**TRAINING OBJECTIVE**

The main objective of the Industrial Training is learn new technology so that we are able work in the organization and competes with the world. Understand and learning new technology in **a Computer Science** is very important. A good training could be really beneficial for us in the future.

Also getting a industrial training from a recognized and experience company help us to understand real life situations in industrial organizations and their related environments and accelerating the learning process of how student’s knowledge could be used in a realistic way.

It provides the exposure to practice and apply the acquired knowledge “hands - on” in the working environment. Industrial training also provides a systematic introduction to the ways of industry and developing talent and attitudes, so that one can understand how Human Resource Development works. Moreover, students can gain hands-on experience that is related to the students majoring so that the student can relate to and widen the skills that have been learnt while being in university. Industrial training also exposes the students to the real career world. Furthermore, students implement what they have learned and learn more throughout this training. Besides, students can also gain experience to select the optimal solution in handling a situation. During industrial training students can learn the accepted safety practices in the industry.

**BRIEF OF ORGANIZATION**

Tech Hive is a Multi Domain Organization, covering almost all major trends of modern day technology. From the day of our establishment, we have been constantly widening our horizons and spreading out our feathers to meet the ever increasing demands of our clients. We deal basically internationally and nationally according to demand of our clients. We feel immense honor to introduce ourselves as one of the leading- Embedded , Industrial Automation , Smart Software based Application and Product Development Company. Tech Hive is a web development company and consulting firm specializing in business & management consulting, IT solutions, web site design & development and Software development. Our teams of consultants are able to provide complete business solutions to clients locally as well as internationally.

We also specialize in ERP systems and custom software development designed according to your business specifications. Whatever your needs, whether you are a small business or a large corporation, our Web Design company will provide you with solutions and designs to suit your budget and requirements.Our highly qualified team of consultants have over five years’ experience in providing quality consulting & guidance as well as offering our expertise in software, design and development. Our developers are qualified to work on C/C++, ..net , JAVA , Oracle , web designing , PHP, Drupal, Joomla

**INTRODUCTION**



**Inventory Control System** is GUI based desktop application developed in java. Inventory Control System application can be very help full for different warehouses of the organization to keep the information of goods in an organized way. It keeps the full information about the present quantity of each and every goods. It also manages the total number of goods that are stock in or stock out. It is very less time consuming as compare to manual system because here we just have to make the entry rest of thing is done by this software automatically.

**OBJECTIVE OF THE PROJECT**

The objective to develop this desktop application is that maintain and managing the record of goods in warehouses is very time consuming, complex, task and also require a lot of workers. So there is requirement to develop the application that manage all the records and take very less time as compare to the manual system. We just need one or two person who will handle this record and all the other things are done by this software.

**MODULES AND THEIR DESCRIPTION**

1. Login Module
2. File
   1. Change Password
3. New Entry
   1. New Company
   2. New Product
4. Stock In
   1. Add Quantity
   2. Product Detail
5. Stock Out
   1. Stock Out Product

**1. Login Module-:**

Login Module is used by the user to login. Login is required start this application. Every user must have a user name and password for login.

**2. File -:**

This Module contain a sub module and that module is

**Change Password-**: This Option can be helpful for the user to change the existing password that is required for successful login.

**3. New Entry-**:

This Module is used when every have to make a new entry either of product or company. This module contains two sub modules.

**New Company-:** It is used add the entry of new company whose product is going to be part of our inventory.

**New Product-:** It is used to add the information about the new product of existing company into our existing system.

**4.** **Stock In-:**

This module is used whenever we add a more quantity of the product into the inventory. It contains two sub modules.

**Add Quantity-:** This is used to add more quantity of existing product into our inventory.

**Product Detail-:** It provides you a complete detail of each and every product.

1. **Stock Out-:**

It Contain a sub module-:

**Stock out Product-:**  This module is used whenever any product is stock out from the inventory.

**HARDWARE AND SOFTWARE REQUIREMENTS**

**Hardware:** Hardware is the physical part of the computer system like mouse, keyboard, monitor etc.

**HARDWARE REQUIREMENTS:**

1. Any PC processor

2. 512 MB Ram

3. Keyboard mouse

4. Internet Connection

**Software:** Software is a set of applications which is used to run the operating system.

The software requirements to successfully run all the programs are:

**SOFTWARE REQUIREMENTS:**

1. Software Java 1.6, Java 1.7, Java 1.8
2. Database SQL,ORACLE 8G,ORACLE 10G
3. Editor Netbeans 8.1

**TOOLS/PLATFORM USED**

**Languages** Java, JDBC

**Front End** Swings **(**Core Java)

**Platform** J2SE (Java to Standard Edition)

**Back End** Oracle 10g

**ABOUT JAVA**

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language
* Java is cohesive and consistent
* Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is to Internet Programming where c was to System Programming.

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language
* Java is cohesive and consistent
* Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is for Internet Programming where C is for System Programming.

**IMPORTANCE OF JAVA TO THE INTERNET**

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. in the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

**APPLICATIONS AND APPLETS**

An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++ .Java’s ability to create Applets makes it important. An Applet is an application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

**JAVA ARCHITECTURE**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

**COMPLILATION OF CODE**

## When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed t executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines .This machine is called Java Virtual Machine.

**COMPILING AND INTERPRETING JAVA SOURCE CODE**

**Source code**

**Pc compiler**

**Macintosh compiler**

**SPARC Compiler**

**Java Byte code**

**Platform independent**

**Java interpreter**

**Java interpreter (macintosh)**

**)))**

**Java interpreter (SPARC)**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

**SIMPLE**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++ . Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

### OBJECT ORIENTED

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

### ROBUST

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

**ANALYSIS**

**FEASIBILITY STUDY:-**

A feasibility study is a preliminary study which investigates the information needs of perspective users and determines the resource requirements, determining the cost effectiveness of various alternatives in the designs of the information system, benefits and feasibility of proposed project.

The goal of the feasibility study is to evaluate alternative systems to propose the most feasible and desirable systems for development.

**The feasibility of our proposed system can be evaluated as:-**

1. **TECHNICAL FEASIBILITY.**
2. **SCHEDULE FEASIBILITY.**
3. **OPERATIONAL FEASIBILITY.**
4. **ECONOMIC FEASIBILITY.**

**TECHNICAL FEASIBILITY:-**

Technical feasibility can be demonstrated if reliable hardware and software capable of meeting needs of proposed system can be developed or acquired by the business in required time. Our project is technically feasible because the required hardware and software needed for our project are available.

**SCHEDULE FEASILBILTY: -**

The scheduled feasibility of the system evaluates whether the system finishes its task within the provided time of the development or not. Our project is feasible according to schedule because our project is being completed within the specified time period of our semester.

**OPERATIONAL FEASIBILITY: -**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. In the manual system, it is very difficult to maintain huge amount of pricing information of products. The development of the new system was started because of the requirements put forward by the management of the concerned department. So it is sure that the system development is operationally feasible.

**ECONOMIC FEASIBILITY:-**

Here the development cost is evaluated by weighing it against the ultimate benefits derived from the new system,. The benefit accrued from the new system is more than the cost involved in its development as everything is related to money.

The proposed system is economically feasible because the cost involved in purchasing that hardware and the software are within approachable. The operating –environment costs are marginal .The less time involved also helped in its economical feasibility.

Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

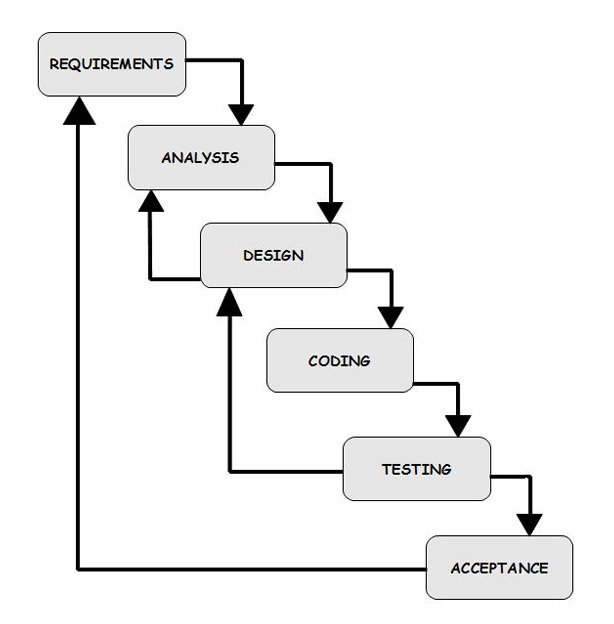
The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, cost and benefits are estimated with the greater accuracy at this stage.

The result of the feasibility study is a formal proposal. This is simply a report — a formal document detailing the nature and scope of the proposed solution. The proposal summarizes what is known and what is going to be done.

**SOFTWARE MODEL**

**Waterfall Model:-**

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Coding, Testing, Implementation, and Maintenance .

****

The Waterfall Model (System Development Life Cycle)

In waterfall model, the following phases are followed in order:

1. Requirements specification
2. Feasibility Study
3. Design & Coding
4. Integration & Testing
5. Implementation
6. Maintenance

**Requirement Analysis & Definition:** All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

**System & Software Design:** Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

**Implementation & Unit Testing:** On receiving system design documents, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality; this is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

**Integration & System Testing:** As specified above, the system is first divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. After successfully testing the software, it is delivered to the customer.

**Operations & Maintenance:** This phase of "The Waterfall Model" is virtually never ending phase (Very long). Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred as Maintenance.

**JDBC**

**WHAT IS JDBC?**

Any relational database. One can write a single program using the JDBC API, and the JDBC is a Java API for executing SQL, Statements(As a point of interest JDBC is trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API

Using JDBC, it is easy to send SQL statements to virtually program will be able to send SQL .statements to the appropriate database. The Combination of Java and JDBC lets a programmer writes it once and run it anywhere.

**WHAT DOES JDBC DO?**

Simply put, JDBC makes it possible to do three things

* Establish a connection with a database
* Send SQL statements
* Process the results
* JDBC Driver Types
* The JDBC drivers that we are aware of this time fit into one of four categories
* JDBC-ODBC Bridge plus ODBC driver
* Native-API party-java driver
* JDBC-Net pure java driver
* Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC, data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than development and very limited development.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavours. There are four driver categories

**TYPE 01-JDBC-ODBC BRIDGE DRIVER**

Type 01 drivers use a bridge technology to connect a java client to an ODBC database service. Sun’s JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

**DRIVER TYPE 02-NATIVE-API PARTY-JAVA**

Type 02 drivers wrap a thin layer of java around database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for **C/C++** programmers, because type-02 drivers are implemented using native code. in some cases they have better performance than their all-java counter parts. They add an element of risk, however, because a defect in a driver’s native code section can crash the entire server

**TYPE 03-NET-PROTOCOL ALL-JAVA DRIVER**

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all java, which makes them useful for applet deployment and safe for servlet deployment

**TYPE-04-NATIVE-PROTOCOL ALL-JAVA DRIVER**

Type o4 drivers are the most direct of the lot. Written entirely in java, Type 04 drivers understand database-specific networking. Protocols and can access the database directly without any additional software

**JDBC-ODBC BRIDGE**

If possible use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC. It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge (that is, the Bridge native library, the ODBC driver manager library, library, the ODBC driver library, and the database client library)

**WHAT IS THE JDBC-ODBC BRIDGE?**

The JDBC-ODBC Bridge is a JDBC driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge is implemented as the sun.jdbc.odbc Java package and contains a native library used to access ODBC. The Bridge is joint development of Intersolv and Java Soft

**ORACLE**

Oracle is a relational database management system, which organizes data in the form of tables. Oracle is one of many database servers based on RDBMS model, which manages a seer of data that attends three specific things-data structures, data integrity and data manipulation.

With oracle cooperative server technology we can realize the benefits of open, relational systems for all the applications. Oracle makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd’s rules.

**FEATURES OF ORACLE**

**PORTABLE**

The Oracle RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

**COMPATIBLE**

Oracle commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from Oracle, which is Oracle compatible with DB2. Oracle RDBMS is a high performance fault tolerant DBMS, which is specially designed for online transaction processing and for handling large database applications.

**MULTITHREADED SERVER ARCHITECTURE**

Oracle adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the Oracle DBMS server code to eliminate all internal bottlenecks.

Oracle has become the most popular RDBMS in the market because of its ease of use

* Client/server architecture.
* Data independence.
* Ensuring data integrity and data security.
* Managing data concurrency.
* Parallel processing support for speed up data entry and online transaction processing used for applications.
* DB procedures, functions and packages.

**Dr.E.F.CODD’S RULES**

These rules are used for valuating a product to be called as relational database management systems. Out of 12 rules, a RDBMS product should satisfy at least 8 rules + rule called rule 0 that must be satisfied.

**RULE 0: FOUNDATION RULE**

For any system to be advertised as, or claimed to be relational DBMS should manage database with in itself, without using an external language

**RULE 1: INFORMATION RULE**

All information in relational database is represented at logical level in only one way as values in tables.

**RULE 2: GUARANTEED ACCESS**

Each and every data in a relational database is guaranteed to be logically accessibility by using to a combination of table name, primary key value and column name.

**RULE 3: SYSTEMATIC TREATMENT OF NULL VALUES**

Null values are supported for representing missing information and inapplicable information. They must be handled in systematic way, independent of data types.

**RULE 4: DYNAMIC ONLINE CATALOG BASED RELATION MODEL**

The database description is represented at the logical level in the same way as ordinary data so that authorized users can apply the same relational language to its interrogation as they do to the regular data.

**RULE 5: COMPREHENSVE DATA SUB LANGUAGE**

A relational system may support several languages and various models of terminal use. However there must be one language whose statement can express all of the following DataDefinitions,ViewDefinitions, DataManipulations,Integrity,Constraints,Authorization .

**RULE 6: VIEW UPDATING**

Any view that is theoretical can be updatable if changes can be made to the tables that effect the desired changes in the view.

**RULE 7: HIGH LEVEL UPDATE, INSERT AND DELETE**

The capability of handling a base relational or derived relational as a single operand applies not only retrieval of data also to its insertion, updating, and deletion.

**RULE 8: PHYSICAL DATA INDEPENDENCE**

Application program and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access method.

**RULE 9: LOGICAL DATA INDEPENDENCE**

Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access methods.

**RULE 10: INTEGRITY INDEPENDENCE**

Integrity constraints specific to particular database must be definable in the relational data stored in the catalog, not in application program.

**RULE 11: DISTRIBUTED INDEPENDENCE**

Whether or not a system supports database distribution, it must have a data sub-language that can support distributed databases without changing the application program.

**RULE 12: NON SUB-VERSION**

If a relational system has low level language, that low language cannot use to subversion or by pass the integrity rules and constraints expressed in the higher level relational language.

**ORACLE SUPPORTS THE FOLLOWING CODD’S RULES**

Rule 1: Information Rule (Representation of information)-YES.

Rule 2: Guaranteed Access-YES.

Rule 3: Systematic treatment of Null values-YES.

Rule 4: Dynamic on-line catalog-based Relational Model-YES.

Rule 5: Comprehensive data sub language-YES.

Rule 6: View Updating-PARTIAL.

Rule 7: High-level Update, Insert and Delete-YES.

Rule 8: Physical data Independence-PARTIAL.

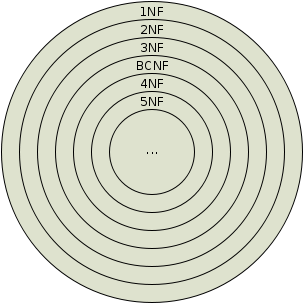
Rule 9: Logical data Independence-PARTIAL.

Rule 10: Integrity Independence-PARTIAL.

Rule 11: Distributed Independence-YES.

Rule 12: Non-subversion-YES

**NORMALIZATION**

****

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files. The tables are organized to:

* Reduced duplication of data.
* Simplify functions like adding, deleting, modifying data etc..,
* Retrieving data
* Clarity and ease of use
* More information at low cost

Normalization is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints on the kind of functional dependencies that could be associated with the relation. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

**FIRST NORMAL FORM**

A relation R is in first normal form if and only if all underlying domains contained atomic values only.

**SECOND NORMAL FORM**

A relation R is said to be in second normal form if and only if it is in first normal form and every non-key attribute is fully dependent on the primary key.

**THIRD NORMAL FORM**

A relation R is transitively depends on the primary key. said to be in third normal form if and only if it is in second normal form and every non key attribute is non transitively depend on the primary key.

**In BCNF:**

Though the relation is in 3NF, 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keysi.e. composite candidate keys with at least one attribute in common.

BCNF is based on the concept of a determinant.A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.

A relation is in BCNF is, and only if, every determinant is a candidate key.

When a relation has more than one candidate key, anomalies may result even**:** A relation is in Boyce-Codd Normal Form (BCNF) if every determinant is a candidate key. (See the links in the box at right for definitions of determinant and candidate key.)  
Whereas BCNF insists that for this dependency to remain in a relation, A must be a candidate key.

**Fourth Normal Form (4th NF)**  
An entity is in Fourth Normal Form (4NF) when it meets the requirement of being in Third Normal Form (3NF) and additionallyhas no multiple sets of multi-valued dependencies. In other words, 4NF states that no entity can have more than a single one-to-many relationship within an entity if the one-to-many attributes are independent of each other.

Fourth Normal Form applies to situations involving many-to-many relationships.In relational databases, many-to-many relationships are expressed through cross-reference-tables.  
  
 A table is in fourth normal form (4NF) if and only if it is in BCNF and contains no more than one multi-valued dependency.

**Fifth Normal Form (5th NF)**  
  
A relation that has a join dependency cannot be decomposed by a projection into other relations without spurious results

A relation is in 5NF when its information content cannot be reconstructed from several smaller relations i.e. from relations having fewer attributes than the original relation

Fifth normal form, also known as join-projection normal form (JPNF), states that no non-trivial join dependencies exist. 5NF states that any fact should be able to be reconstructed without any anomalous results in any case, regardless of the number of tables being joined. A 5NF table should have only candidate keys and it's primary key should consist of only a single column.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM & DISADVANTAGES**

In the existing system, all the work done manually.

* It is very time consuming
* It is Very Complex
* It requires a lot of man force.

**PROPOSED SYSTEM & ITS ADVANTAGES**

In the proposed system the all the work done automatically by our software, we just have to enter the detail of work.

* Less time Consuming
* Result generate by the system is very accurate.
* Less Complex
* Requires less man force.

**SYSTEM DESIGN**

Systems design is the process or art of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is someoverlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

**UML {UNIFIED MODELING LANGUAGE}**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules..A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**USER MODEL VIEW**

This view represents the system from the users perspective.The analysis representation describes a usage scenario from the end-users perspective.

**STRUCTURAL MODEL VIEW**

In this model the data and functionality are arrived from inside the system.This model view models the static structures.

**BEHAVIORAL MODEL VIEW**

It represents the dynamic of behavoriral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**IMPLEMENTATION MODEL VIEW**

In this the structural and behavioural as parts of the system are represented as they are to be built.

**ENVIRONMENTAL MODEL VIEW**

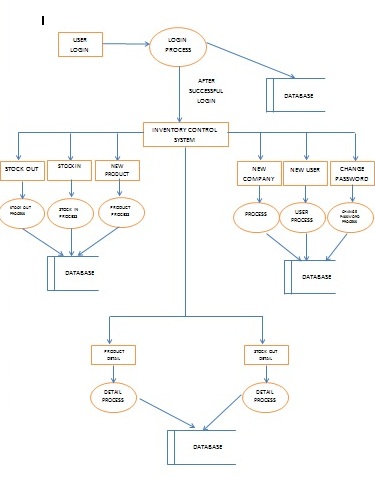
In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are:

UML Analysis modelling, this focuses on the user model and structural model views of the system.UML design modelling, which focuses on the behaviouralmodelling, implementation modelling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

**DATA FLOW DIAGRAMS**

****

**SYSTEM TESTING**

****

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

**TESTING IN STRATEGIES**

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**UNIT TESTING**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Each module can be tested using the following two Strategies:

**BLACK BOX TESTING**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structure or external database access
* Performance errors
* Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

**WHITE BOX TESTING**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

* Guarantee that all independent paths have been Executed.
* Execute all logical decisions on their true and false Sides.
* Execute all loops at their boundaries and within their operational bounds
* Execute internal data structures to ensure their validity.

**INTEGRATING TESTING**

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

**ACCEPTANCE TESTING**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

**TEST APPROACH**

Testing can be done in two ways:

* Bottom up approach
* Top down approach

**BOTTOM UP APPROACH**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

**TOP DOWN APPROACH**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

**VALIDATION**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed

**IMPLEMENTATION**

**OVERVIEW OF IMPLEMENTATION**

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes; they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next. The implementation phase is less creative than system design. It is primarily with user training, site preparation and file conversion. When the candidate system is linked to terminals or remote sites, the telecommunication network and test of the network along with system are included under the implementation. During final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place at about the same time the user is being trained.

Evaluation of the system is performed to identify its strengths and weaknesses.The actual evaluation can occur along any of the following dimensions:-

1. **Operational Evaluation:**Assessment of the manner in which the system functions, including case of use, response time, overall reliability and level of utilization.
2. **Organizational Impact:**Identification and measurement of benefits to the organization in such areas as financial concerns, operational efficiency and competitive impact.
3. **User ManagerAssessment:**Evaluation of the attitudes of seniors and user manager within the organization, as well as end-users.

**IMPLEMENTATION TOOLS:**

* Training of personnel
* Conversion procedures
* Post-implementation review or evaluation

**Training of personnel:**

The high quality training is an essential step in systems implementation. Hence to provide the training to personnel they have been provided user manuals. They are asked to read it carefully and same thing try practically on computer. If they don’t understand anything, can ask without any hesitation. They are asked to enter the data, which is more frequently entered, and print the reports are frequently printed. Users are told about those situations, which they must understand and should able to handle it.

**Conversion Method:**

The direct conversion method is applied. This method converts from old to the new system abruptly. The old system is used till a planned conversion day. The organization relies fully on the new system.

**Evaluation Or Post Implementation Review:**

After the system is implemented and conversion is complete, a review is conducted to determine whether system is meeting expectations and where improvements are needed. A post implementation review measures the systems performance against pre-determined requirements. It determines how well the system continues to meet performance specifications. It also provides information to determine whether major re-design or modification is required. In evaluation system is checked against the pre-determined requirements. All the requirements have been fully attained.

**REVIEW PLAN:**

For review a team is planned. The team shall prepare a formal plan around the objectives of the review. An overall plan covers the following areas:-

* **Administrator Plan:**Reviewarea objectives, operating costs, actual operating performance and benefits.
* **Personnel Requirements Plan:**Review performance objectives and training performance to data.
* **Hardware Plan**:Review performance specifications
* **Documentation Review Plan**:Review the system development efforts

**Administrative Plan**

**User Objective**:This is an extremely crucial area since it may be possible that over a period of time the system does not meet the initial objectives of the user or objectives get changed as a result of changes in the overall objectives of the organization. The results of the evaluation are documented for future reference.

Operating costs and benefits: Under the administration plan, current budget designed manipulate costs and savings of the system will closely reviewed.

**Personnel Requirement Plan**

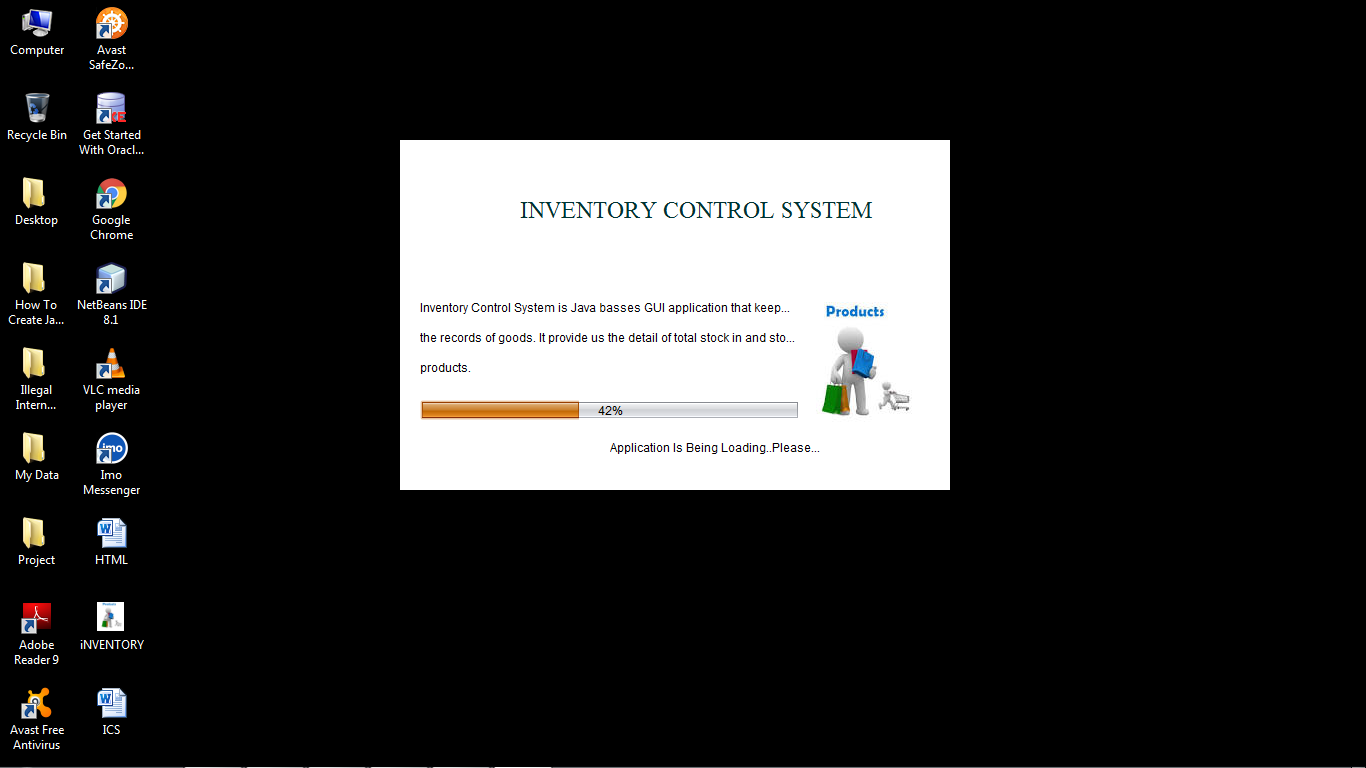
Personnel old performance objectives will be compared with current performance levels. Training performance through testing, conducting interviews and other data gathering techniques.

**Hardware Plan**

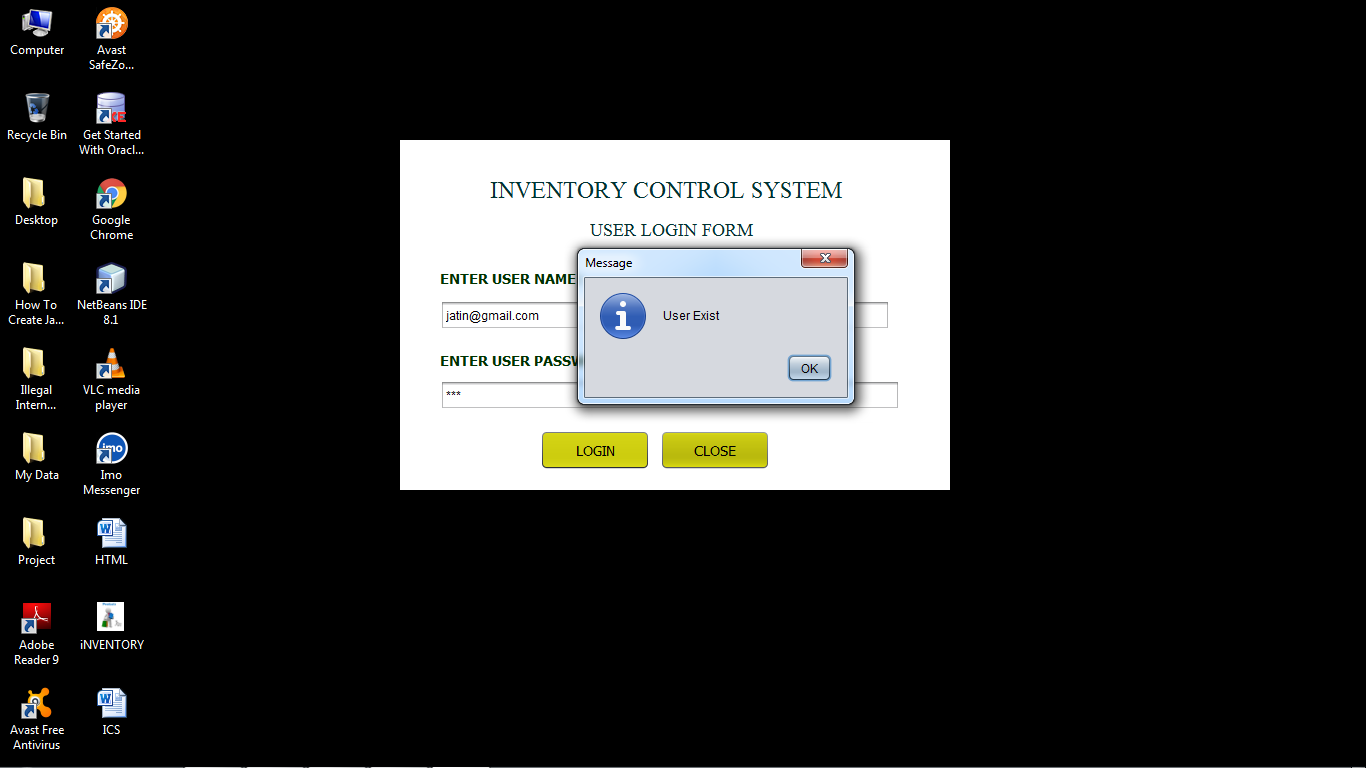
The hardware of the new system will also review. But as we are using existing hardware which fulfill the requirement of the project. Hence the cost of hardware is not increased.

**SCREEN SHOTS**

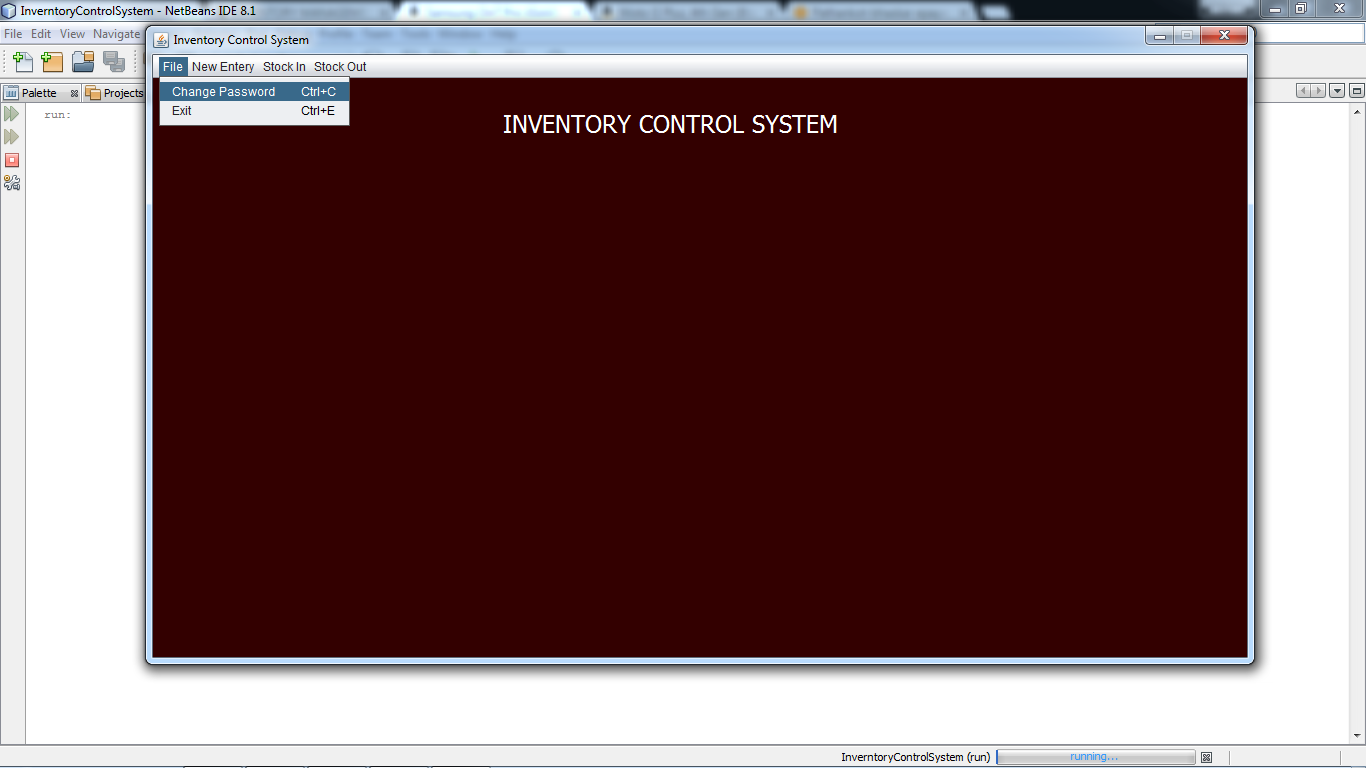
**SPLASH BOX**



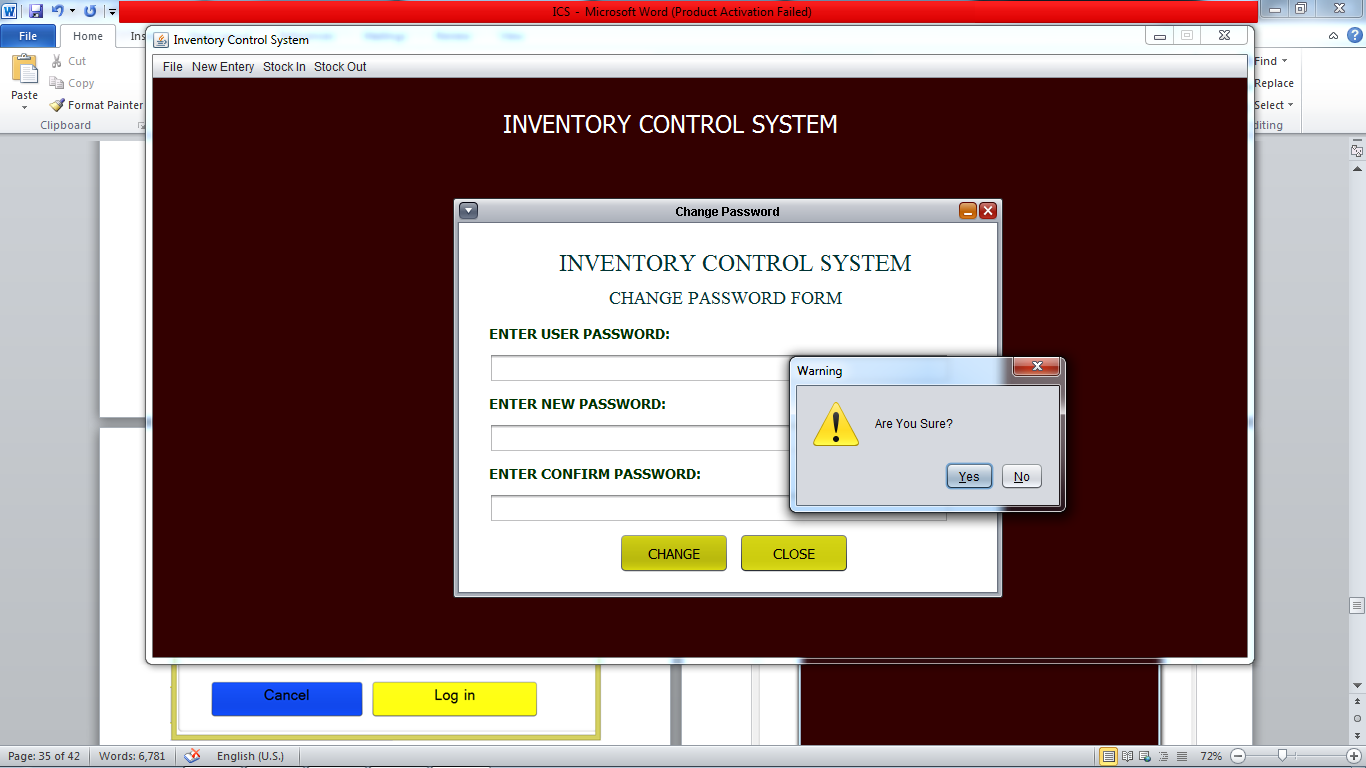
**LOGIN PAGE**



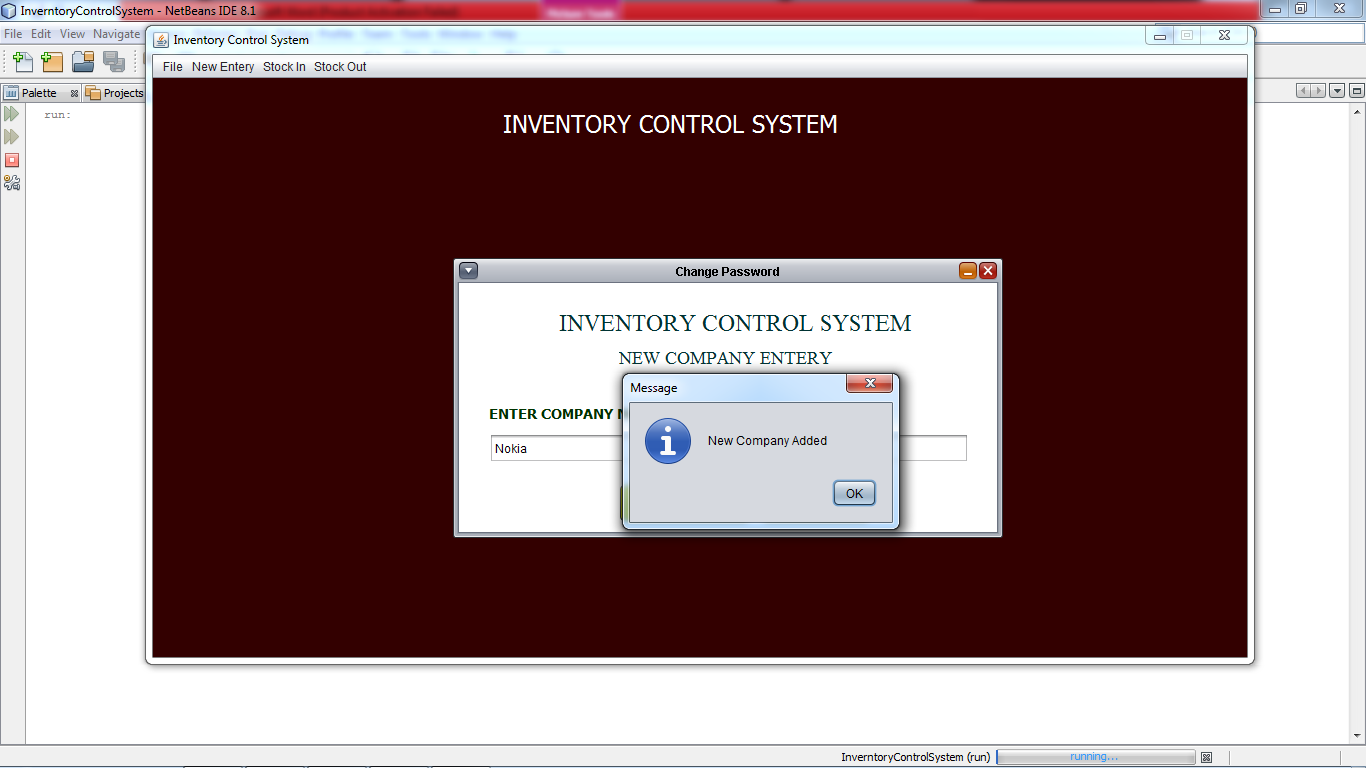
**MENU FORM**



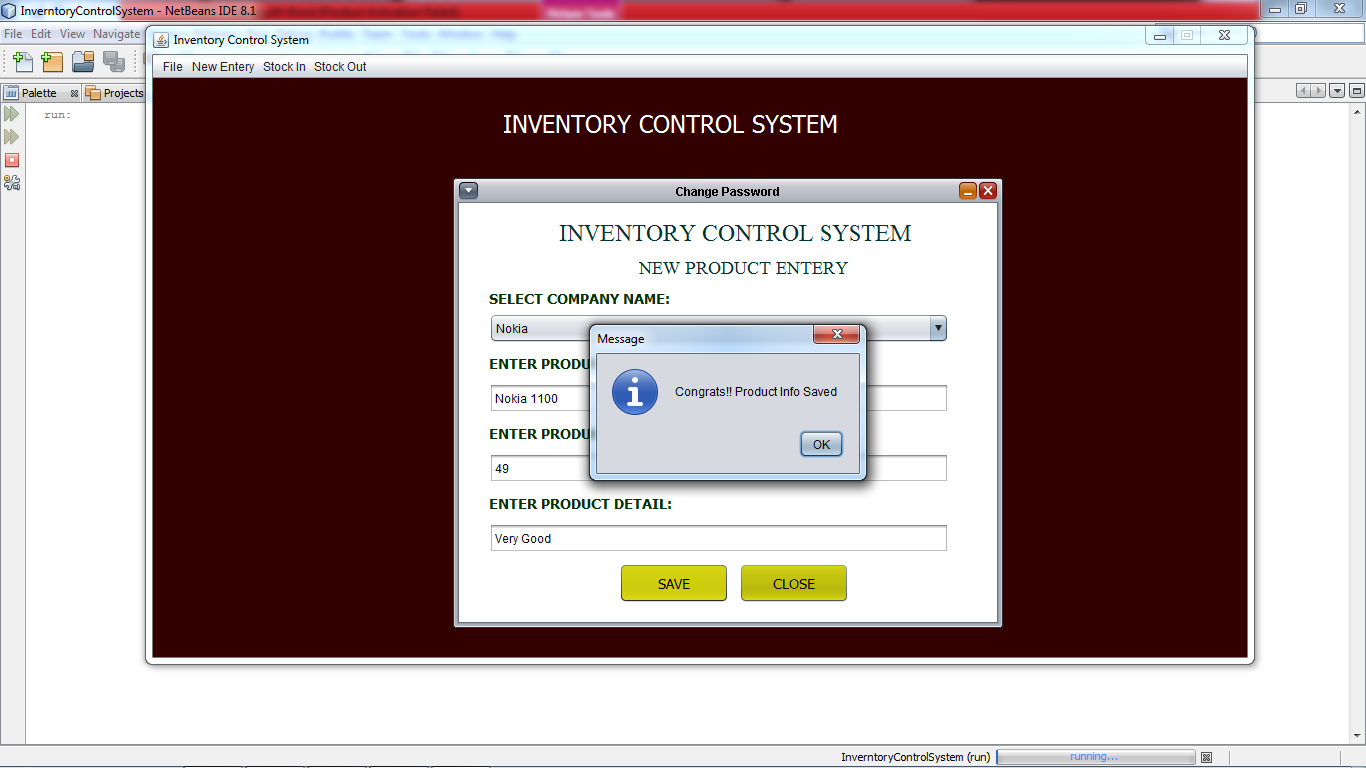
**CHANGE PASSWPRD**



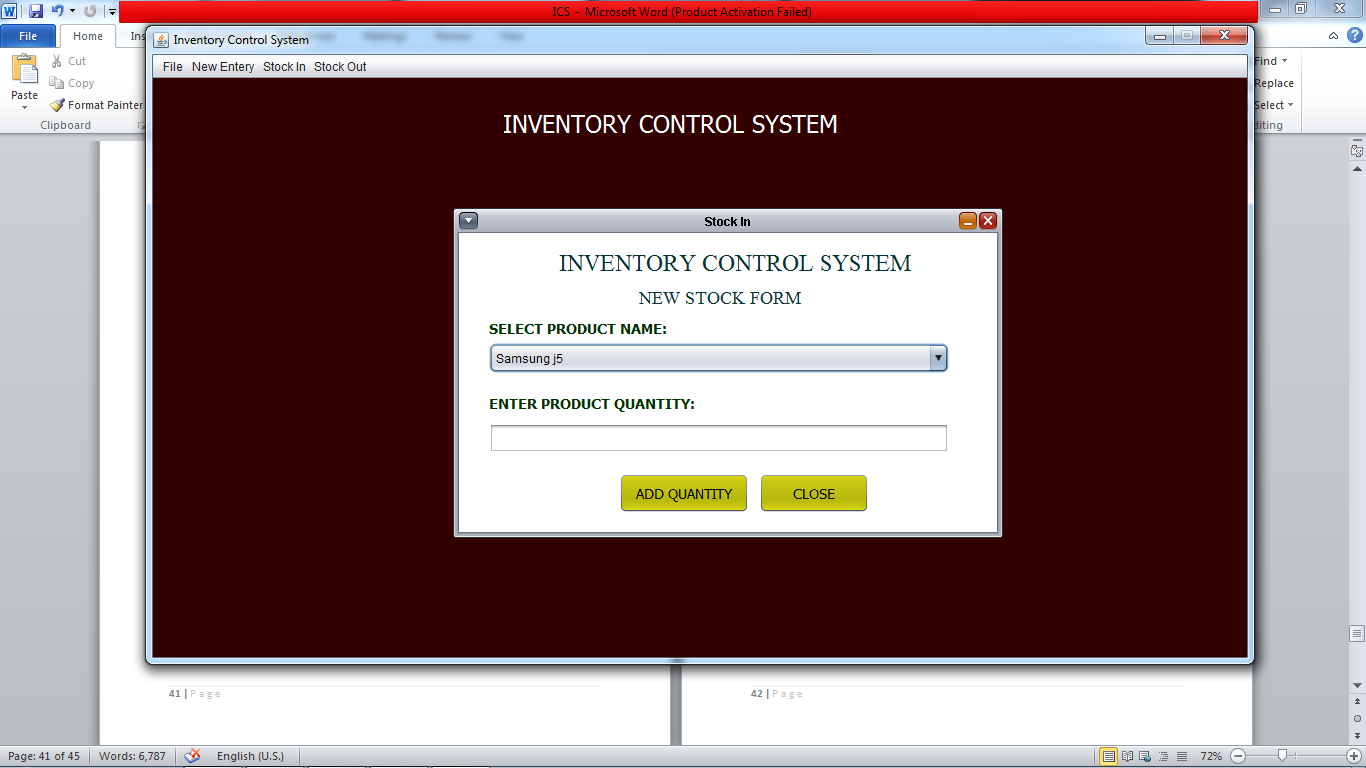
**NEW COMPANY**



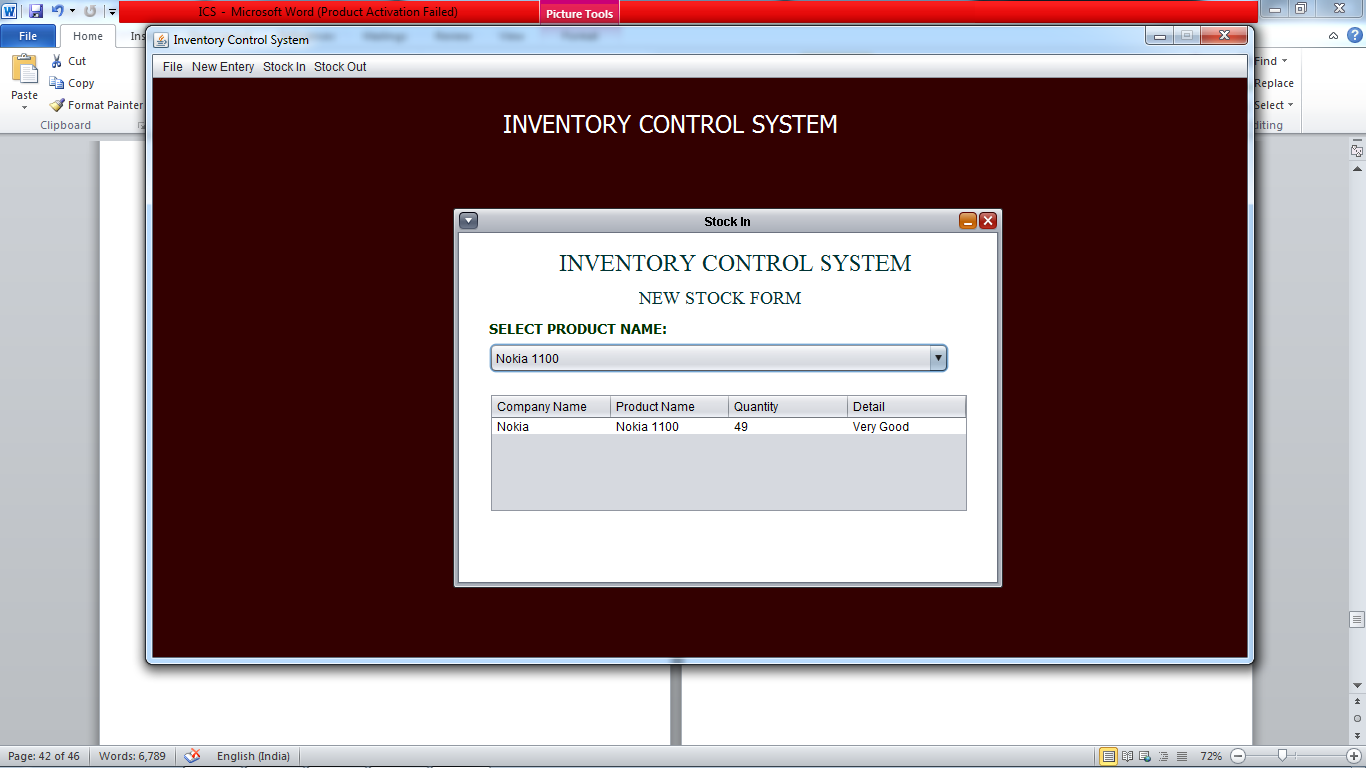
**NEW PRODUCT**



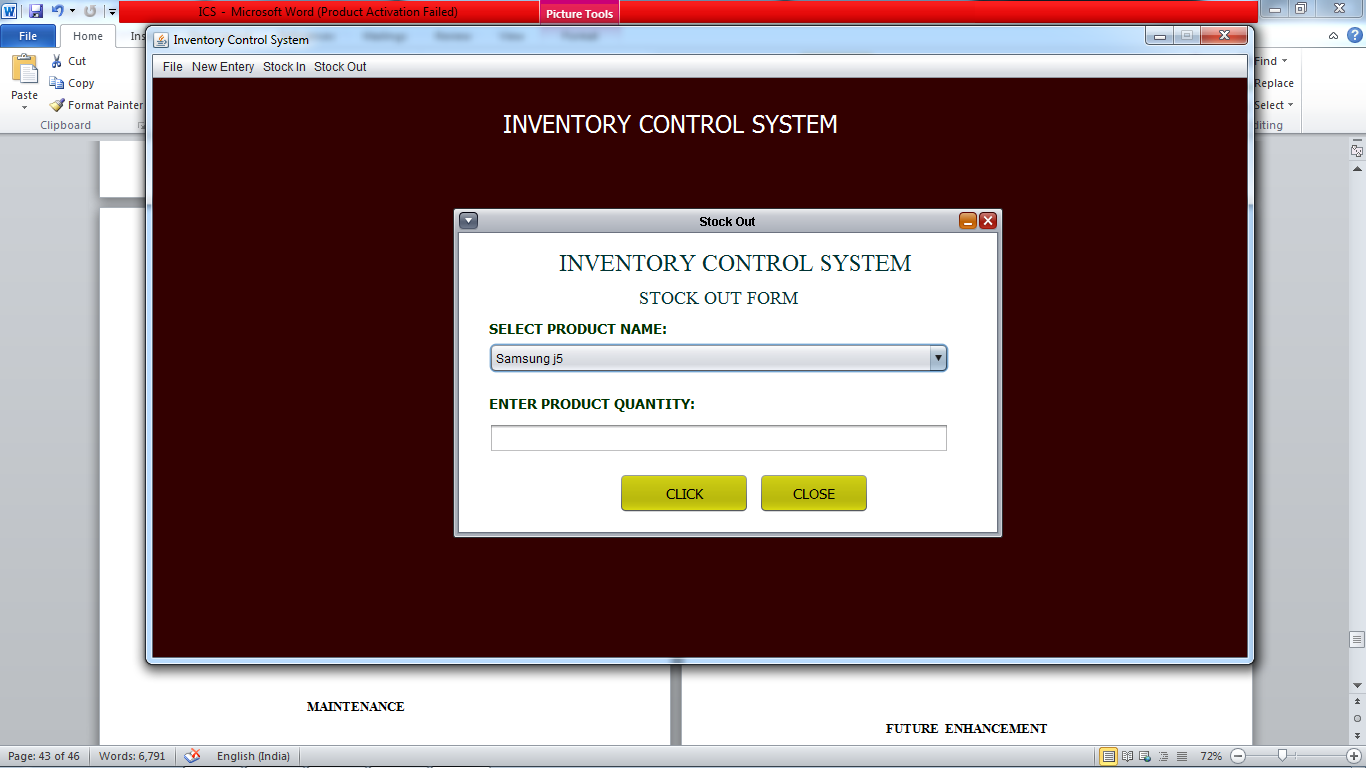
**NEW STOCK**



**PRODUCT DETAIL**



**STOCK OUT**



**MAINTENANCE**

**Introduction to Software Maintenance:** Software maintenance denotes any changes to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. It is practically impossible to make the software completely error free because the input domain of most software products is very large and it is not practical to test the software exhaustively with respect to each value that the input data may assume. Maintenance is also needed to enhance the features of the software to add more functionality to it and to port to new platforms etc.

**Types of Software Maintenance**

The requirement of software maintenance arises on account of the three main reasons:-

1. **Corrective:**Corrective Maintenance of a software product becomes necessary to rectify the bugs observed while the system is in use.
2. **Adaptive:**A software product might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to be interfaced with new hardware or software.
3. **Perfective:**Asoftware product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to customer demands, or to enhance the performance of the system.

**FUTURE ENHANCEMENT**

Even though the project full fills the requirements of the present application there is always scope for further work. According to the emerging changes and new versions, further work can be done to improve the application since project is designed in flexible software.

Our current application is work only for one warehouse at time, we can make it more advance that only single application is work for all the warehouses of the organization all over the country or world.

**BIBLIOGRAPHY**

**BOOKS REFERED**

* The Complete Reference (English) 5th Edition

**JAVA TECHNOLOGIES**

* Larne Pekowsley, JAVA Server Pages
* Nick Todd, JAVA Server Pages
* Pistoria, JAVA2 Networking
* Scott oaks, JAVA Security
* Shadabsiddiqui, J2EE Professional
* Sierra Bates, Head First EJB
* Yehuda Shiran, Java Script Programming
* Mastering JAVA Security
* JAVA Complete Reference

**JDBC**

Patel Moss, JAVA Database Programming with JDBC

**SOFTWARE ENGINEERING**

Roger Pressman, Software Engineering