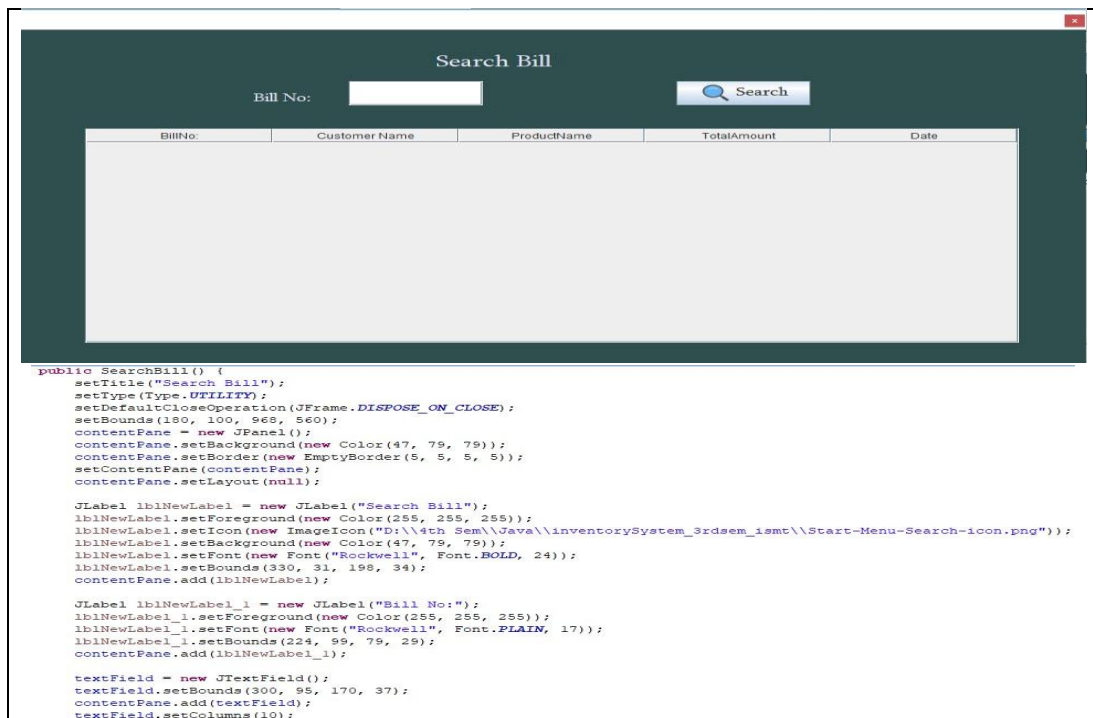
    displayData();
}

```

Admin Dashboard

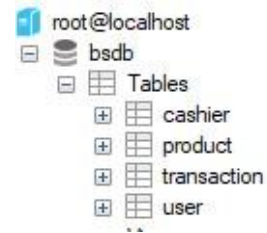


Search Bill



Tables created in the database

For the database, I had created four tables, such as cashier, product, transaction and user. These tables play an important role in the completion of the solution.



- **Cashier:** This table is made for putting away the id, username, name, secret word, telephone, address, email and mobile.no of the cashier. While including another cashier, admin require these details about cashier. Likewise, cashier require a portion of these information to login to the Inventory Management System.
- **Product:** This table is made for storing id, name, quantity, price and quantity available. Administrator manages the product information. The product information can be accessed to by administrator, cashier and client. Client require these data while purchasing products.
- **Transaction:** This table is created with id, customer name, invoice no, invoice information, total price, product name and available quantity. This information is required while creating invoice for customer. Cashier can produce the invoice for the client. Both administrator and cashier can view the details of transaction.
- **User:** This table is created for storing id, first name, last name, username, password, date of birth and address. The username, id, first name, last name and password of administrator and cashier are storing away while registering their account.

Control Structure

A control is a statement to be executed once, several times, or not at all. Control structures make up some of the statements of the Java language. Generally programming is composed of three types of structures, these control structures are:

- Sequential structure
- Iteration structure
- Selection structure

Sequential Structure

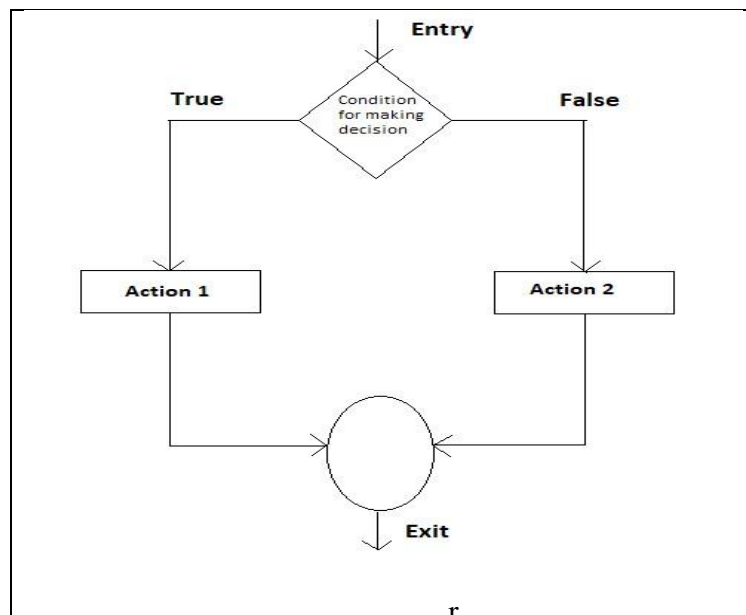
This is the default structure used by every compiler. In this structure the program instructions are executed in the order that they appeared in the program. The compiler scans the program, instruction by instruction and run the instruction one by one from top to bottom.

Iteration Structure

If a group of instructions is executed repeatedly, until some logical condition has been satisfied. Then the structure is called Iteration structure, repetition structure or looping. To express repetitive execution of statements we use another construct, Java's while statement. A while statement is often called a loop. A while loop contains a Boolean valued expression which is known as its test (or sometimes its condition or sometimes its guard). It also contains a statement which is the body of the loop.

Selection Structure

Selection structure is also known as conditional logic, used to execute a set of instructions depending over a condition. This structure is also known as branching. It first evaluates an expression to decide between the possibilities. If there are only two, then a Boolean valued expression (also called a logical expression) will be enough to allow us to choose.



a. If Statement

An if statement consists of a condition followed by one or more statements

Syntax of If Statement:

```
if (condition) { statement
    (s);
```

```
}
```

Use in application

```
if (panel == null) {
    panel = new JPanel();
    panel.addMouseListener(new MouseAdapter() {
    });
    panel.setBackground(Color.WHITE);
    panel.setBounds(0, 0, 1008, 627);
    panel.setLayout(null);
    panel.add(getPanel_1());
    panel.add(getPanel_2());
    panel.add(getPanel_3());
}
```

b. If Else Statement

An if statement can be followed by an optional else statement, which executes when the condition is false.

Syntax of If Else Statement:

```
if (condition) {
    statement(s);
} Else {
    statement(s);
}
```

Use in application

```
if(adao.adminLogin(un, psw)) {
    JOptionPane.showMessageDialog(null, "Login Success!");
    new AdminDashBoard().setVisible(true);
    dispose();
} else {
    JOptionPane.showMessageDialog(null, "Login Failed!");
}
```

c. If Else If Statement

An if else if statement is to check multiple condition. It checks all the condition one by one and execute finding the true condition.

Syntax of If Else If Statement:

```

if (condition 1) {
    statement (s);
} else if (condition
2) {
    statement (s);
} else
{
    statement (s);
}

```

d. Nested If Statement

It is always legal to nest if-else statements which means, can use one if or else if statement inside another if or else if statement.

Syntax of Nested If Statement:

```

if (condition 1) {
    statement (s);
if (condition 2) {
    statement (s);
}
}

```

e. Switch Statement

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.

Syntax of Switch Statement

```

switch(expression) {
    case value:
        Statements

```

break; case

value:

Statements

break;

default:

Statements

}

Statement	Effects
if (x == 0) System.out.println("zero");	Prints "zero" if x has the value 0.
If (x == 0) System.out.println("zero"); If (x != 0) System.out.println("non-zero");	Prints "zero" if x has the value 0, prints "nonzero" otherwise.
If (x == 0) System.out.println("zero"); Else System.out.println("non-zero");	Prints "zero" if x has the value 0, prints "nonzero" otherwise.
If (x == 0) System.out.println("zero"); Else If (x > 0) System.out.println("positive"); Else System.out.println("negative");	Prints "zero" if x has the value 0, prints "positive" if x is positive, Prints "negative" if x is negative.
If (x != 0) If (x > 0) System.out.println("positive"); Else System.out.println("negative");	Prints "positive" if x is positive, and non-zero. Prints "negative" if x is negative.

<pre>Switch (x) { Case 0: System.out.println("zero"); break; default: System.out.println("non-zero"); }</pre>	Prints "zero" if x has the value 0, prints "nonzero" otherwise.
--	---

Event Handler

Even means the change in state of an object i.e. event describes the change in state of source. Event handling is a mechanism which controls the event and specify what to be done if event occurs. This mechanism has the code which is known as event handler that is executed when an event occurs. According to **javaworld.com** Event class is the primary player in the event game. It attempts to capture the fundamental characteristics of all user-generated events. For example, click on button, dragging mouse etc. The java.awt.event package provides many event classes and Listener interfaces for event handling.

Event Classes	Description	Listener Interface
TextEvent	generated when value of textarea or textfield is changed	TextListener
WindowEvent	generated when window is activated, deactivated, deiconified, iconified, opened or closed	WindowListener

Error Handling and Exceptional Handling

Error handling is a way of handling error in a way by displaying error to normal user using human readable language. So, in the Micro pharmaceuticals I have handled the error using different error handling widgets. Exception means it is a problem that may arise during the execution of program. Exception Handling is the mechanism to handle runtime malfunctions. It can be achieved using four keywords: try, catch, finally, throw.

a. Try-Catch

"Try" and "catch" are keywords that represent the handling of exceptions due to data or coding errors during program execution. A try block is the block of code in which exceptions occur. A catch block catches and handles try block exceptions.

The try/catch statement is used in many programming languages, including C programming language (C++ and C#), Java, JavaScript and Structured Query Language (SQL). Try defines a block of statements that may throw an exception. When a specific type of exception occurs, a catch block catches the exception. If an exception is not handled by try/catch blocks, the

exception escalates through the call stack until the exception is caught or an error message is printed by the compiler.

Try catch syntax

```
try{
    //code that may throw exception
}catch(Exception_class_Name ref){}
```

Fig: Try-Catch Syntax

```
try {
    //codes
} catch (ExceptionType name) { }

catch (ExceptionType name) {
}
}
```

Each catch block is an exception handler that handles the type of exception indicated by its argument. The argument type, *ExceptionType*, declares the type of exception that the handler can handle and must be the name of a class that inherits from the *Throwable* class. The handler can refer to the exception with name.

The catch block contains code that is executed if and when the exception handler is invoked. The runtime system invokes the exception handler when the handler is the first one in the call stack whose *ExceptionType* matches the type of the exception thrown. The system considers it a match if the thrown object can legally be assigned to the exception handler's argument.

(thebopedia, anon, n.d.).

b. Use of Try-Catch in application

```
public static Connection getDBCon() {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/bsdb","root","");
        return con;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}
```

Fig: Use of Try-catch in application

Integrated Development Environment

An integrated development environment also known as IDE in short form is a software application that provides an environment for developers to develop, debug software programs. Instead of making every bit of functionality again and again to execute the developed programs, IDE provides all the tools that are required into a single application and workplace. According to **g2crowd (2012)**, Integrated Development Environments, or IDE, are software platforms that provide programmers and developers a comprehensive set of tools for software development in a single product. Each of the tools has an awareness of the environment, and they work together to present a seamless development set for the developer.

In order to implement the Java programming solution, I used Eclipse. This is a popular and one of the best IDE for developing web-based java application.

Eclipse

Eclipse is a free, Java-based development platform known for its plug-ins that allow developers to develop and test code written in other programming languages. Eclipse got its start in 2001 when IBM donated three million lines of code from its Java tools to develop an open source integrated development environment (IDE). The IDE was initially overseen by a consortium of software vendors seeking to create and foster a new community that would complement Apache's open source community. Rumor has it that the platform's name was derived from a secondary goal, which was to eclipse Microsoft's popular IDE, Visual Studio.

The Eclipse platform, when combined with the JDT, offers many of the features one would expect from a commercial quality IDE: a syntax-highlighting editor, incremental code compilation, a thread-aware source-level debugger, a class navigator, a



file/project manager, and interfaces to standard source control systems, such as CVS and ClearCase (tutorialspoint.com, anon, n.d).

Eclipse also includes a number of unique features such as code refactoring, automatic code updates/installs (via the Update Manager), a task list, support for unit testing with JUnit, and integration with the Jakarta Ant build tool. Despite the large number of standard features, Eclipse is different from traditional IDEs in a number of fundamental ways. Perhaps the most interesting feature of Eclipse is that it is completely platform- and language-neutral. In addition to the extensive mix of languages supported by the Eclipse Consortium (Java, C/C++, Cobol), there are also projects underway to add support for languages as diverse as Python, Eiffel, PHP, Ruby, and C# to Eclipse.

Advantages of Eclipse

- Code Completion, instead of digging through documentation you should be able to tab your way through methods and save yourself a lot of writing
- Refactoring, Global Find and Replace is no replacement for good refactoring support, that starts with renaming functions, variables, classes, ... and ends with some of the functionality that you can find in current Java IDE (IntelliJ for example)
- Syntax Checking, helping you out with writing correct code while you type

Use of Eclipse for the Departmental Store

Below are certain evidences that shows the uses of the Eclipse IDE for the implementation of the Java programming solution.

Index page

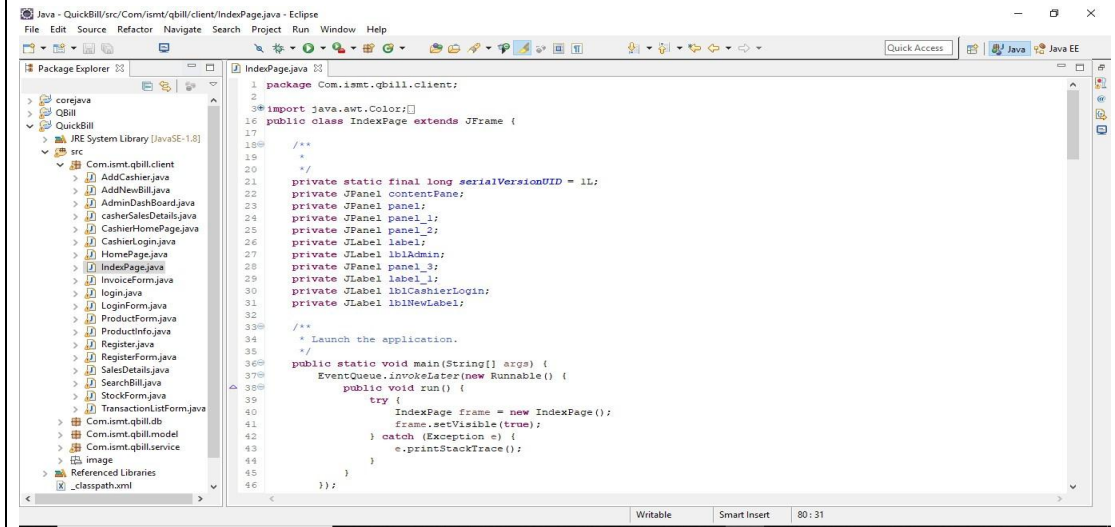
Welcome to QBill System



Admin Login



Cashier Login

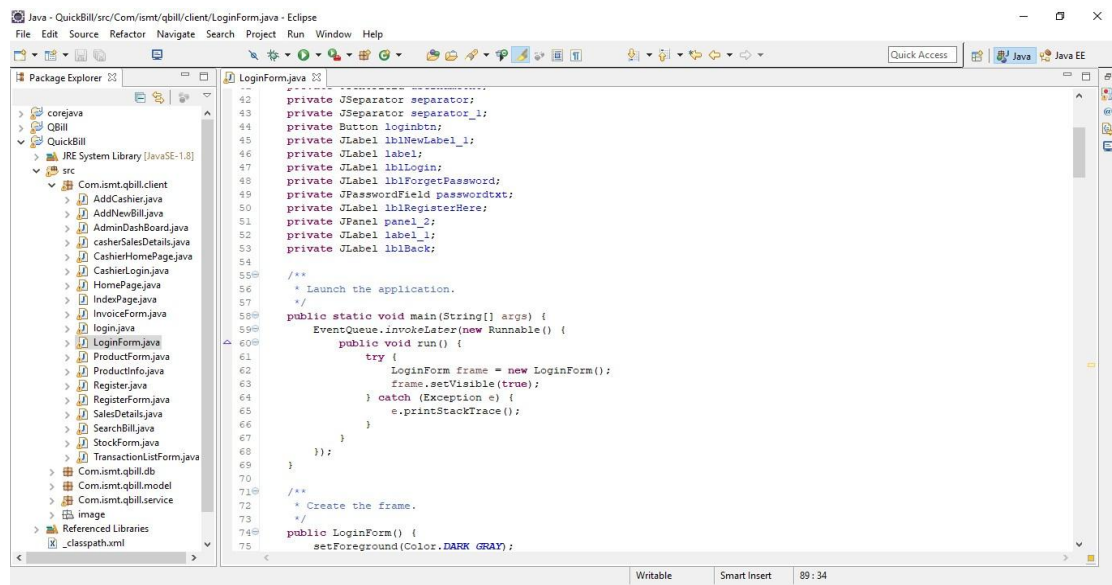


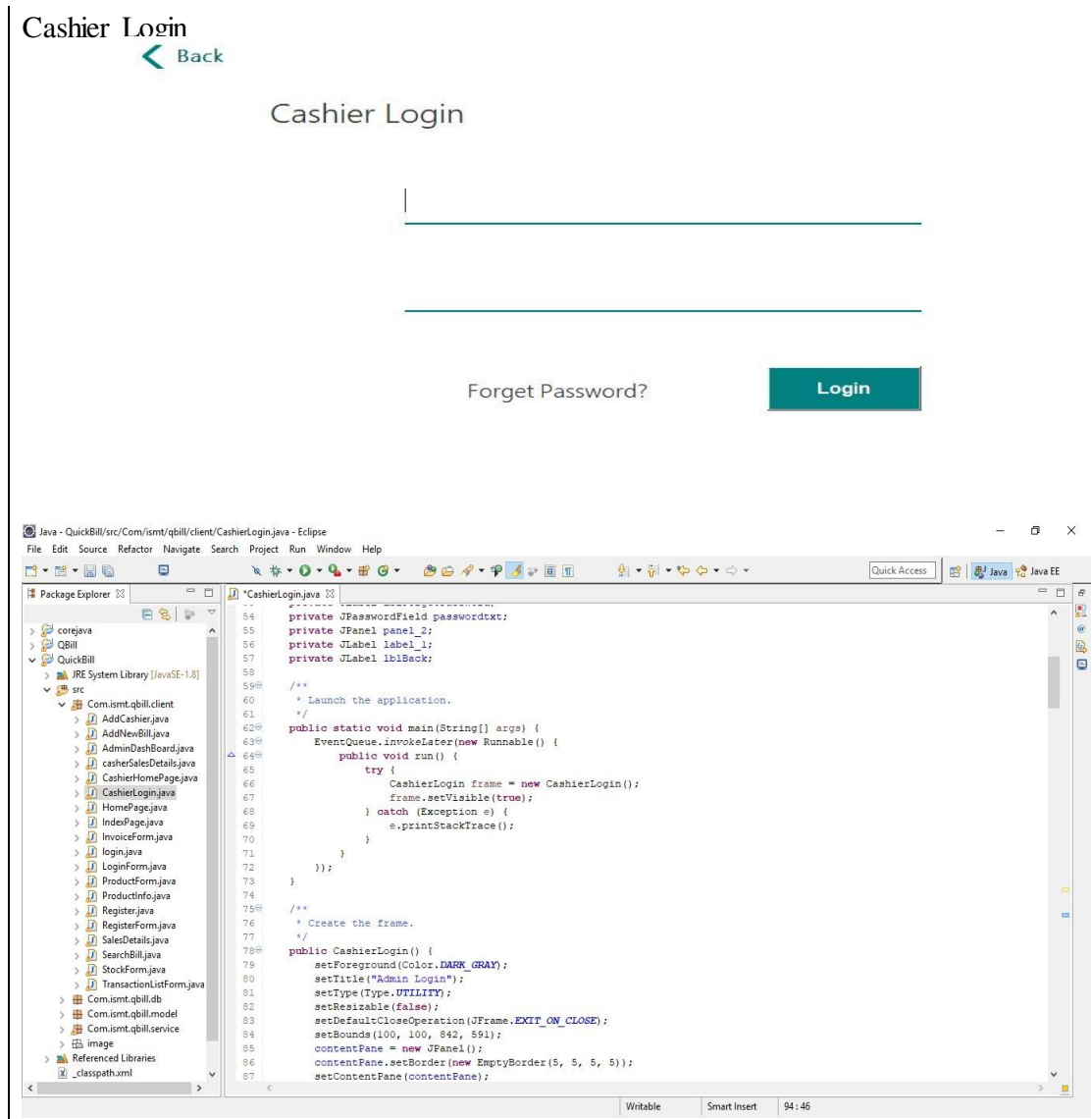
Admin Login

[Back](#)

Admin Login

Sanjil

[Forget Password?](#)[Login](#)[Register Here](#)



Database Management System

Database is the collection of related persistent data and contains information relevant to an enterprise. The database acts as a container for the collection of data files. The database selected for the Inventory Management System for the Departmental Store is SQLyog Community – MySQL GUI v12.2.0 (64 bit). SQLyog is one of the best database management systems. It is easy to use and backups are also easy and much faster. It is the best breed of all the SQL administrators out there says TJ Jones (**webyog, 2016**). It is the most complete and easy to use MySQL GUI. The schema and data changes can be done while moving from the test to production and also, the data replication can be scheduled using the superfast checksum-based algorithm. In this database management system, the streamline data transfer/migration from any

ODBC complaint can be managed to the data source to MySQL.

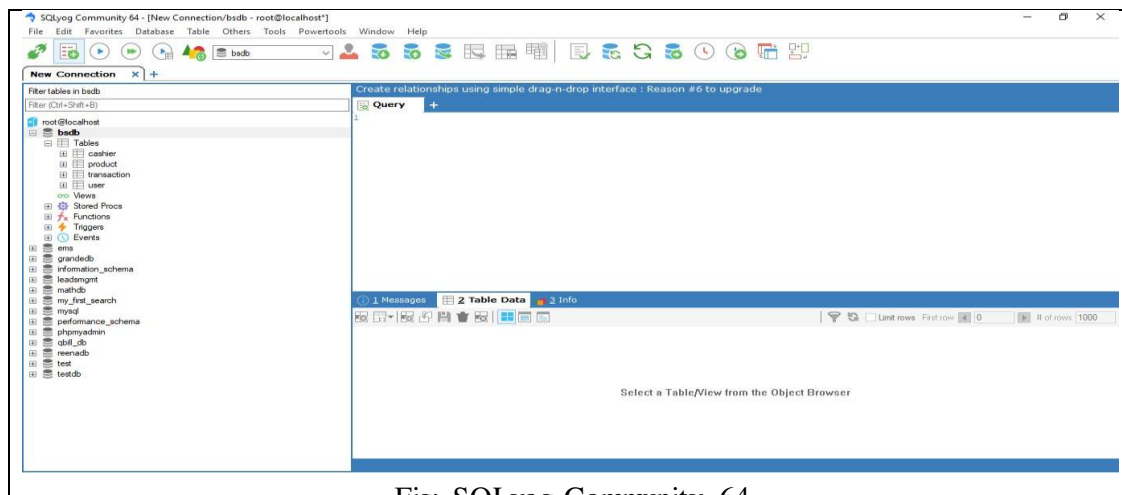


Fig: SQLyog Community 64

Conclusion

In this task, I have implemented and defined the Java programming solution based on a prepared design. As per the requirement, the solution has been developed. I have defined the relationship between objects to implement the design requirement. For the completion of the task, I have considered the terms such as control structures, design algorithms, error handling and reporting, Integrated Development Environment, codes and screen templates. All the required implementation and definition for the Departmental Store. Hence, I conclude my task with the assurance that I have done all the tasks based on the Inventory Management System from head to toe.

Task 6

Critically review and test the Java programming solution and analyze actual test results against expected results to identify discrepancies. [4.1, 4.2]

Introduction

Reviewing and testing are the activities that must be conducted after every completion of the any activity. In the case of Departmental Store, I had developed an inventory management system that manages and stores the records of the products, sales, budgets etc. The solution has been developed in the Java programming language on Eclipse IDE. As said, after completing the implementing phase of the solution, I will be reviewing and testing the system.

Review of the Java programming solution

The inventory management system for the Departmental Store was developed for different purposes in the store. The management wanted to increase the facilities and conditions of the product sales, billing and stock maintenance. The management wants the system to be built in such a way that could operate in no-lights condition, which means offline. As per the requirements, I had implemented the inventory management system with the features such as security, flawless and paperless operation. For the record management, I have used the free version database called SQLyog Community 64. The data can be stored in the database with vast storage space. The system is capable of calculating the stocks and also the billing system. With this system, I can assure that there will be very less flaws in the inventory management of the store. The system will provide the staffs with the benefits of no more paper works and hard writing and self-calculating. There will definitely be less errors in the store and also the frauds can be extremely decreased. The system will provide the invoice for the customer and view the details of the sales records as well. The user can search, create or even print the bills of the product sales. There are various sections in the system and various activity panels concentrating on various functional activities. Some of the sections are listed below:

- Index Page Activity
- Login Activity
- Register Cashier Activity
- Search Product Activity
- Sales bill Activity

Testing of the Java programming solution

Testing is the checking out how good the thing performs. It is the judging of the level of knowledge or skill has been acquired. In other words, testing is the key checkpoints in the overall process to identify whether the objectives are being met or not. As the task of developing the system is completed, testing is done to check whether the system works fine or not or the system is in best possible condition or not. “The process of testing an integrated system to verify that it meets specified requirements” (**Software Testing Fundamentals, n.d**). The testing is done to the system implemented.

Advantages of testing for Departmental Store

- To know about the software quality.
- For an effective performance of software application or product.
- To ensure that the application should not result into any failures.
- To ensure the quality of the product.
- To provide the facilities to the customers like the delivery of high-quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.

Test Log

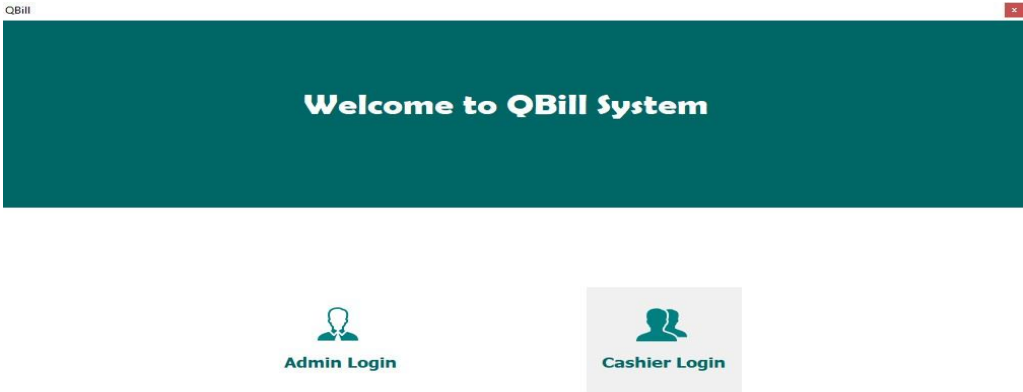
S.N.	Test Type	Test Title
1	Unit Test	<ul style="list-style-type: none"> • As I click exit button, the application must terminate.
2	Integrated Test	<ul style="list-style-type: none"> • If I try to login with empty credentials, then error message must pop up.
3	Compatibility Test	<ul style="list-style-type: none"> • The application must run on at least two operating system.
4	Stress Test	<ul style="list-style-type: none"> • If I try to login with an invalid credentials, then error message should appear.

Unit Testing

Unit testing, or developer testing, is the kind of test that focuses on the small pieces of the code, such as class, links, etc., that the developer writes. “Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation” (**techtarget, 2015**). Unit testing furthermore makes sense of whether the codes demonstration correctly or not as indicated by the anticipation.

Errors contained in various complex pages could be lessened and removed with the assistance of this testing.

The unit testing of the application is done below:

What was tested: Exit Button		Date :
S. N	Expected output	Actual output
1.	As I click exit button, the application must be terminated.	As expected the application terminated.
		
Analysis: Just as expected the application was terminated after clicking it. Hence it is successful test.		

Integrated Testing

Integrated testing is the kind of test in which the units of the program are joint and tested in several ways. A portion of the program is combined and placed in a group for testing purpose.

“Integration Testing is a level of software testing where individual units are combined and tested as a group” (Softwaretestingfundamentals, n.d).

The integrated testing for the application is done below:

What was tested: Login page validation		Date :
S. N	Expected output	Actual output
1.	If I try to login with empty credentials, then error message must pop up.	As expected the error message popped up.

The screenshot shows the 'Admin Login' page. At the top left is a 'Back' button. The title 'Admin Login' is centered. Below it are two horizontal lines for input fields. To the right, a 'Message' dialog box is displayed with the text 'Login Failed!' and an 'OK' button. Below the input fields, there is a 'Forget Password?' link and a green 'Login' button. At the bottom, there is a 'Register Here' link.

Analysis: Just as expected the error message popped up. Hence it is successful test.

Compatibility Testing

Compatibility testing is the kind of testing where the application is tested under various environments such as different operating systems, browsers, or even devices. It is a sort of nonfunctional testing. The compatibility testing for the system is done under two different operating systems (Windows 10 and Windows 8.1).

The compatibility test for the application is below:

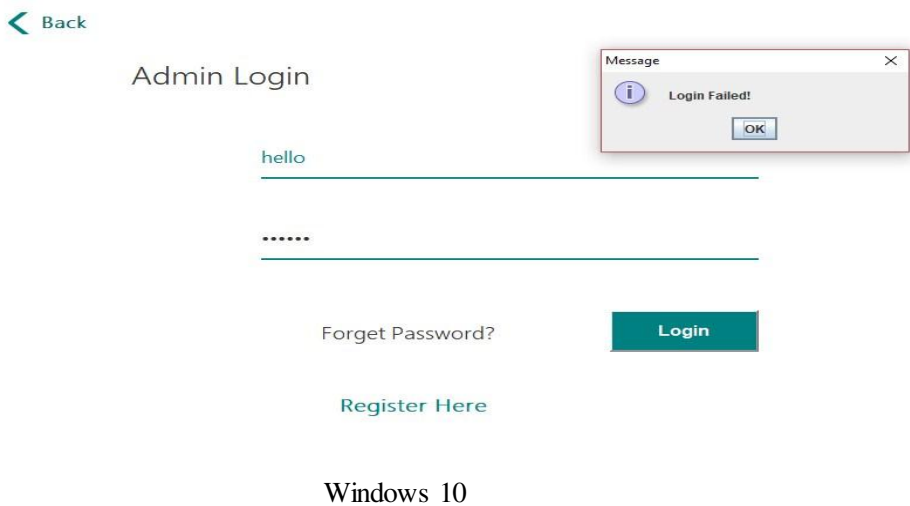
What was tested: Application Compatibility		Date:
S. N	Expected output	Actual output
1.	The application must run on at least two operating system.	As expected the application was running in both operating systems.
<p>Windows 10</p>		<p>Windows 8.1</p>

Analysis: Just as expected the application was running on both the operating system. However, the application was a bit slower in Windows 8.1 than in Windows 10. Hence it is successful test.

Stress Testing

Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. Stress testing is also known as endurance testing. Under Stress Testing, AUT is be stressed for a short period of time to know its withstanding capacity. Most prominent use of stress testing is to determine the limit, at which the system or software or hardware breaks. It also checks whether system demonstrates effective error management under extreme conditions.

The stress testing for the application is below:

What was tested: Login Page Validation		Date:
S. N	Expected output	Actual output
1.	If I try to login with an invalid credentials, then error message should appear.	As expected the error message appeared.
		
<p>Analysis: Just as expected the error message appeared. This is because the username hello is not registered in the database. Hence it is successful test.</p>		

Critical Review

Departmental store was the required of an inventory management system. I have been appointed as a system developer and designer. I was handed a responsibility of developing a system that

increases facilities and conditions of the product sales, billing and stock maintenance. Moreover, the management wanted the application to operate even in no-lights condition. Hence, I was required to present the management with the suitable design and solution for secure, smooth, flawless and paperless operation to cater the needs of sales person, stock manager using Java.

For the application, I used the Java programming language. Java is a computer programming language that enables programmers to write computer instructions using English based commands, instead of having to write in numeric codes. Java is a powerful object-oriented programming language. Java is backed up with the principles like inheritance, encapsulation, polymorphism and abstraction. Java programming language is highly secured and portable as well. The java programming language has the characteristics of robustness which refers to reliability. Considering the advantages and key features, I had chosen Java as a programming language to write the inventory management system for the Departmental Store. Similarly, for the IDE, I have chosen Eclipse. Eclipse is a free, Java-based development platform known for its plug-ins that allow developers to develop and test code written in other programming languages.

Before developing the java programming solution, I had been informed that there would be two categories of users; admin and cashier. According to the instructions, I followed the lead and developed the system. The program can be logged in with two categories of the users. One is admin users, which would be by default set as admin and other cashier users can only be registered by the admin users. The roles of the different categories of users are as follows:

1. Admin

- a. Enter information of all the products in the database.
- b. View details of products, sales records (transactions) and stock.
- c. Allow stocks monitoring frequently.
- d. Can create, delete, edit and view the cashier

2. Cashier

- a. Allow to get the product details to the sales department.
- b. Can create, search and print bill of the sales products.
- c. Generate invoice for the customer.
- d. View details of sales records.

The principle reason for the application is to mechanize the majority of the store financial sections diminishing human effort in writing and printing. The proposed application is all about

record management of the products manufactured in the store and saving their sales records accordingly. It is outlined with straight forward interfaces utilizing diverse instruments and procedures. A high-level Java programming has been utilized to implement the application for storing and retrieving the data in SQLyog database.

After the completion of the implementation, the system has gone through various testing. Testing is the phase where the system gets the assurance that it is error free. The testing included unit testing, integrated testing, compatibility testing and stress testing. After the successful testing, I can personally conclude that the system is working very well. They won't find any error while using the application. Testing the application has showed clearly that the application is not buggy and the can perform their work very efficiently here without any disturbance.

Conclusion

After the completion of implementation of the application, the testing phase has been conducted in this task. The Java programming solution developed for the Departmental store is working absolutely fine. The application is totally error free and is also compatible with the different operating systems such as windows 7, windows 8.1 and also windows 10. Unfortunately, the application was not developed regarding to the mac OS x. Hence, I conclude this task.

Task 7

Evaluate independent feedback on a developed Java programming solution and make recommendations for improvements. [4.3]

Introduction

The application development is a long process with the importance of various skills, creativity and good practice of using the different programming languages. The application for the Departmental store has been designed with the proper use of skills, creativity and different programming languages. As per the requirements of the management, the application has been developed. However, this application might contain error/s and on-to-be improved portions. For those errors and improvements, the evaluation of the application must be conducted. **UNODC (2017)** defines evaluation as a systematic and objective valuation of a current or completed project, program or plan, its design, operation and consequences. Through the process of evaluation, errors (if any) and proper judgements can be collected and valued.

In this task, the developed application of the inventory management system is evaluated in order to point out the possible errors and necessary improvements. As this is very responsive and critical task, some with good experience is only capable of doing this task. the evaluation procedure is carried out by Mr. Nikesh Tandukar. I had assigned Nikesh with the task of evaluation for the application and after completing the evaluation, he handed me the feedback which is included in this task.

Independent Feedback

Independent Feedback	
Project Title: Feedback of Inventory Management System	
Feedback given by: Beeky Tandukar	Signature Date:

Project Description (in brief):

As per the recommendation of Mr. Reeky Tandukar this application was provided for me to do further evaluation of the inventory management system developed by Reeky. Overviewing all the project and watching the application designed by Reeky. I can conclude that the Departmental store needs the system that can even operate in no-lights conditions. Furthermore, the store wants the system to increase the facilities and conditions of the product sales, billing and stock maintenance. The design is well managed and well structured. The application includes the modules such as users, product details, billing, invoice, etc.

Important features of the project

- The application is attractive, polished and looks professional.
- Each and every part of the application works perfectly.
- The application is GUI based and it has good user-friendly nature.
- All the forms are properly validated with the input information.
- The application is compatible with different versions of windows OS.
- It is easy to add, update, retrieve and delete the data.
- The search mechanism works perfectly.
- The bills can be presented into a printed form if wanted.
- All the product details can be viewed at once.
- Stock management is also made easy.

Checklists	YES	NO	Comment
Is the application compatible on different operating system?	√		Yes, the application is tested on various OS such as Windows 7, Windows 8.1, Windows 10 and it is compatible all these operating systems.
Is the navigation through the sections is easy?	√		Yes, the navigation is easy as all the sections can be accessed from the navigation menu on the home pages.
Are there any broken links in the application?		√	No, the application doesn't contain any broken link.
Are the included login forms properly validated?	√		Yes, all the login forms are properly validated. If wrong input is entered then the form won't be submitted.
Is there any onscreen help and user manual created to assist the user to use the web application?		√	No, there is no onscreen help but there is a user manual created in a document format.
Are the requirements fulfilled?	√		Yes, the application fulfils the requirements.
Is the application up-to-date and accurate concerning the contents?	√		Yes, the application is updated with the contents of the store.

Significance of the Project	<ul style="list-style-type: none"> • Different types of users can login. • Quick data searching. • Uses Java programming language. • Different policies for different users.
Project Limitation	<ul style="list-style-type: none"> • The application does not operate in Mac OS. • The user cannot change their password. • There is no onscreen help.

Feedback Evaluation**Feedback Evaluated by:**

Reeky Tandukar

Signature:**Date:****Evaluation of the Feedback:**

Feedback is given to address the information or criticism about the behaviors or characteristics of something from an individual, groups or experts. With the response to the feedback, there can take place an improvement and betterment functioning. The feedback given by Mr. Beeky Tandukar shows both the positive and negative components of the application of the Departmental store. According to Mr. Beeky Tandukar, the application is very suitable for the Departmental store. The thing that he found significant in the application is that the application is developed under Java programming language. He also added that the application fulfills all the requirements given by the management. The different users can login to the application and there are different policies for different users. The application uses the suitable database as well. SQLyog is one of the most popular databases in the database market. The application has undergone into various levels of testing and after passing all the tests, the application is all set to be handed to the Departmental Store.

There are many pros and certain limitations of the application developed. The system is designed in Java programming language, which should be frequently updated as more current versions are propelled in the market. The hierarchy of the content and menu are perfectly clear and are easy to use and simple.

Recommendation for future consideration

- The application must be updated time to time.
- Provide a FAQ page.
- If possible then change the color combination.
- Provide onscreen help.

Conclusion

In this task, the independent feedback for the application designed for Departmental Store has been provided. This document includes an effective independent feedback form and the evaluation and recommendations for the developed Java programming solution. Mr. Beeky Tandukar provided me the well evaluated feedback. His honest feedback helped me to know more about the system I had developed. I could be acknowledged with the strengths and weaknesses of the application after studying and analysing the solution. I further made the evaluation about the application and also included certain recommendations for the betterment and improvement of the Java programming solution.

Task 8

Create user documentation for the developed Java program solution. **[4.4]**

Introduction

This is the task where I will be creating a documentation for the developed Java programming solution. After the complete development of the Java programming solution, there is the responsibility of providing a complete guide that supports the solution. “A user guide is essentially a book-length document containing instructions on installing, using, or troubleshooting a hardware or software product” (McMurrey, n.d). User manual is the guide that gives the instructions and shows the procedure of the using certain software, website or any system. It helps to understand the product in details. It guides the methods of using the product efficiently and smoothly.

Similarly, in this task, I will be creating a user manual guide for the inventory management system for the Departmental Store.

User Documentation

User Manual for QBill

This is the user manual that will help you to use the application with the ease. This guide is specially based for those users who are new to using the application platforms for the record management and face problems while collaborating with this application. Considering the users with least knowledge, this user manual has been created.

Contents

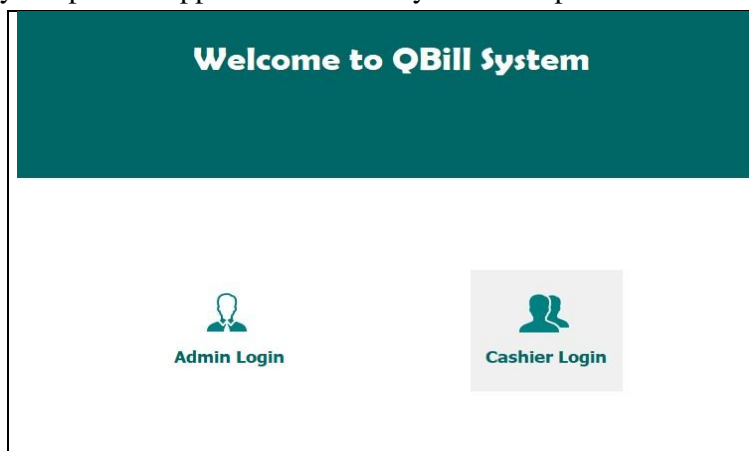
1. Introduction
2. Logging into QBill as an admin
3. Logging into QBill as a cashier
4. Product Details
5. Product Search
6. Invoice form
7. Stock Details
8. Transaction list
9. Add New Bill
10. Search Bill

11. Exit

Introduction

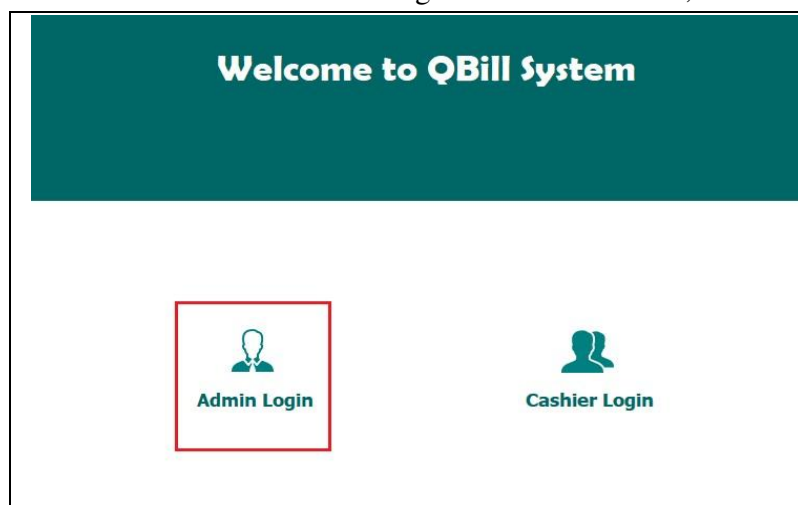
Departmental Store is the convenient store that has the transactions of varieties of electronic gadgets. With the advancement of the technologies and the backwardness of the record management system in the store, the management has decided to implement the inventory management system. This java programming solution for the inventory management system increases facilities and conditions of the product sales, billing and stock maintenance. This system even operates in no-lights condition.

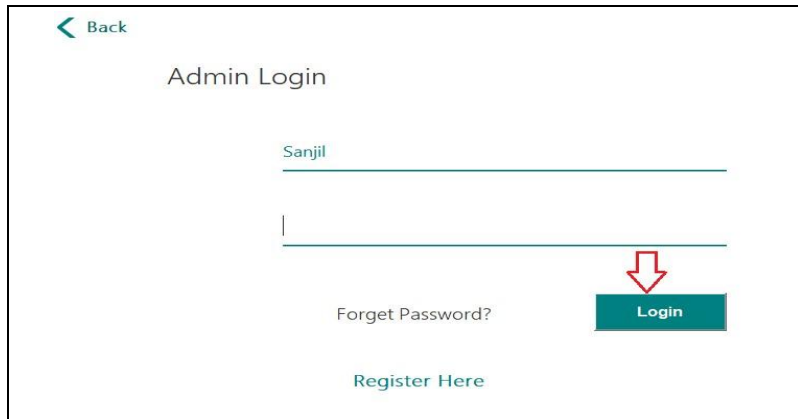
When you open the application the first you come up with is the index page.



2. Logging into QBill as an admin

When you are logging in as an admin then you should select the Admin Login option and enter the credentials of the admin user. After entering the valid credentials, click on **Login** button.





Admin Login

Sanjil

Forget Password?

Register Here

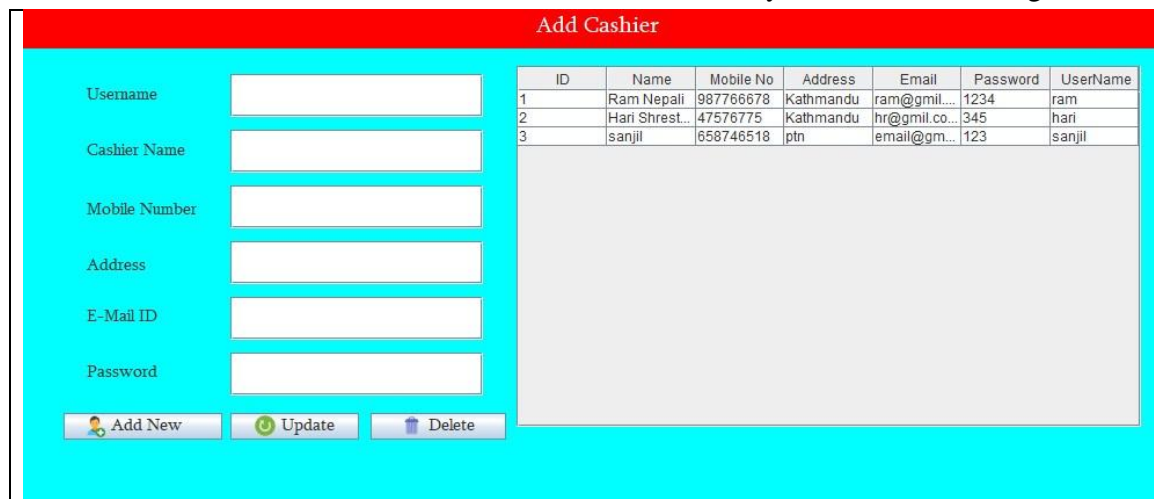
Login

Then you will be leaded to the Admin Dashboard.



• Cashier Operations

In the Admin Dashboard, click on the **Cashier** section. Then you will see something like this:



Add Cashier

Username:

Cashier Name:

Mobile Number:

Address:

E-Mail ID:

Password:

ID	Name	Mobile No	Address	Email	Password	UserName
1	Ram Nepali	987766678	Kathmandu	ram@gmil...	1234	ram
2	Hari Shrest...	47576775	Kathmandu	hr@gmil.co...	345	hari
3	sanjil	658746518	ptn	email@gm...	123	sanjil

o Add Cashier

In order to add cashier, you have to fill all the fields such as Username, Cashier Name, Mobile Number, Address, E-Mail ID and Password and then click on **Add New** button.

Add Cashier

Username

Cashier Name

Mobile Number

Address

E-Mail ID

Password

ID	Name	Mobile No	Address	Email	Password	UserName
1	Ram Nepali	987766678	Kathmandu	ram@gmil....	1234	ram
2	Hari Shrest...	47576775	Kathmandu	hr@gmil.co...	345	hari
3	sanjil	658746518	ptn	email@gm...	123	sanjil

➡ Add New
 Update
 Delete

o Update Cashier

In order to Update the Cashier, you have to select the cashier on the right-side panel and the edit the information and then click on **Update** button.

Add Cashier

Username

Cashier Name

Mobile Number

Address

E-Mail ID

Password

ID	Name	Mobile No	Address	Email	Password	UserName
1	Ram Nepali	987766678	Kathmandu	ram@gmil....	1234	ram
2	Hari Shrest...	47576775	Kathmandu	hr@gmil.co...	345	hari
3	sanjil	658746518	ptn	email@gm...	123	sanjil
4	New	1234567890	New	tobeupdate...	Deletelt	New

Add New
 Update
 Delete

o Delete Cashier

In order to delete the cashier, you have to select the cashier and then click on the **Delete** button.

Add Cashier

Username:

Cashier Name:

Mobile Number:

Address:

E-Mail ID:

Password:

ID	Name	Mobile No	Address	Email	Password	UserName
1	Ram Nepali	987766678	Kathmandu	ram@gmail....	1234	ram
2	Hari Shrest..	47576775	Kathmandu	hr@gmail.co...	345	hari
3	sanjil	658746518	ptn	email@gm...	123	sanjil
4	Updated N...	1234567890	Updated	alreadyupd...	Deletelt	Updated

Add New
 Update
 Delete

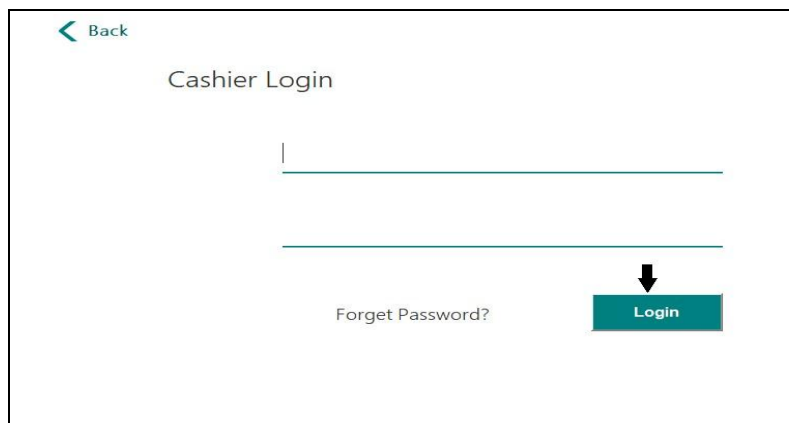
3. Logging into QBill as a cashier

When you are logging in as an admin then you should select the Admin Login option and enter the credentials of the admin user. After entering the valid credentials, click on **Login** button.

Welcome to QBill System

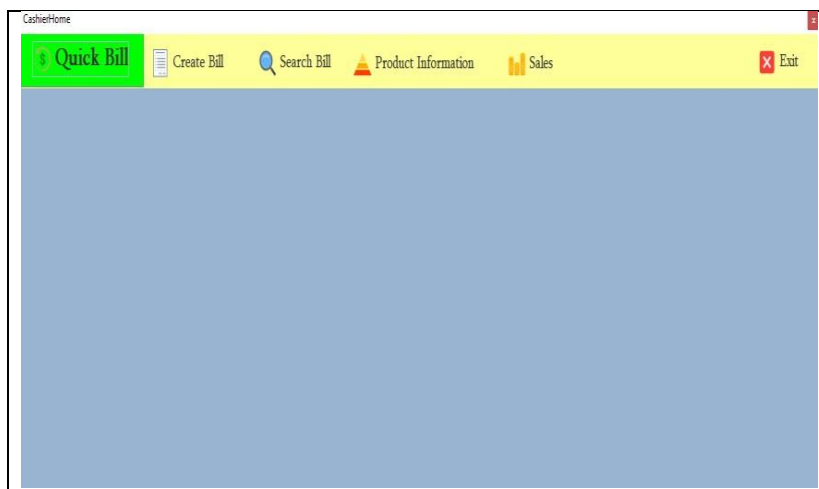
Admin Login

Cashier Login



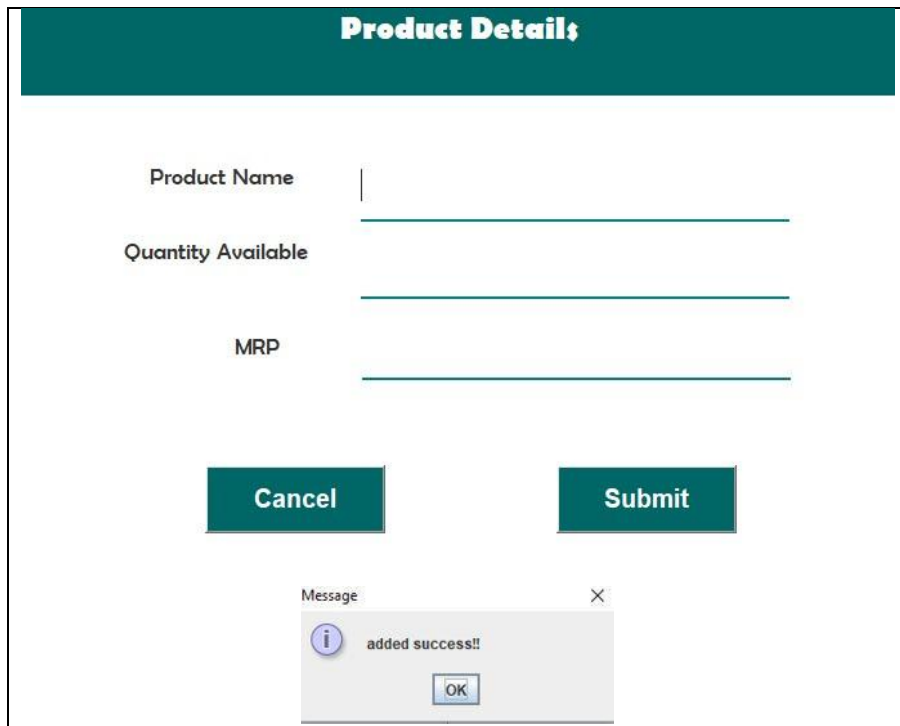
The image shows a 'Cashier Login' form. At the top left is a '< Back' link. The title 'Cashier Login' is centered. Below the title are two input fields for username and password. Below the password field is a 'Login' button with a downward arrow icon. To the left of the button is a 'Forgot Password?' link.

Then you will be led to the Cashier Dashboard.



4. Product Details

Adding the product details are easy. You are just required to fill up the fields with the correct information and click on **Submit** button. Then you will the message of success.



The image shows a web form titled "Product Detail:" in a dark green header. The form contains three input fields: "Product Name", "Quantity Available", and "MRP". Below these fields are two buttons: "Cancel" and "Submit". At the bottom, a small message box is displayed with the text "added success!!" and an "OK" button.

5. Product Search

Searching for the product, you have to enter the name of the product and click on **Search** button then you will see the results.



The image shows a web form titled "Product Information" in a green header. The form has a "Product" label next to a text input field. To the right of the input field is a "Search" button with a magnifying glass icon. Below the input field is a table with four columns: "Product ID", "Product Name", "Available", and "Mrp:". The table body is currently empty.

6. Invoice form

The invoice form can be filled up by filling up all the information and clicking on **Submit**

button.

Customer Name

Date

Invoice No.

Units Left:

Select Item

---- select items ----

Price
Units
Total

CustomerName	Item	Unit	TotalPrice
Roman Poudel	Dell Laptop	30	2100000.0
Hari	Printer	11	88000.0
Gopal	Samsung TV	6	180000.0
Gopal	Washing Machine	8	480000.0
Ram	Vaccum Cleaner	2	40000.0
Alisa	Samsung TV	10	300000.0
Kishor	AC	5	400000.0
pawan	Washing Machine	5	300000.0
Ruesh	acer Laptop	30	1500000.0

7. Stock Details

This section shows the number of products remaining in the store. The new products can be added by clicking on **AddNew Product** button.

AddNew Product

Stock Details

Product ID	Product Name	Total Quantity	AvailableQuantity	Mrp
1	Dell Laptop	200	190	70000.0
2	acer Laptop	100	70	50000.0
3	AC	30	30	80000.0
4	Vaccum Cleaner	60	60	20000.0
5	Printer	90	90	8000.0
6	Samsung TV	120	50	30000.0
7	Washing Machine	75	70	60000.0
8	Mouse	60	60	350.0
9	NeoteBook	50	50	25000.0

8. Transaction list

This section shows the number of transactions that took place up till the current date. You can print the section by clicking on **Print** button.

Transaction Details					
CustomerName	ProductName	Unit	TotalPrice	Date	
Roman Poudel	Dell Laptop	30	2100000.0	2018-08-01	
Hari	Printer	11	88000.0	2018-08-08	
Gopal	Samsung TV	6	180000.0	2018-08-08	
Gopal	Washing Machine	8	480000.0	2018-08-08	
Ram	Vaccum Cleaner	2	40000.0	2018-08-30	
Alisa	Samsung TV	10	300000.0	2018-08-02	
Kishor	AC	5	400000.0	2018-08-16	
pawan	Washing Machine	5	300000.0	2018-08-22	
Ruesh	acer Laptop	30	1500000.0	2018-08-09	
sarina	Dell Laptop	10	700000.0	2018-09-11	

Print

9. Search Bill

This is the section where the user can search for the bill using the bill no. and clicking on **Search**.

Search Bill					
Bill No:	<input type="text"/>	<input type="button" value="Search"/>			
BillNo:	Customer Name	ProductName	TotalAmount	Date	

11. Exit

If you want to exit the application, then click on **Exit**.



Conclusion

In this task, I created a user documentation to help end users to use this application. Using this application is not so tough. However, I have created this documentation, in case they need any kind of help while using the Inventory Management System. Hence, I presented every single screenshot with steps to guide the users of Inventory Management system application.

Task 9

Create technical documentation for the support and maintenance of a Java program solution [4.5, M3]

Introduction

I have successfully developed the Inventory Management System with every functionality required by the Departmental Store. In this task, I will be creating a technical documentation for the support and maintenance of a Java programming solution. I will be considering the designing layout, choice of tools, task breakdown approach, data and information collection mechanism, product deployment and maintenance technique for the project.

Scenario

The Departmental Store is one of the most reliable and convenient stores in the locality. It deals with a broad range of electronic gadgets such as computers, mobiles, printers etc. The management of the Departmental Store has appointed me as a system developer and designer to increase facilities and conditions of product sales, billing and stocks maintenance. The application is to facilitate the store by managing their inventories.

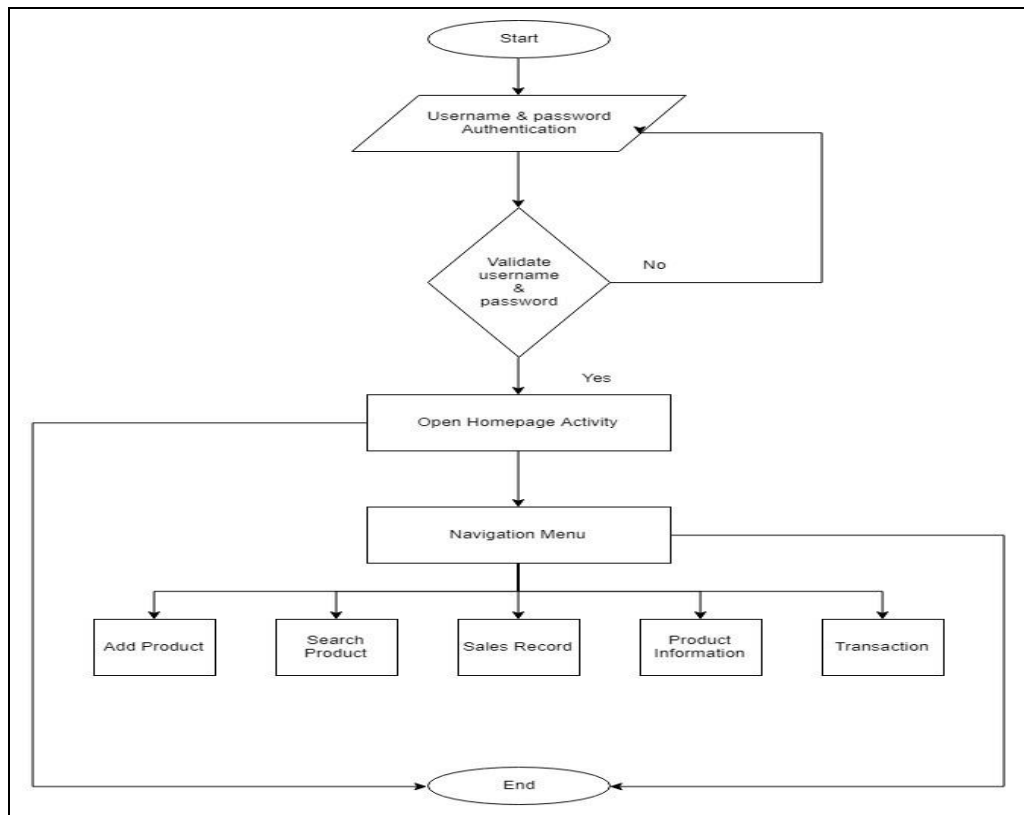
As I am appointed as a system developer and designer to develop an Inventory Management System, my responsibility is to develop an application that is suitable for the store considering the CRUD operations with a suitable design and solution, secure, smooth, flawless and paperless operation to cater needs of sales person, stock manager using reliable tool, programming language, database. The application should work in no-lights condition as well.

Technical Documentation

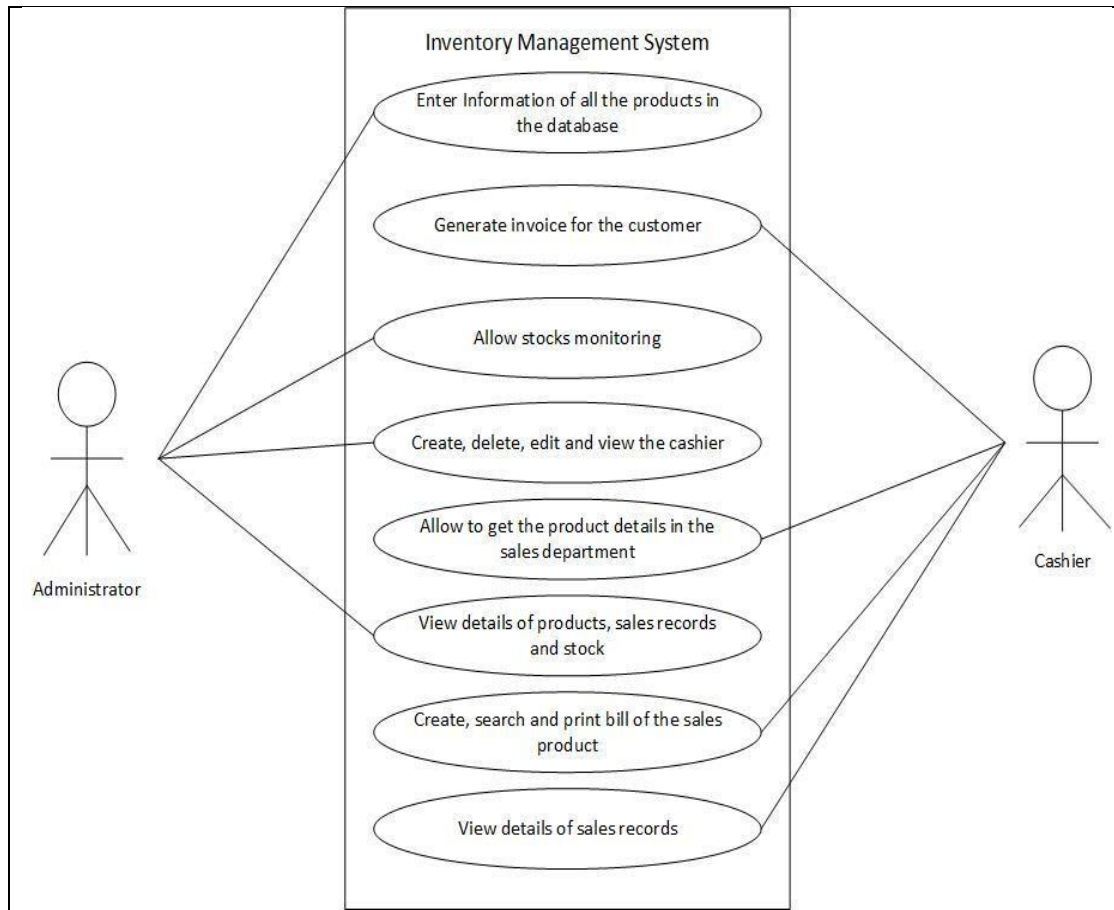
Technical documentation is the way of creating a documentation about a developed software application in a technical way. Technical documentation is the general term for documentation with regard to a product. People mainly associate the term with the documents and information that are passed on to the public by the manufacturer. The technical documentation consists of the tools, language, IDE, deployment, product specification, maintenance technique and many more. Technical documentation of Inventory Management System also consists of everything that are required for future references to new programmer or for the company itself.

System Design

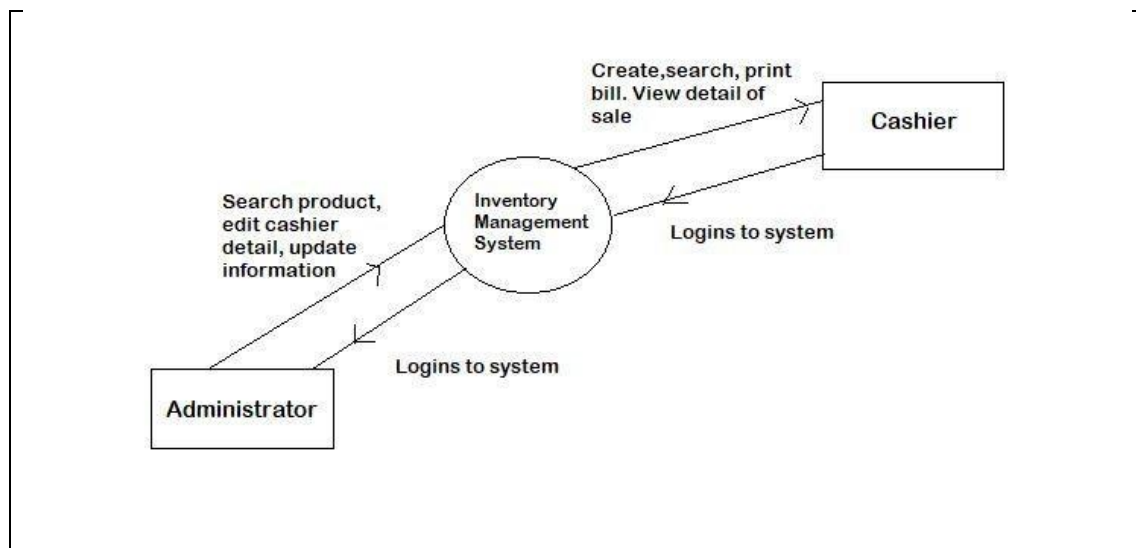
For the system design, I used different modules such as flowchart, context diagram and Data flow diagram.



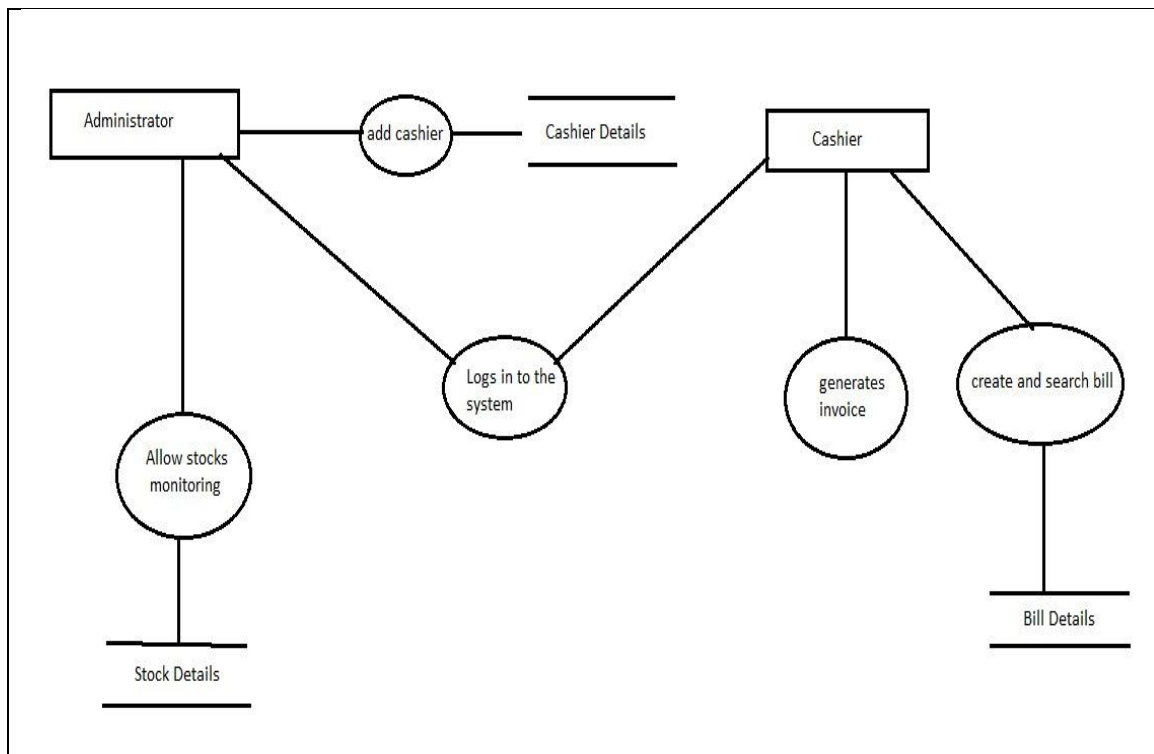
Use case diagram



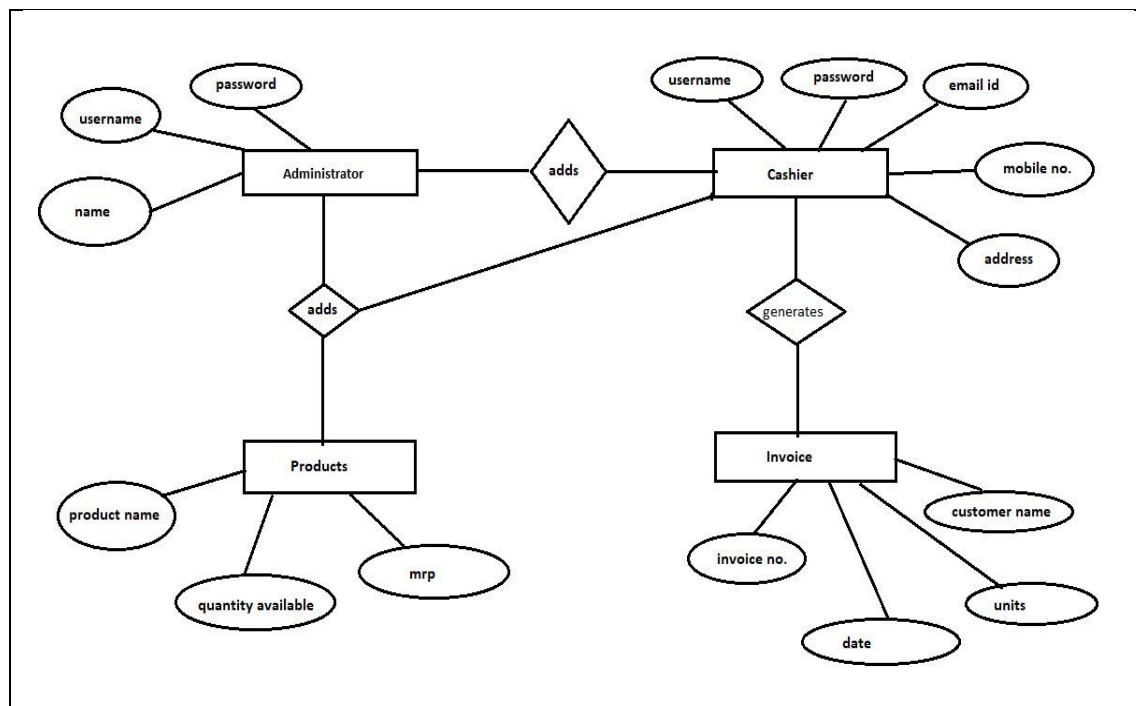
Context Diagram



Data flow Diagram



ER Diagram



Choice of Tools

For the development of the Inventory Management System, various tools have been used such as Java as a programming language, Eclipse as an IDE, SQLyog as a database and XAMPP as an Apache Web server.

a. Java Programming Language

Java is a computer programming language. It enables programmers to write computer instructions using English based commands, instead of having to write in numeric codes. From mobile phones to handheld devices, games and navigation systems to e-business solutions, Java is everywhere (**Oracle.com, 2014**). Once a program has been written, the high-level instructions are translated into numeric codes that computers can understand and execute.

Java is a simple and yet powerful object-oriented programming language and it is in many respects similar to C++. Java originated at Sun Microsystems, Inc. in 1991. It was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. It was developed to provide a platform-independent programming language. Java was designed with a concept of 'write once and run everywhere'. It is a software that is implemented on top of real hardware and operating system. When the source code (.java files) is compiled, it is translated into byte codes and then placed into (.class) files. The JVM executes these bytecodes. So, Java byte codes can be thought of as the machine language of the JVM.

JAVA programming language follows the four principles of OOPs; they are inheritance, polymorphism and encapsulation. All the concepts of JAVA are based on classes and its objects. An object is a real-world entity that has some attributes or properties.

The advantages of Java are as follows:

- Java is easy to learn.
- Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
- Java is object-oriented. This allows you to create modular programs and reusable code.
- Java is platform-independent. One of the most significant advantages of Java is its ability to move easily from one computer system to another.

b. Eclipse

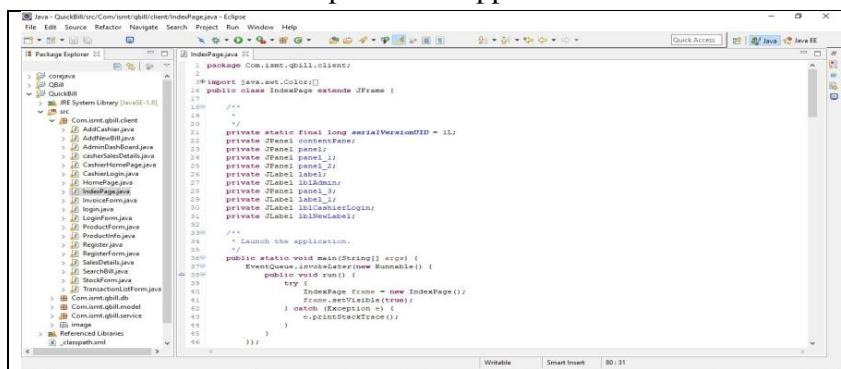
Eclipse is a free, Java-based development platform known for its plug-ins that allow developers to develop and test code written in other programming languages. Eclipse got its start in 2001 when IBM donated three million lines of code from its Java tools to develop an open source integrated development environment (IDE). The IDE was initially overseen by a consortium of software vendors seeking to create and foster a new community that would complement Apache's open source community. Rumor has it that the platform's name was derived from a secondary goal, which was to eclipse Microsoft's popular IDE, Visual Studio.

The Eclipse platform, when combined with the JDT, offers many of the features one would expect from a commercial-quality IDE: a syntax-highlighting editor, incremental code compilation, a thread-aware source-level debugger, a class navigator, a file/project manager, and interfaces to standard source control systems, such as CVS and ClearCase (tutorialspoint.com, anon, n.d).



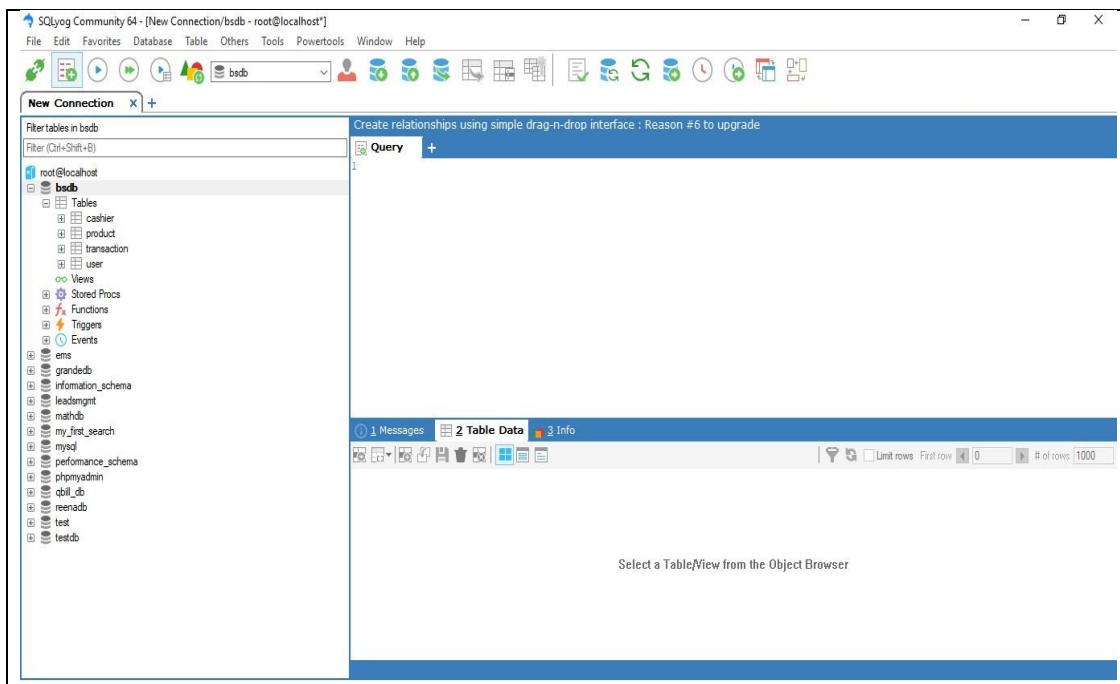
Eclipse also includes a number of unique features such as code refactoring, automatic code updates/installs (via the Update Manager), a task list, support for unit testing with JUnit, and integration with the Jakarta Ant build tool. Despite the large number of standard features, Eclipse is different from traditional IDEs in a number of fundamental ways. Perhaps the most interesting feature of Eclipse is that it is completely platform- and language-neutral. In addition to the extensive mix of languages supported by the Eclipse Consortium (Java, C/C++, Cobol), there are also projects underway to add support for languages as diverse as Python, Eiffel, PHP, Ruby, and C# to Eclipse.

The image below shows the use of Eclipse in the application.



c. SQLyog

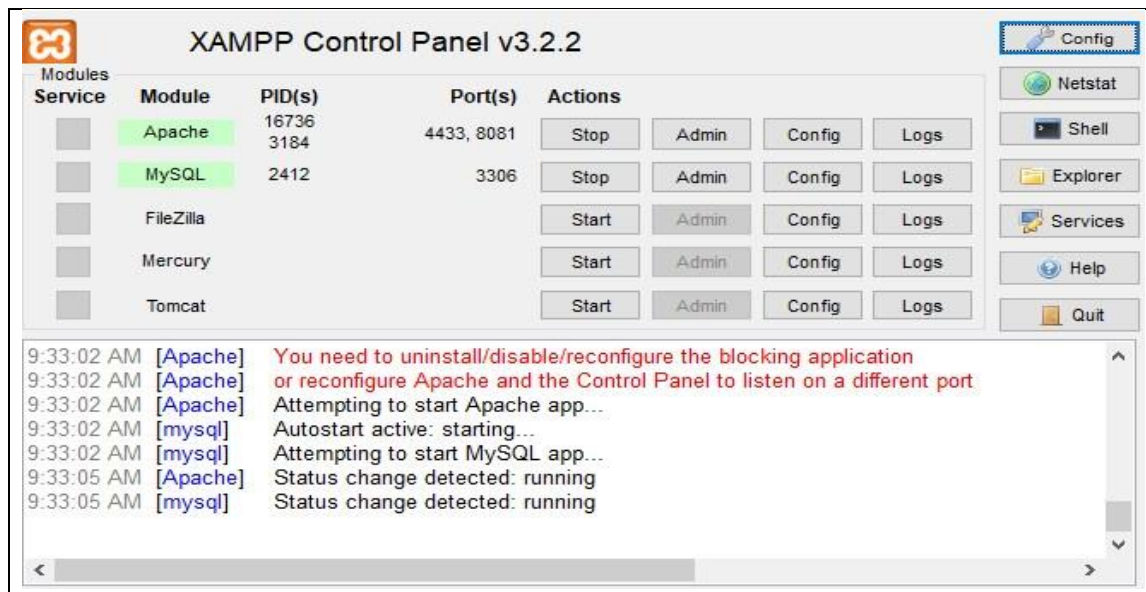
SQLyog is one of the best database management systems. It is easy to use and backups are also easy and much faster. It is the best breed of all the SQL administrators out there says TJ Jones (webyog, 2016). It is the most complete and easy to use MySQL GUI. The schema and data changes can be done while moving from the test to production and also, the data replication can be scheduled using the superfast checksum-based algorithm. In this database management system, the streamline data transfer/migration from any ODBC complaint can be managed to the data source to MySQL.



d. XAMPP

XAMPP stands for Cross-platform (X), Apache Server (A), MySQL (M), PHP application programming language (P) and Perl Application programming language (P). XAMPP is an open source free software developed by Apache friends. XAMPP software package contains Apache distributions for Apache server, MariaDB, PHP, and Perl. And it is basically a local host or a local server. This local server works on your own desktop or laptop computer. You can just install this software on your laptop or desktop and test the clients or your website before uploading it to the remote web server or computer. This XAMPP server software gives you suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer.

XAMPP contains tools such as Apache, MYSQL, PHP, and Perl.



Task Breakdown Approach

Task breakdown is the method that arranges a task into manageable segments. It helps to manage the time specifications and time availability denoted for the completion of the task in a certain time period. Task breakdown can be shown in various methods. Here, I have used a Gantt Chart to illustrate the time period for the development of the application. Being a programmer, I have the responsibility to plan, design and implement the application on time with high level of perfection. In order to manage time and schedule, I used Gantt Chart so that I can get motivated and gain focus on the work reviewing the timeline and pending tasks. Gantt chart helps in avoiding the confusions and distraction while the performing tasks. The Gantt Chart is the visual timeline that shows the starting to ending of the project.

The Gantt Chart prepared for developing the web application for Inventory Management System is mentioned below:

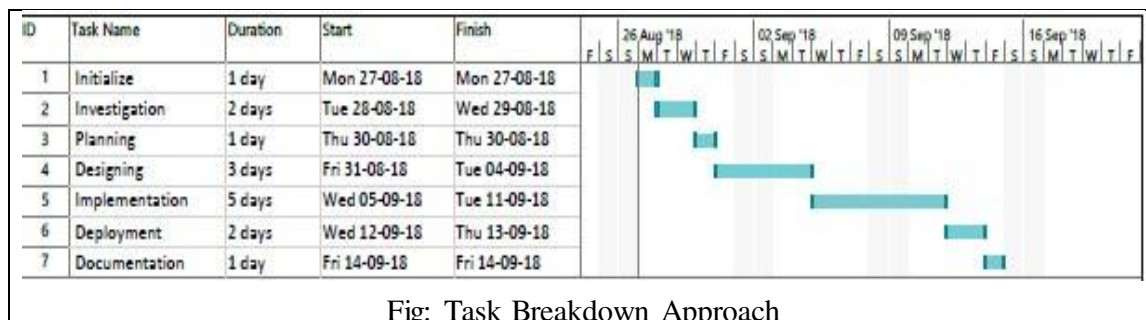


Fig: Task Breakdown Approach

Data and information collection mechanism

Data collection process will include attention to all the elements of the logic model: what resources had available, what activities actually provided, how many of each output delivered, and to what degree outcomes are accomplished. In collecting indicator data, developers are likely to use one or more of these four methods: surveys, interviews or focus groups, observations, and record or document review. In selecting the best method for data collection, one will need to consider the type of information need; the method's validity and reliability; the resources that are available, such as staff, time, and money; and cultural appropriateness, or how well the method fits the language, norms, and values of the individuals and groups from whom the data are being collected.

Features of Data and Information Collection Mechanism

- Collect data offline without being connected to the Internet.
- Design mobile survey with the drag and drop editor.
- No programmer required.
- Capture data on smartphones, iPads and Tablets.
- Easily manage one or hundreds of devices and field workers.
- Receive email notifications when something needs your immediate attention.
- Conduct mobile surveys in English and Nepali languages.
- Upload survey results to the Cloud or directly to own server.
- Benefit from higher response rates and faster results.

Data Collection Mechanisms

The various data mechanisms are discussed below:

a. Interview

This mechanism is applied to gather the data from groups or people. The analyst chooses the public who relate to the system for the interview. In this technique the expert sits up close and personal with the public and records their responses. The interviewer must plan about the kind of inquiries he/she will ask and ought to be prepared to answer any sort of question. He should also choose reasonable place and time which will be comfortable for the respondent.

b. Observations
Observation is another data collection mechanism that helps to get good positive response. In

this technique, the analyst himself visits the association and watches and comprehend the stream of records, working of the current framework, the clients of the framework and so forth. For this strategy to be received it takes an analyst to play out this activity as he most likely is aware which focuses ought to be seen and featured. In analyst may watch the undesirable things also and essentially cause delay in the improvement of the new framework.

c. Questionnaires

It is the mechanism used to extract information from number of individuals. This technique can be received and utilized only by the analyst. The Questionnaire comprises of arrangement of questions confined together in sensible way. The questions are basic, clear and to the point. This technique is exceptionally helpful for achieving information from individuals who are worried about the utilization of the framework and who are living in various nations. The questionnaire can be sent or send to individuals by post.

d. Investigation of documentation

In this mechanism, the document created for the system is checked for any possible defects. The analyst investigates whether the document is created in a correct way or not. In the Departmental store, the documents have been investigated and evaluated. During the developmental phase of the system, if the proper document is not prepared then the chance of facing difficulties in the future is relatively sure. The documents for the system of Departmental store is also investigated and evaluated so that the system works well in the future.

Product Deployment

Product deployment is the process of getting the program ready for market. A newly created program may work fine on the local computer, but that does not mean it is really ready for others to use. It is the process of giving an application the appropriate market and the use. After deployment, the application can be officially launched. Since we have used Eclipse as an IDE, there is a feature of export in it. We can export the application from the IDE itself.

Before deploying the application, there are requirements to be considered for any computer for it to run the application. The requirements are as follows:

- RAM – 1 GB or higher
- Hard Disk space – 2 GB

- Processor – Dual core process or higher
- Java – Java SE 8 (LTS) or higher
- Operating System – Windows 7, Windows 8.1 or Windows 10

Maintenance Techniques

Application development efforts result in the delivery of an application that satisfies company requirements. Accordingly, the application must change or evolve. Once in operation, defects are uncovered, operating environments change, and new user requirements surface. Application maintenance is an integral part of a software life cycle. However, it has not received the same degree of attention that the other phases have. Historically, application development has had a much higher profile than application maintenance in most organizations. This is now changing, as organizations strive to squeeze the most out of their application development investment by keeping software operating as long as possible. The open source paradigm has brought further attention to the issue of maintaining application artifacts developed by others.

Some of the maintenance techniques are as follows:

- **Corrective maintenance:** Reactive modification of a software product performed after delivery to correct discovered problems. It deals with fixing bugs in the code.
- **Adaptive maintenance:** Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment. It deals with adapting the software to new environments.
- **Perfective maintenance:** Modification of a software product after delivery to improve performance or maintainability. It deals with updating the software according to changes in user requirements.
- **Preventive maintenance:** Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults. It deals with updating documentation and making the software more maintainable (**Rice University, 2015**).

Maintenance for Inventory Management System

- Regular back up of the data and application is must for future reference.
- Frequent update should be done to maintain the security according to the current date.
- Many bugs can be seen after the deployment of the application. So, these bugs should be

fixed in a regular basis by providing updates.

- Proper backup must be done in the case of unexpected crashes or dumps.

Conclusion

Technical documentation has been successfully created in this task following feasible approach. The designing of layout, choice of tools, task breakdown approach, data and information collection mechanism, product deployment and maintenance technique for the project has all been documented in this report. This report can be used at any time as references while maintaining the Inventory Management System.