Inventory Management System For Retailers

Team Id: **PNT2022TMID38522**

Team size: 4

Team Leader: SUNDARAPANDIYAN S

Team member: INZHAMAM AL HUQ B

Team member: DHINESH KUMAR S

Team member: DHIVEEN KUMAR E

Table of Contents

1 INTRODUCTION

1.1 About Project

Smart Inventory Management System is an online software application which fulfills the requirement of a typical Stock Analysis in various godowns. It provides the interface to users in a graphical way to manage the daily transactions as well as historical data. Also provides the management reports like monthly inwards, monthly deliveries and monthly returns.

This application maintains the centralized database so that any changes done at a location reflects immediately. This is an online tool so more than one user can login into system and use the tool simultaneously.

The aim of this application is to reduce the manual effort needed to manage transactions and historical data used in various godowns. Also this appli cation provides an interface to users to view the details like the daily Stock Statements of all godowns.

1.2 Business Requirement Analysis:

It is a Stock Storage department related to Reliance Industries which sells different items to the public through their retail outlets like Reliance Fresh, Reliance Foot Wear etc. Since the company needs to maintain all their goods and items at a separate storage area which requires separate department also for tracking the details related to in and outs of all godowns. The in and out's will henceforth be referred to as inwards and outwards.

Prior to storage of goods this department has to manage different godowns, product- wise stocks in each godown, inwards and outwards of each godown. It has to hold the details of all gowdowns like godown id, godown location, capacity in quintals, godown manager and start date. Whenever a product comes into a particular godown then the details like Godown ID, Name of the Supplier, Date of Supply, Item Name, Invoice No, quantity, received by, receipt no and bill checked by need to be stored in the inwards register of the that godown by godown manager.

Whenever stock comes out from the godown then the details like Godown ID, Item Name, Invoice No, Date of Supply, Date of delivery, Delivered to, Quantity, Purpose (Sale/Service), Receipt No, Bill Value, Bill Checked by need to be stored in the outwards register by the godown manager.

Whenever a customer returns a stock to the gowdown then we need to check the reason for returning that item. If it is a damage then the details like Item Name, Date of delivery, date of return, Return Godown ID, Quanity, invoice no, returned by, receipt no, bill value and checked by needs to stored in returns register. If the reason is order cancelled then we need to update the stock no in that godown. Checking for particular inwards, outwards or returns entry info takes lot of time here.

Thus the cycle is repeated for every day. Currently all the above activities are done manually. The process is a tedious one. To arrive at the Inwards, outwards or returns for items, data has to be gathered from various sources. Because of this errors are occurring in the process, which is leading to delayed deliveries to the customers. Some times because of the errors wrong products are sent out which have no requirement & thus a lot of money is being wasted in maintaining the stock.

Automating such a process will not only eliminate the errors in the process, but also bring down the delivery times & make the company more competitive. So it was decided that an automated system should be developed to make the whole process simpler & easier.

The following is the system developed for the above stated needs. An initial feasibility study was performed & a conclusion was arrived at that automating such a system will not only achieve all the things mentioned above, but will also provide additional Reports which will enable the Management to look at the statistical side of the inwards, inwards & returns related to each godown. This would also create an effective Stock management system, which would reduce the confusion in maintaining the stocks at different godowns, thus effectively reducing the expenditure costs of the company. Another advantage was that the whole Accounts system could be linked to this system in future, which would finally reduce the Overheads of the company.

1.3 Project Plan

It was decided to use good Software engineering principals in the development of the system since the company had quite a big Stock management & was aiming to add new godowns & employees & expand their operations in the near future. So the following Project Plan was drawn up:

1. The Analysts will interact with the current manual system users to get the Requirements. As a part of this the Requirements Specification Document will be created.

2. The requirements Specifications document will contain the Analysis & Design of the system & UML will be used as the modeling language to express the Analysis & Design of the System. According to Grady Booch et al, in The Unified Modeling Language User Guide **[UML-1998]**, “The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML gives you a standard way to write a system's blueprints, covering conceptual things, such as business processes and system functions, as well as concrete things, such as classes written in a specific programming language, database schemas, and reusable software components”.

3. The Analysis, Design, Implementation & testing of the System itself will be broadly based on the Rational Unified Software Development process. According to Ivar Jacobson et al, in The Unified Software Development Process (The Addison- Wesley Object Technology Series) **[USDP-2000]**, the Unified Software Development Process contains Inception, Elaboration, Construction & Transition as the main Phases, which contain further cycles & iterations. This process will be followed to

produce an incremental cycle, which will deliver milestones like the Requirements Specification Document etc., at the end of each of the iterations, Phases or cycles.

4. The Architecture & Technologies will be decided as a part of the Analysis of the requirements.

5. Once the Design is ready the Implementation & Testing strategy of the system will commence. Each will be independent of the other. The implementation of the system itself will be broken down into sub-systems following the Software Engineering principles for the development of robust software.

6. Once the implementation is ready, the System testing will take place. If the system is judged to be stable then Acceptance testing by the Users will take place & once the Users are satisfied the System will be rolled out to the Users & they will be trained on how to use it for an initial period.

The following chapters contain an account of how the Technology & architecture for the system were chosen.

1.4 Requirements Specification Document

According to Roger Pressman in Software Engineering: A Practitioner's Approach (McGraw-Hill Publications) **[SEPA–1997]**, the requirement specification document is produced at the end of Analysis of the system. This document is a very comprehensive document & contains all the User requirements & Analysis diagrams. The Requirements are broadly divided into two groups:

1. Functional requirements
2. Non-functional requirements

1.5 Functional Requirements

The main purpose of functional requirements within the requirement specification document is to define all the activities or operations that take place in the system. These are derived through interactions with the users of the system. Since the Requirements Specification is a comprehensive document & contains a lot of data, it has been broken down into different Chapters in this report. The depiction of the Design of the System in UML is presented in a separate chapter. The Data Dictionary is presented in the Appendix of the system.

But the general Functional Requirements arrived at the end of the interaction with the Users are listed below. A more detailed discussion is presented in the Chapters, which talk about the Analysis & Design of the system.

1. The System holds all the details of the all the employees who are working in the organization.

2. It allows admin to manage two types of users, hold their details, authenticate these users at the time of login and accordingly provide different options.

3. It holds the details of all the godowns which are part of our organization.

4. It holds the details of all Product Stocks held in the ware-house of the company.

5. The system allows the godown manager to log into the system and enter their inwards entries related to their godown.

6. It also allows them to view the list of inward entries.

7. The system allows the godown manager to log into the system and enter their outward entries and their purpose related to their godown.

8. It also allows them to view the list of Outward entries.

9. Whenver an inwards entry is entered then accordingly the stock number will be automatically updated.

10. Whenver an outward entry is entered then accordingly the stock number will be automatically updated.

11. The system allows the godown manager to log into the system and enter stock return entries and the reason for return.

12. Whenver a return entry is entered then accordingly the stock number will be automatically updated if the reason is order cancelled otherwise It need not update the stock.

12. It allows the users to change their password for future security.

13. It allows the authorized users to a new employee at the time of creating a godown if the employee is a newly appointed for this godown.

14. It allows the admin to view the list of users and take the print.

15. It allows admin to generate godown details report.

16. It allows admin to generate inwards details report.

17. It allows admin to generate outwards details report.

18. It allows admin to generate returns details report.

19. It allows admin to generate stock statement report.

20. It allows any user to logout when he wants to come out from the system.

1.6 Non-Functional Requirements

The non-functional requirements consist of

1. Analysis, Design & Data requirements (Use-case diagrams, textual analysis, sequence diagrams, data dictionary etc.)
2. Constraints.
3. Guidelines.

1.7 Validation Criteria

1.7.1 Analysis, Design & Data requirements

The use case diagrams, textual analysis and sequence diagrams & data dictionary fall into this category. Since each category above is of considerable importance, they have been dealt in separate chapters. An outline is only included here.

The Analysis & Design phases of the system yield Use Case diagrams, textual analysis, Sequence Diagrams, Class diagrams & Data Dictionary. Data dictionary consists of process statements showing how data is flowing from starting point to end point.

1.7.2 Constraints

These are the requirements that are not directly related to the functionality of the system. These should be considered as mandatory when the system is developed. The following Constraints were arrived at for the system:

1. The system should be available over the intranet so that the Users like the godown managers & clerks can use the system from their respective locations which could be anywhere in the organization.

2. For gaining entry into the system the admin should register user info and the user should be able use login & passwords for gaining access to the system.

3. The users should be able to change their passwords for increased security.

4. The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.

6. There should be no limitation about the hardware platform that is to be used to run the system.

7. Data integrity should be maintained if an error occurs or the whole system comes down.

8. An inward entry should be entered in the database whenever stock comes into the godown. That is the number of items should be updated automatically.

9. An outward entry should be entered in the database whenever stock goes out into the godown. That is the number of items should be updated automatically.

10. A return entry should be entered in the database whenever stock returned into the godown. That is the number of items should be updated automatically.

1.7.3 Guidelines

We have discussed mandatory requirements in the previous section. The requirements in this section should be taken as suggestions & they should be thought of as recommendations to further enhance the usability of the system.

1. The system should display a user friendly menu for users to choose from.

2. The system should display godown ID and item to be selected from the popup list in the forms .

3. Services of the system should be available 24 hours a day.

4. The system should be designed in such a way that it is easy to enhance it with more functionality. It should be scalable & easily maintainable.

1.7.4 Validation Criteria

The Validation Criteria are dealt separately in the Chapter dealing with the Test Strategy & Test cases.

2 PROBLEM DEFINITION

2.1 Existing System:

Current system is a manual one in which users are maintaining ledgers, books etc to store the information like suppliers details, inwards, deliveries and returns of items in all godowns, customer details as well as employee details. It is very difficult to maintain historical data. Also regular investments need to purchase stationary every year.

2.1.1 Disadvantages:

The following are the disadvantages of current system

1. It is difficult to maintain important information in books
2. More manual hours need to generate required reports
3. It is tedious to manage historical data which needs much space to keep all the previous years ledgers, books etc

4. Daily transactions are to be entering into different books immediately to avoid conflicts which are very difficult.

2.2 Proposed System:

Proposed system is a software application which avoids more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the employees to use this application. They can easily use the tool that decreases manual hours spending for normal things and hence increases the performance. As the data is centralized it is very easy to maintain the stocks of the various items in all godowns.

2.2.1 Advantages:

The following are the advantages of proposed system

Easy to manage all the daily transactions

Can generate required reports easily

Easy to manage historical data in a secure manner

Centralized database helps in avoiding conflicts

Easy to use GUI that does not requires specific training.

3 DESIGN DOCUMENT

3.1 SOFTWARE REQUIREMENTS:

Database: MySQL 5.0, Server:

Apache Tomcat 4.1

Front end: JSP / Servlets, J2SDK 1.4, HTML, DHTML, Java Script

Editor: Edit plus

3.2 HARDWARE REQUIREMENTS

Pentium IV processes architecture

1. 256 MB RAM.
2. 40 GB Hard Disk Space.
3. Ethernet card.

3.3 Database Design:

3.3.1 Database Tables:

The total number of database tables that were identified to build the system is 7. The major part of the
Database is categorized as

Transactional components: The Transactional components are useful in recording the transactions made by the system. All the inwards, deliveries, returns etc information handled by these components

Data Dictionary components: These components are used to store the major information like Employee details, Godown details, Customer details, Items information etc.

General components: These components are used to store the general information like login information etc.

4 SYSTEM ANALASYS

4.1 System Design:

4.1.1 Users:

The major functionality of this product is divided into two categories.

1. Administrative User Functions.
2. Normal User Functions.

4.1.2 **Administrative User Functions:** Administrators can perform the following task

- Create new users
- Change the password
- Add/Update the details of Employees of the Company
- Add the information about the Godowns
- Can view the information about the Inwards
- Can view the information about the Deliveries
- Can view the information about the Returns
- Can view/generate management reports

4.1.3 **Normal User Functions:** Normal users can perform the following task

- Change the password
- View the details of Employees of the Company
- View information of different Godowns
- Can add the information about the Inwards
- Can add the information about the Deliveries
- Can add the information about the Returns
- Can view management reports

4.2 UML Diagrams

4.2.1 UML (UNIFIED MODELLING LANGUAGE)

The unified modeling language is a standard language for specifying, Visualizing, Constructing and Documenting the software system and its components. It is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand ,design , configure, maintain and control information about the systems.

4.2.2 Visualizing :

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think we cannot implement.

UML helps to visualize how the components of the system communicate and interact with each other.

4.2.3 Specifying :

Specifying means building models that are precise, unambiguous and completeUML addresses the specification of all the important Analysis Design, Implementation decisions that must be made in developing and deploying a software system. Constructing :UML's models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like Java or C++ or VB.
Forward Engineering and Reverse Engineering is possible through UML.

4.2.4 Documenting:

The deliverables of a project apart from coding are some artifacts which are critical in controlling, measuring and communicating about a system during its development viz.

Requirements, Architecture, Design, Source code, Project plans, Tests, Prototypes, Releases etc.

4.2.5 Diagrams in UML :

Diagrams are graphical presentation of set of elements . Diagrams projects a system, or visualize a system from different angles and perspectives.

The UML has Nine diagrams these diagrams can be classified into the following groups.
Static:

1. Class diagrams.
2. Object diagrams.
3. Component diagrams.
4. Deployment diagrams

4.2.6 Dynamic:

1. Use case diagram.
2. Sequence diagram.
3. Collaboration diagram.
4. State chart diagram.
5. Activity diagram.

4.3 Modules

The System after careful analysis has been identified to present with the following modules.

4.3.1 **EMPLOYEE INFORMATION MODULE**:

This module maintains all the information which belongs to the employees who are working for the company. It allows the administrator to add an employee record to the database very easily and it allows to view the list of employees in tabular format out of which he can edit a particular employee. Admin can take the print of employee report just by making a single on print icon and It also allows the administrator to remove an employee from list. It makes all the above can be done very flexibly.

4.3.2 **INWARDS MODULE:** This module maintains all the information related to manage inwards done in the godowns. All the inwards are recorded to database and can be viewed as a report that displays all the inwards made by the company at each godown. It allows the normal user to enter godown-wise inwards details whenever inwards done in any godown. It facilitates the user to select godown id from the list which prevents entering invalid godown ids and allows the user to select the directly from a calendar which reduces lot of confusion in date formats and all. It doesn't allow admin to enter the above details.

4.3.3 **DELIVERIES MODULE:** This module deals with major and crucial part which includes deliveries of items whose purpose is for sale or service. This module provides interface to add the deliveries details and can be viewed as a report that displays all the deliveries made by the company at each godown. It allows the normal to enter whenever some delivery to has to done from any godown. It facilitates the select godown id and item id from the list for better user-friendliness. It also asks the user to select purpose of the delivery whether it is sale or service. It provides an option to take the print out of delivery report.

4.3.4 **RETURNS MODULE:** This module deals with another major and crucial part which includes returns of items whose purpose is of damage or order cancelled. This module provides interface to add the returns details and can be viewed as a report that displays all the returns made by the customer at each godown. It allows the normal user to enter return details whenever a return will takes place at any godown. It provides the facility for the user to select the delivery items list out of which he can select id of return item very easily. It also facilitates the user to view returns report in tabular format.

4.3.5 **ADMINISTRATOR MODULE:** This module is used to manage the details of users of the application. Users are divided into two categories.

Admin Normal
user

It allows administrator to add a new user, view the list of user and delete a user from the list. It allows to send a print request to the printer for printing user report.

4.3.6 **REPORTING MODULE:** This module used to provide reports required by the higher management. It provides a facility to generate dynamic reports like information about the godown, monthly inwards, monthly deliveries, monthly returns and stock statements very easily.

4.4 Feasibility Analysis

Feasibility study is an important phase in the software development process. It enables the developer to have an assessment of the product being developed. It refers to the feasibility study of the product in terms of outcomes of the product, operational use and technical support required for implementing it.

Feasibility study should be performed on the basis of various criteria and parameters. The various feasibility studies are:

- ✔ Technical Feasibility
- ✔ Economic Feasibility

4.4.1 **Technical Feasibility:** The system is self-explanatory and does not need any extra sophisticated training. As the system has been built by concentrating on the Graphical User Interface Concepts, the application can also be handled very easily with a novice User. The overall time that is required to train the users upon the system is less than half an hour.

The System has been added with features of menu-driven and button interaction methods, which makes the user the master as he starts working through the environment. The net time the customer should concentrate is on the installation time.

4.4.2 Financial Feasibility:

i) **Time Based:** Contrast to the manual system management can generate any report just by single click. In manual system it is too difficult to maintain historical data which become easier in this system. Time consumed to add new records or to view the reports is very less compared to manual system. So this project is feasible in this point of view

ii) **Cost Based:** No special investment need to manage the tool. No specific training is required for employees to use the tool. Investment requires only once at the time of installation. The software used in this project is freeware so the cost of developing the tool is minimal and hence the overall cost.

4.4.3 Implementation Plan:

The main plan for the system developed is to mimic the existing system as it is in the proposed system.

5 Software Requirement Specification

5.1 What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

i) **Problem/Requirement Analysis**:
The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

ii) Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

5.2 Role of SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

5.3 PURPOSE:

The purpose of this document is to describe all external requirements for the Stock Analyzer. It also describes the interfaces for the system.

The Basic purpose of developing this project to cater the needs any large scale stock management departments like FCI

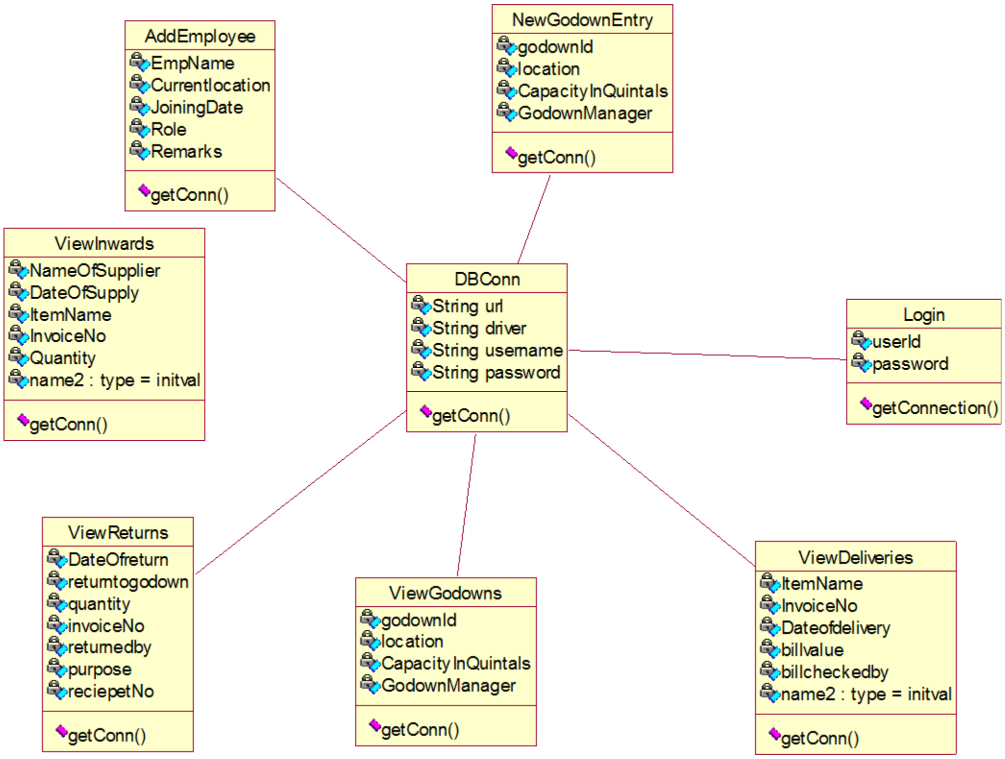
5.4 SCOPE:

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also by the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

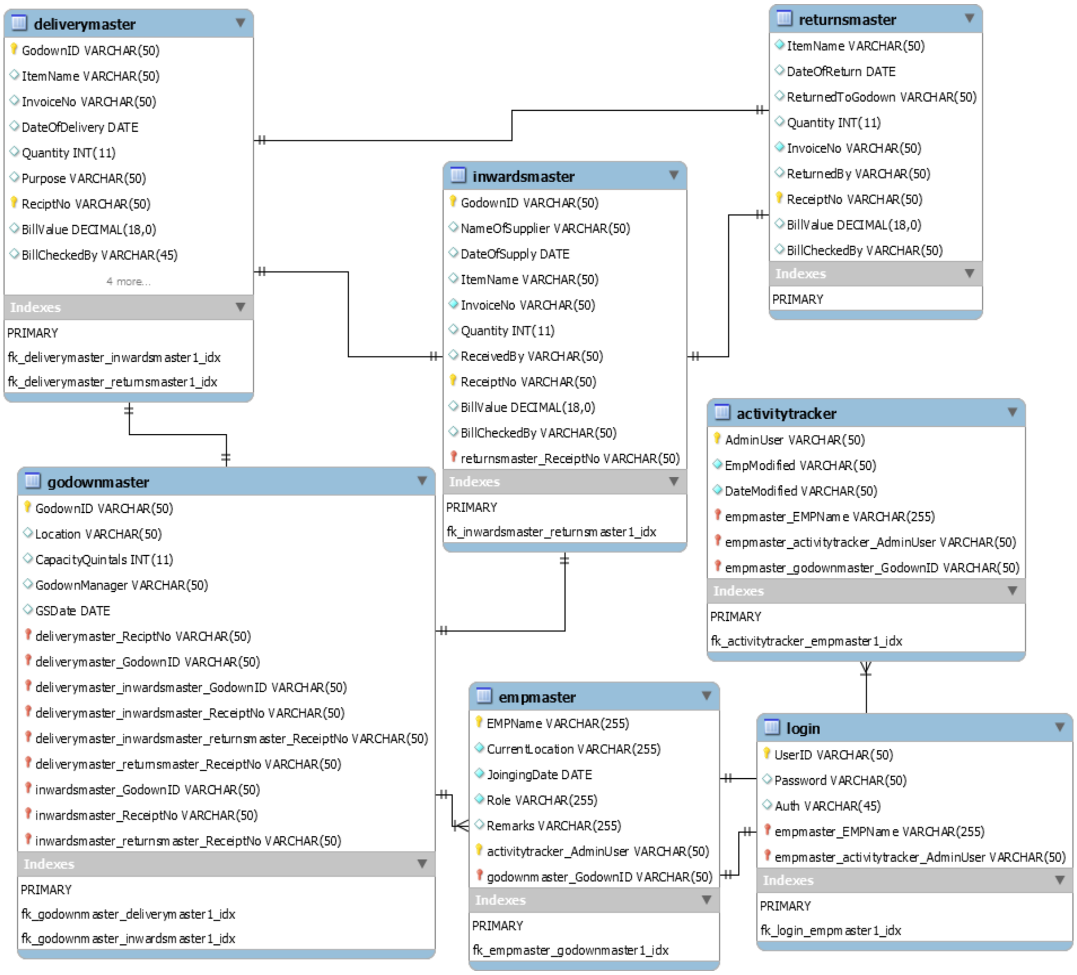
The scope of this system to allow the stock management departments to manage gowdowns, inwards info, delivers info, order cancellation info, damages info and generate the reports dynamically by updating the info very effectively with user friendly screens.

6 Diagrams

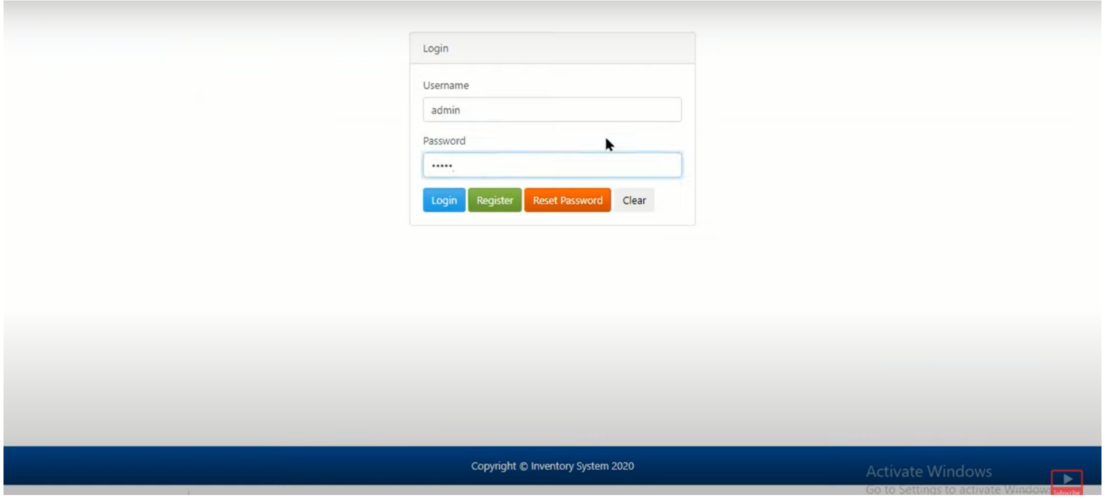
6.1 Class Diagram:



6.2 ER Diagram:


deliverymaster
GodownID VARCHAR(50)
ItemName VARCHAR(50)
InvoiceNo VARCHAR(50)
DateOfDelivery DATE
Quantity INT(11)
Purpose VARCHAR(50)
ReciptNo VARCHAR(50)
BillValue DECIMAL(18,0)
BillCheckedBy VARCHAR(45)
4 more...
Indexes
PRIMARY
fk_deliverymaster_inwardsmaster1_idx
fk_deliverymaster_returnsmaster1_idx
returnsmaster
ItemName VARCHAR(50)
DateOfReturn DATE
ReturnedToGodown VARCHAR(50)
Quantity INT(11)
InvoiceNo VARCHAR(50)
ReturnedBy VARCHAR(50)
ReceiptNo VARCHAR(50)
BillValue DECIMAL(18,0)
BillCheckedBy VARCHAR(50)
Indexes
PRIMARY
inwardsmaster
GodownID VARCHAR(50)
NameOfSupplier VARCHAR(50)
DateOfSupply DATE
ItemName VARCHAR(50)
InvoiceNo VARCHAR(50)
Quantity INT(11)
ReceivedBy VARCHAR(50)
ReceiptNo VARCHAR(50)
BillValue DECIMAL(18,0)
BillCheckedBy VARCHAR(50)
returnsmaster_ReceiptNo VARCHAR(50)
Indexes
PRIMARY
fk_inwardsmaster_returnsmaster1_idx
activitytracker
AdminUser VARCHAR(50)
EmpModified VARCHAR(50)
DateModified VARCHAR(50)
empmaster_EMPName VARCHAR(255)
empmaster_activitytracker_AdminUser VARCHAR(50)
empmaster_godownmaster_GodownID VARCHAR(50)
Indexes
PRIMARY
fk_activitytracker_empmaster1_idx
godownmaster
GodownID VARCHAR(50)
Location VARCHAR(50)
CapacityQuintals INT(11)
GodownManager VARCHAR(50)
GSDate DATE
deliverymaster_ReciptNo VARCHAR(50)
deliverymaster_GodownID VARCHAR(50)
deliverymaster_inwardsmaster_GodownID VARCHAR(50)
deliverymaster_inwardsmaster_ReceiptNo VARCHAR(50)
deliverymaster_inwardsmaster_returnsmaster_ReceiptNo VARCHAR(50)
deliverymaster_returnsmaster_ReceiptNo VARCHAR(50)
inwardsmaster_GodownID VARCHAR(50)
inwardsmaster_ReceiptNo VARCHAR(50)
inwardsmaster_returnsmaster_ReceiptNo VARCHAR(50)
Indexes
PRIMARY
fk_godownmaster_deliverymaster1_idx
fk_godownmaster_inwardsmaster1_idx
empmaster
EMPName VARCHAR(255)
CurrentLocation VARCHAR(255)
JoingingDate DATE
Role VARCHAR(255)
Remarks VARCHAR(255)
activitytracker_AdminUser VARCHAR(50)
godownmaster_GodownID VARCHAR(50)
Indexes
PRIMARY
fk_empmaster_godownmaster1_idx
login
UserID VARCHAR(50)
Password VARCHAR(50)
Auth VARCHAR(45)
empmaster_EMPName VARCHAR(255)
empmaster_activitytracker_AdminUser VARCHAR(50)
Indexes
PRIMARY
fk_login_empmaster1_idx

7 Screens

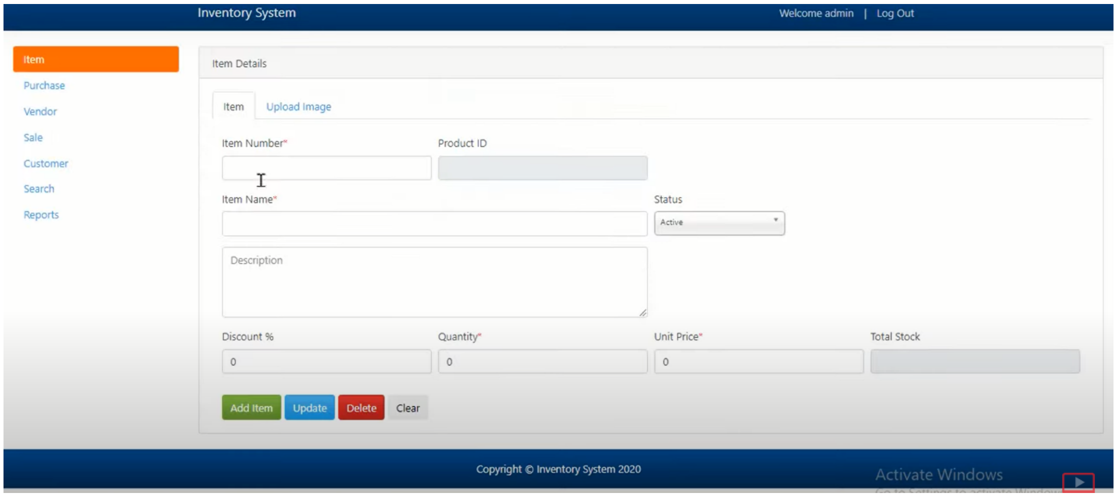


Login
Login
Register
Reset Password
Clear
Copyright © Inventory System 2020
Activate Windows
Go to Settings to activate Windows.

Username: admin

Password: •••••

- Login as admin with wrong username or password



Item Number*:

Product ID:

Item Name*:

Status: Active

Description

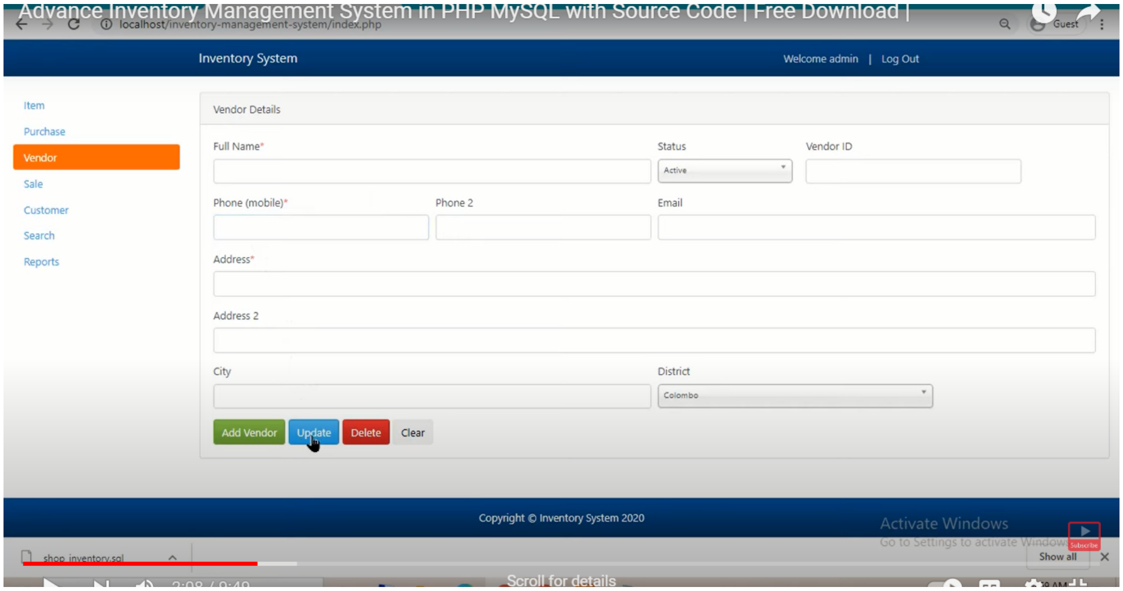
Discount %: 0

Quantity*: 0

Unit Price*: 0

Total Stock:

- Homepage will be displayed like this.



Advance Inventory Management System in PHP MySQL with Source Code | Free Download |

localhost/inventory-management-system/index.php

Inventory System

Welcome admin | Log Out

Vendor Details

Full Name*

Status: Active

Vendor ID

Phone (mobile)*

Phone 2

Email

Address*

Address 2

City

District: Colombo

Copyright © Inventory System 2020

- Select view employees option from Employees menu.
- List of employees will be displayed like this.

8 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.

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:

8.1 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:

i) 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:

a) Incorrect or missing functions
b) Interface errors
c) Errors in data structure or external database access
d) Performance errors
e) Initialization and termination errors.

In this testing only the output is checked for correctness.

The logical flow of the data is not checked.

ii) 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:

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

8.2 Integrating Testing:

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

8.3 System Testing:

Involves in-house testing of the entire system before delivery to the user. It's aim is to satisfy the user the system meets all requirements of the client's specifications.

8.4 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.

8.5 Test Approach:

Testing can be done in two ways:

Bottom up approach
Top down approach

8.5.1 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.

8.5.2 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.

8.6 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.

9 Implementation & Maintenance

9.1 User Manual:

9.1.1 Login:

If you (end user) want to enter into the form, then if you are admin user then you should enter through login form which checks for authorized access . If you are normal user then you to need to be created by the administrator then only this user will be allowed to start the operations. Adding the user details through registration form with your own identification name and password which gives you an unique identification to you and firm.

9.1.2 Godown Management:

Then the administrator can create a new godown in which the normal user can save transaction details regarding inwards, deliveries and returns. Whenever the transaction goes on then those details will be updated in the centralized database which helps in generating the updated reports very effectively.

9.1.3 Employees & User Management:

An administrator can create, edit and delete the employees information. He can also create a user of type administrator or a normal employee.

10 Conclusion

The efficiency of any system designed to suit an organization depends cooperation during the implementation stage and also flexibility of the system to adopt itself to the organization.

“Stock Analyzer” has been developed to overcome the problems with traditional stock management in large scale. Advantages over traditional manual systems are online application access through out all the godowns from the same location, reducing the manual work, storage the data at a secured centralized locations and quick generation of reports as per our requirements.

11 References

- Java jdk Downloads | Oracle. 2015. Downloads | Oracle. [ONLINE] Available at: https://www.oracle.com/downloads/index.html

- Java Developer Tutorials and Online Training. 2015. [ONLINE] Available at: http://www.oracle.com/technetwork/java/index-jsp-135888.html

- SQL Tutorial. 2015. [ONLINE] Available at: http://www.w3schools.com/sql/default.asp

- MySQL Downloads. 2015. [ONLINE] Available at: http://www.mysql.com/downloads/

- Apache Tomcat - Apache Tomcat 7 Downloads. 2015. [ONLINE] Available at: https://tomcat.apache.org/download-70.cgi

- Netbeans downloads 2015. [ONLINE] Available at: https://netbeans.org/downloads/