

# **STOCK ANALYSER**

A Project Report submitted in the partial fulfillment of the requirements for the  
Award of the degree of

## **BACHELOR OF TECHNOLOGY**

In

### **COMPUTER SCIENCE AND ENGINEERING**

Submitted by

**VASISTA AVINASH B (13471A0567)**

Under the esteemed guidance of

**Mr. G.Raphi M.Tech,Assisitant Profssor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA**  
**(Affiliated to J.N.T.U, Kakinada and Approved by AICTE & Accredited by NBA)**

**2016-2017**

**NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled "**STOCK ANALYSER**" is bonafide work done by "**VASISTA AVINASH B (13471A0567)**" in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in the Department of COMPUTER SCIENCE AND ENGINEERING during 2016-2017.

**PROJECT GUIDE**

**G. Raphi** M.Tech, Assistant Professor

**PROJECT COORDINATOR**

**S. Siva Nageswara Rao** M.Tech.,(Ph.D.)

**HEAD OF THE DEPARTMENT**

**Dr. S.N. Tirumala Rao** M.Tech., Ph.D.

**EXTERNAL EXAMINER**

## **ACKNOWLEDGEMENT**

I wish to express my thanks to various personalities who are responsible for the completion of the project. I extremely thankful to our Chairman **Sri M. V. Koteswara Rao** B.Sc. who took keen interest on us in every effort throughout this course. I owe out gratitude to our principal **Dr. B.V. Rama Mohana Rao** M.Tech., Ph.D. for his kind attention and valuable guidance throughout the course.

I extend my sincere thanks to **Dr. S. N. Tirumala Rao** M.Tech., Ph.D., Professor and Head, Department of Computer Science & Engineering, who supervised, guided and helped us at every problem in this project.

I highly thankful to our Project Guide **Mr. G. Raphi** B.Tech, M.Tech, and Coordinator **Mr. S. Siva Nageswara Rao** M.Tech., (Ph.D.) Associate Professor in Department of Computer Science & Engineering, whose valuable guidance helped us to understand the project better.

I extend my sincere thanks to all other teaching and non-teaching staff of the Department for their cooperation and encouragement during my B.Tech course. I have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

I affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarified my doubts which had really helped us in successfully completing our Project.

**By:**

**Vasista Avinash B (13471A0567)**

## **ABSTRACT**

Stock Analyser is an online software application that fulfils 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. It 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. In addition, this application provides an interface to users to view the details like the daily Stock Statements of all godowns.

# **INDEX**

<b>S. No</b>	<b>CONTENTS</b>	<b>PAGE NO</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
	1.1    Project Plan	1
	1.2    Requirements Specification Document	2
	1.3    Functional Requirements	2
	1.4    Non Functional Requirements	4
	1.5    Validation Criteria	4
	1.6    System and Software Architecture	5
	1.7    Business Requirement Analysis	6
<b>2</b>	<b>Problem Definition</b>	<b>6</b>
	2.1    Existing System and its Disadvantages	8
	2.2    Need for new System	8
	2.3    Proposed System and its Advantages	9
<b>3</b>	<b>Design Document</b>	<b>10</b>
	3.1    Software Requirements	10
	3.2    Hardware Components	10
	3.3    Application	10
	3.4    Database Design	11
	3.4.1    Constraints	11
	3.4.2    Normalization of Database	13
	3.4.3    First Normal Form	13
	3.4.4    Second Normal Form	13
	3.4.5    Third Normal Form	13
	3.4.6    Database Tables	14
	3.5 Database Design Tables	15
<b>4</b>	<b>System Analyses</b>	<b>18</b>
	4.1    System Design	18
	4.2    UML Diagrams	19

<b>S. No</b>	<b>CONTENTS</b>	<b>PAGE NO</b>
4.3	Modules	20
4.3.1	User Information Module	20
4.3.2	Inwards module	20
4.3.3	Deliveries Module	21
4.3.4	Returns Module	21
4.3.5	Admin Module	21
5	<b>Software Requirement Specification</b>	22
6	<b>Design Diagrams</b>	24
7	<b>Implementation</b>	27
8	<b>Testing and Test Cases</b>	49
8.1	Testing Survey	49
8.2	Types of Testing	49
8.2.1	Unit Testing	49
8.2.2	Integration testing	49
8.2.3	System Testing	50
8.2.4	Acceptance Testing	51
8.3	Test Cases	52
9	<b>Output Screens</b>	56
10	<b>Conclusion</b>	68
11	<b>Future Enhancement</b>	69
12	<b>Bibliography</b>	70

# **1. INTRODUCTION**

Stock Analyser is an online software application that fulfils 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. In addition, this application provides an interface to users to view the details like the daily Stock Statements of all godowns.

## **1.1 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 modelling language to express the Analysis & Design of the System. According to Grady Booch et al “The Unified Modelling Language (UML) is a graphical language for visualising, 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, 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.

## **1.2 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

## **1.3 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. Whenever an inwards entry is entered then accordingly the stock number will be automatically updated.
10. Whenever 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. Whenever 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.
13. It allows the users to change their password for future security.
14. 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.
15. It allows the admin to view the list of users and take the print.
16. It allows admin to generate godown details report.
17. It allows admin to generate inwards details report.
18. It allows admin to generate outwards details report.
19. It allows admin to generate returns details report.
20. It allows admin to generate stock statement report.

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

## **1.4 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.

## **1.5 Validation Criteria**

### **1.5.1 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.

### **1.5.2 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.

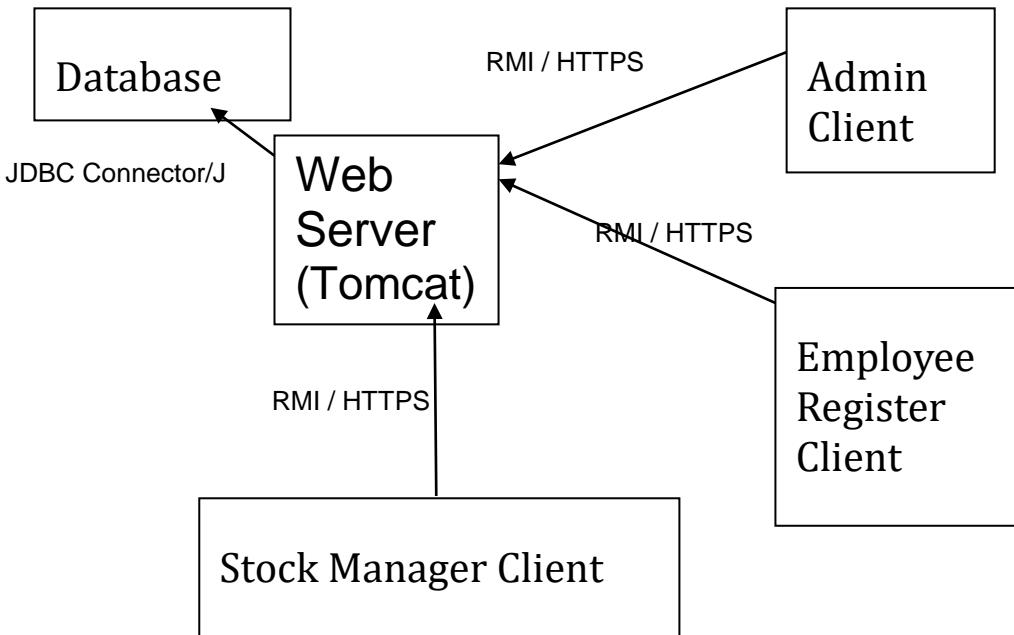
8. An inward entry should be entered in the database whenever stock comes into the godown. That is the number of items should be updated automatically.
9. An outward entry should be entered in the database whenever stock goes out into the godown. That is the number of items should be updated automatically.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

## **1.6 System and Software Architecture**

The programming language for the *stock Management System* application will be in Java since the JSP/Tomcat architecture will be used. Both types of clients will communicate with the server using Java Remote Method Invocation (RMI) with Secure Socket Layer (SSL). Thus, security will be upheld by HTTPS in both a private network and a network connected to the Internet. The application will use a client-server model. *See Figure 3.*

Furthermore, the Java Database Connectivity (JDBC) will use the MySQL Connector/J driver for the server to communicate to the inventory database. Upon receiving requests from the clients, the server will issue transactions to the MySQL database with ACID properties. The Cash Register clients will maintain local log files in the event of communication downtime between Cash Register clients and the server. Upon re-establishing communication, the Cash Register clients will re-synchronize with the server by issuing the requests that occurred during downtime.

Alternatively, if the Microsoft ASP.NET framework is used, then the programming language for the application will be Microsoft C# with a Microsoft SQL Server database. The Microsoft .NET framework includes built-ins for easy database interfacing and .NET Framework Remoting (analogous to skeleton and stub), which has the ability to also use HTTPS.



## 1.7 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 ins and outs of all godowns. The ins and outs 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 godowns like godown id, godown location, and 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 godown 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, Quantity, invoice no,

returned by, receipt no, bill value and checked by needs to store 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.

## **2. PROBLEM DEFINITION**

### **2.1 Existing System**

- Paper-based systems originated as a system where details are entered and retrieved by hand
- 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.
- Regular investments need to purchase stationary every year.

#### **Disadvantages:**

- Chances of hacking user or admins account. Security can be implemented by adding extra pattern matching and database securities.
- Daily transactions are to be entering into different books immediately to avoid conflicts which are very difficult.

### **2.2 Need for new System**

- Easy to manage large amount of user and data store in electronic for long time.
- Access Database from anywhere
- The project is mainly aimed at providing employee friendly and effortless godown Stock management system.
- No paper work
- Display transaction results in tables and instantly display result for the administrator and user to analyse.
- Prevent users from manipulating other employee details.

## **2.3 Proposed System**

- Proposed system is a software application that avoids more manual hours that need to spend in record keeping and generating reports.
- This application keeps the data in a centralized way that is available to all the users simultaneously.
- 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.

### **Advantages:**

- Fast and easy way of managing godowns.
- Admin can manage all the details.
- Unique ID and Password gives more Secure Logins.
- Stock Statements are generated fast.

### **3. DESIGN DOCUMENT**

#### **3.1 Software Requirements:**

- Web Browser - Google Chrome any version,  
Internet Explorer 4.2 or above.
- OS - WIN XP, 7, 8, 8.1, 10.
- Frontend - HTML, CSS.
- Backend - JAVA (JSP), JS.
- IDE - Eclipse
- Server - Apache Tomcat
- JAVA Version - JDK 1.7 or above.
- Database - MySQL or any equivalent.

#### **3.2 Hardware Components:**

- Processor - Pentium IV 2.0 or above.
- Hard Disk - 50 GB
- Memory - 512MB RAM

#### **3.3 Application:**

This system can be used by any product based Organisation to manage their stocks and godowns all over the country.

## **3.4 Database Design**

### **3.4.1 Constraints:**

- NOT NULL
- UNIQUE
- PRIMARY KEY / CANDIDATE KEY
- FOREIGN KEY
- CHECK
- DEFAULT CONSTRAINT

#### **Not Null:**

- Not Null constraint is used to avoid null values for a specific column.
- Not Null should be defined only at column level.

#### **Unique:**

- UNIQUE constraint ensures that a field or column will only have unique values.
- A UNIQUE constraint field will not have duplicate data.
- UNIQUE constraint can be applied at column level or table level.

**Key:** A key is an attribute or set of attributes in a relation that identifies a tuple in a relation.

#### **Primary Key:**

- Primary key constraint uniquely identifies each record in a database.
- Primary key doesn't allow duplicates and null values (combination of UNIQUE and NOT NULL).
- Usually Primary Key is used to index the data inside the table.
- Only one primary key is allowed per table.

## **Candidate Key:**

- A candidate key is a column, or set of columns, in a table that can uniquely identify any dB record without referring to any other data.
- Each table may have one or more candidate keys, but one candidate key is special, and it is called the primary key. This is usually best among the candidate keys.
- When a key is composed of more than one column, it is known as a composite key.

## **Foreign Key:**

- Foreign keys are the columns of a table that points to the primary key of another table. They acts as a cross-reference between tables.
- The foreign key constraint provides referential integrity rules (either within table or b/w tables) i.e., we can place a value in TABLE B if the values exist as a primary key in TABLE A.
- We use foreign key if we want to ensure that for every child table record there is a reference in parent table.

## **Check:**

- CHECK constraint is used to restrict the value of a column between ranges.
- It performs check on the values, before storing them into the database.
- It is like condition checking before saving data into a column.
- The check constraint can be defined at the column level or table level.

## **Default Constraint:**

- The DEFAULT constraint is used to insert a default value into a column.
- The default value will be added to all new records, if no other value is specified.
- The default value can be a literal, an expression, or sql function.

### **3.4.2 Normalization of Database**

If a database design is not perfect it may contain anomalies, which leads to inconsistency of database itself. Normalization is the process of efficiently organizing data in the DB.

There are two goals of the normalization process:

1. Eliminate redundant data( for example, storing the same data in more than one table ) and
2. Ensure data dependencies make sense (only storing related data in a table).

Both of these worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

### **3.4.3 First Normal Form (1NF):**

A relation is said to be in 1NF if it contains no non-atomic values and each row can provide a unique combination of values.

### **3.4.4 Second Normal Form (2NF):**

A relation is said to be in 2NF if it is already in 1NF and each and every attribute fully depends on the primary key of the relation. Speaking inversely, if a table has some attributes which is not dependent on the primary key of that table, then it is not in 2NF.

Full dependency: Non-key attributes are depended on key attributes.

Partial dependency: Non-key attributes are depended on part of key attributes.

### **3.4.5 Third Normal Form (3NF):**

A relation is said to be in 3NF, if it is already in 2NF and there exists no transitive dependency in that relation. Speaking inversely, if a table

contains transitive dependency, then it is not in 3NF, and the table must be split to bring it in 3NF.

### **Advantages of Normalization:**

- Avoid redundant fields or columns.
- Better understanding of data.
- Ensure that distinct tables exists when necessary.

### **Disadvantages of normalization:**

- You cannot start building the DB before you know what the user needs.
- It is very time consuming & difficult process in normalizing relations of high degree.

On normalizing the relations to higher normal forms i.e. 4NF, 5NF the performance degrades.

### **3.4.6 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.

### **3.5 DATABASE DESIGN (TABLES):**

#### **1. Table name: godown**

**Description:** To manage godown details.

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	GodownID	Number(10)	Primary key	Godown identification
2.	Location	Varchar2(15)	Not Null	Godown location
3.	CapacityInQunitals	Number(5)	Not Null	Capacity in quintal
4.	GManager	Varchar2(20)	Not Null	Godown manager
5.	GSDate	Date	Not Null	Godown start date
5.	StockInQunitals	Number(5)	Not Null	Stock in quintals

#### **2. Table name: Emp**

**Description:** user details

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	EmpName	Varshar2(20)	Not Null	Name of employee
2.	EmpID	Number(10)	P Key ,NN, Unique	Employee ID
3.	GodownID	Number(10)	Foreign Key	Work location
4.	Role	Varchar2(15)	Not Null	Designation
5.	JoiningDate	Date	Not Null	Date of join
6.	Password	Varchar2(45)	Not Null	Password of emp

#### **3. Table name: Inwards**

**Description:** details of stocks received

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	GodownID	Number(10)	Foreign key	Godown id
2.	SID	Number(10)	Foreign key	ID of supplier
3.	ItemID	Number(10)	Foreign Key	Item ID
4.	DateOfSupply	Date	Not Null	Item supply date
5.	Quantity	Number(5)	Not Null	Quantity received
6.	ReceiptNo	Number(10)	Not Null	Receipt generated
7.	REmpID	Number(10)	Foreign Key	Employee who received
8.	Bill	Number(10)	Not Null	Bill generated
9.	CEmpID	Number(10)	Foreign Key	Employee who checked bill
10.	Invoice	Number(10)	Primary key	Inward supply details

#### 4. Table name: delivery

**Description:** details of stocks delivered

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	GodownID	Number(5)	Foreign key	Godown id
2.	ItemID	Varchar2(20)	Foreign Key	Item id
3.	DateOfSupply	date	Not Null	Item supplied date
4.	Invoice	BIGINT(20)	Primary key	delivery details
5.	DateOfDelivery	Date	Not Null	Item delivery date
6.	Quantity	Number(10)	Not Null	Quantity delivered
7.	ReceiptNo	Number(10)	Not Null	Receipt generated
8.	CID	Number(10)	Foreign Key	Delivered to customer
9.	Bill	Number(10)	Not Null	Bill generated
10.	EmpID	Number(10)	Foreign Key	Bill checked by employee

#### 5. Table name: return

**Description:** details of stocks returned

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	GodownID	Number(5)	Foreign key	Godown id
2.	ItemID	Number(10)	Foreign key	Item number
3.	DateOfDelivery	Date	Not Null	Item delivered date
4.	Invoice	Number(20)	Primary key	return details
5.	DateOfReturn	Date	Not Null	Item return date
6.	Quantity	Number(10)	Not Null	Quantity returned
7.	Purpose	Varchar2(20)	Not Null	Purpose of return
7.	ReceiptNo	Number(10)	Not Null	Receipt generated
8.	CID	Number(10)	Foreign key	Returned by customer
9.	Bill	Number(10)	Not Null	Bill generated
10.	EmpID	Number(10)	Foreign Key	Bill checked by employee

## 6. Table name: Stock

**Description:** details of stocks in godowns

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1.	GodownID	Number(5)	Primary key	Godown identification
2.	ItemID	Number(10)	Foreign Key	Item ID
3.	InitialStockDate	date	Not Null	Starting stock date
4.	InitialStock	Number(10)	Not Null	Initial stock amount
5.	ReceivedStock	Number(10)	Not Null	Stock received
6.	DeliveredStock	Number(10)	Not Null	Stock delivered
7.	ReturnedStock	Number(10)	Not Null	Stock returned
8.	TotalStock	Number(10)	Not Null	Total stock

## 7. Table name: Item

**Description:** details of items

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1	ItemID	Number(10)	Primary Key	Item ID
2	ItemName	Varchar2(20)	NOT NULL	Item name
3	Itemdetails	Varchar2(20)	NOT NULL	Item Details

## 8. Table name: Supplier

**Description:** details of supplier

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1	SID	Number(10)	Composite PKey	Supplier ID
2	SName	Varchar2(20)	NOT NULL	Supplier Name
3	ItemID	Number(10)	Composite PKey	Item of supplier

## 9. Table name: Customer

**Description:** details of Customer

S.NO	ATTRIBUTE	DATA TYPE	CONSTRAINT	DESCRIPTION
1	CID	Number(10)	Composite PKey	Customer ID
2	CName	Varchar2(20)	NOT NULL	Customer Name
3	ItemID	Number(10)	Composite PKey	Item of Customer

## **4 SYSTEM ANALASES**

### **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 new suppliers, customers, items
- 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**

### **UML (UNIFIED MODELLING LANGUAGE)**

The unified modelling 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.

#### **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.

#### **Specifying:**

Specifying means building models that are precise, unambiguous and complete. UML 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.

#### **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, and Source code, Project plans, Tests, Prototypes, Releases etc.

## **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.

## **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 USER 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. 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.

#### **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 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 - employee

Apart, there are suppliers and customers who cannot access anything.

## **5 SOFTWARE REQUIREMENT SPECIFICATION**

### **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.

### **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.

### **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.

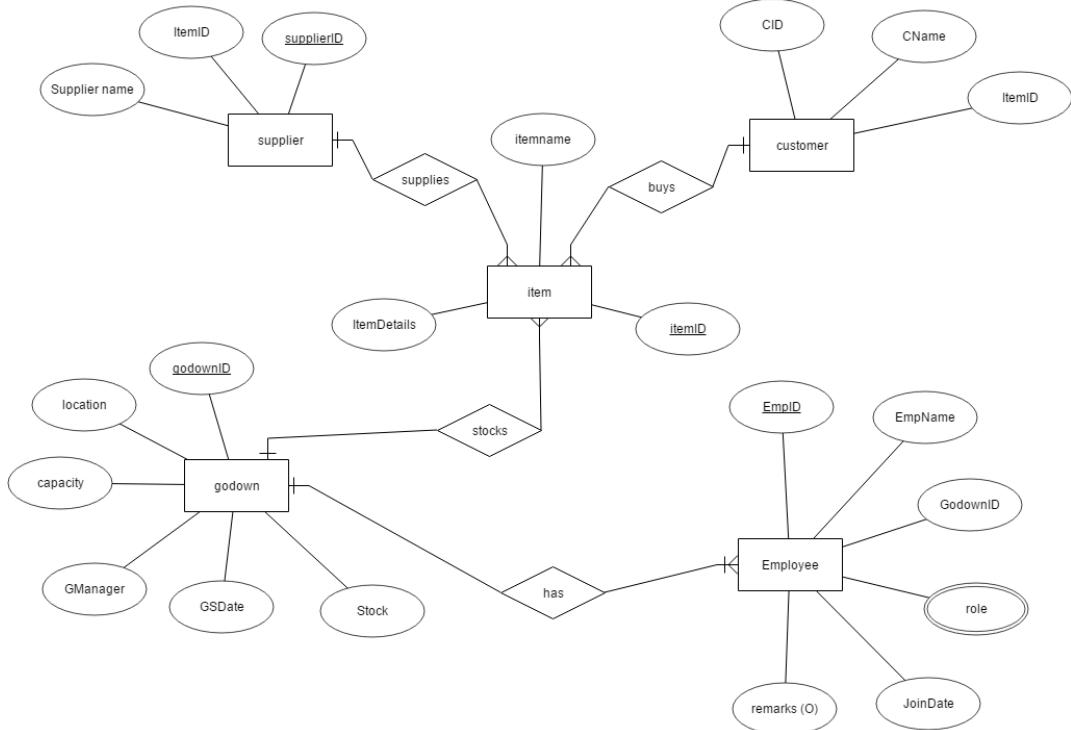
### **Scope:**

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 be the basis for

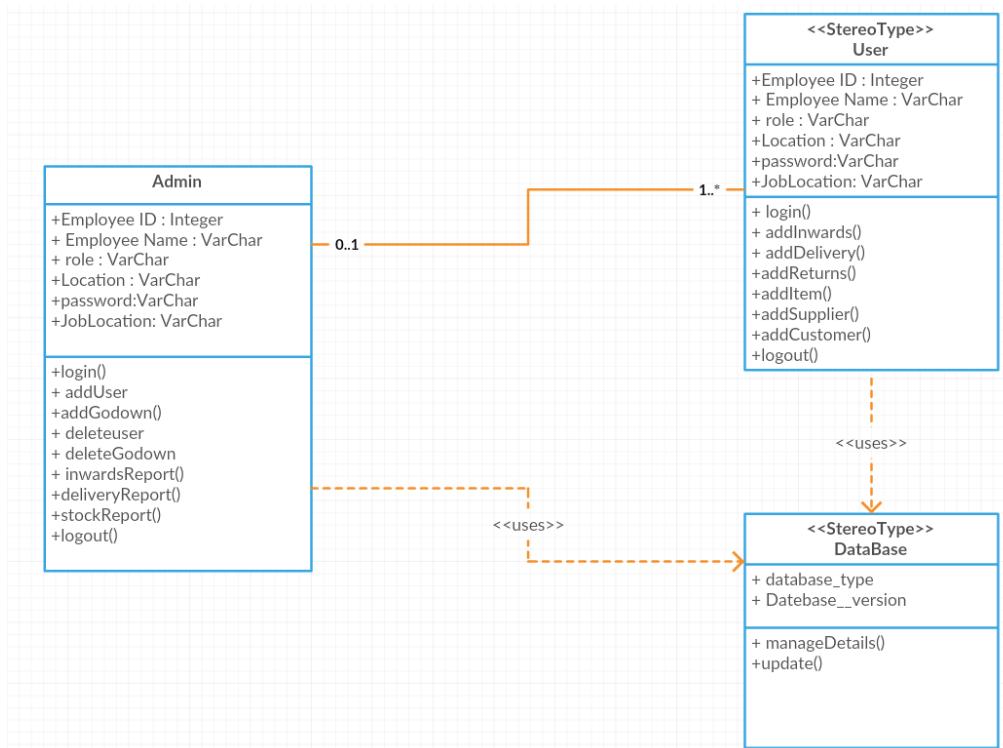
validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client. The scope of this system to allow the stock management departments to manage godowns, 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 Design Diagrams

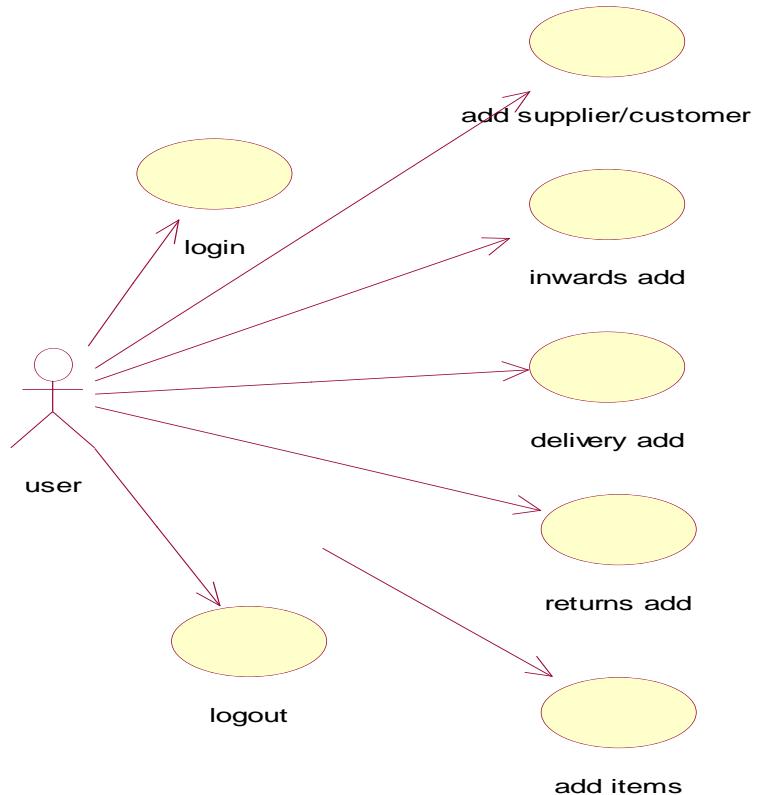
### ER Diagram:



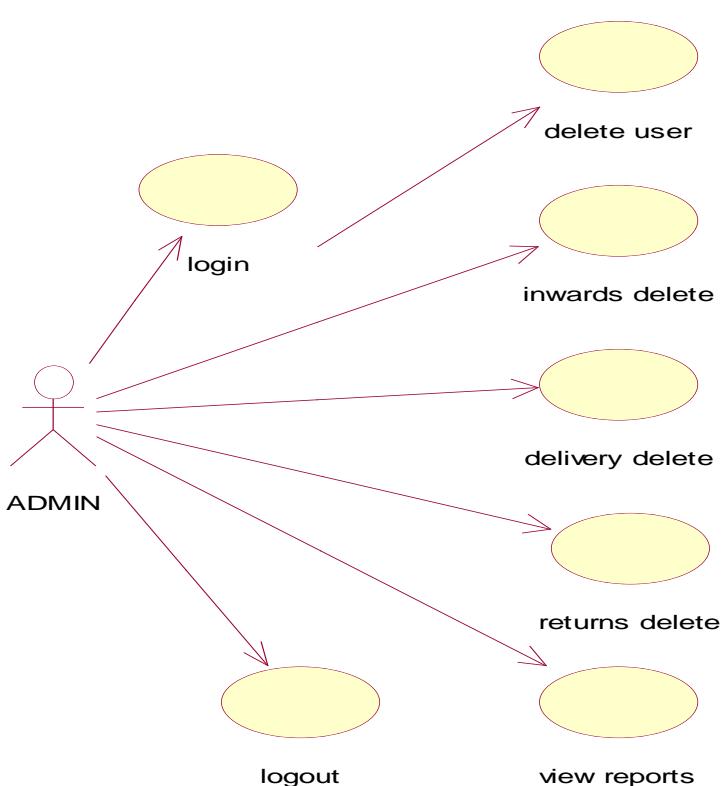
### Class Diagram:



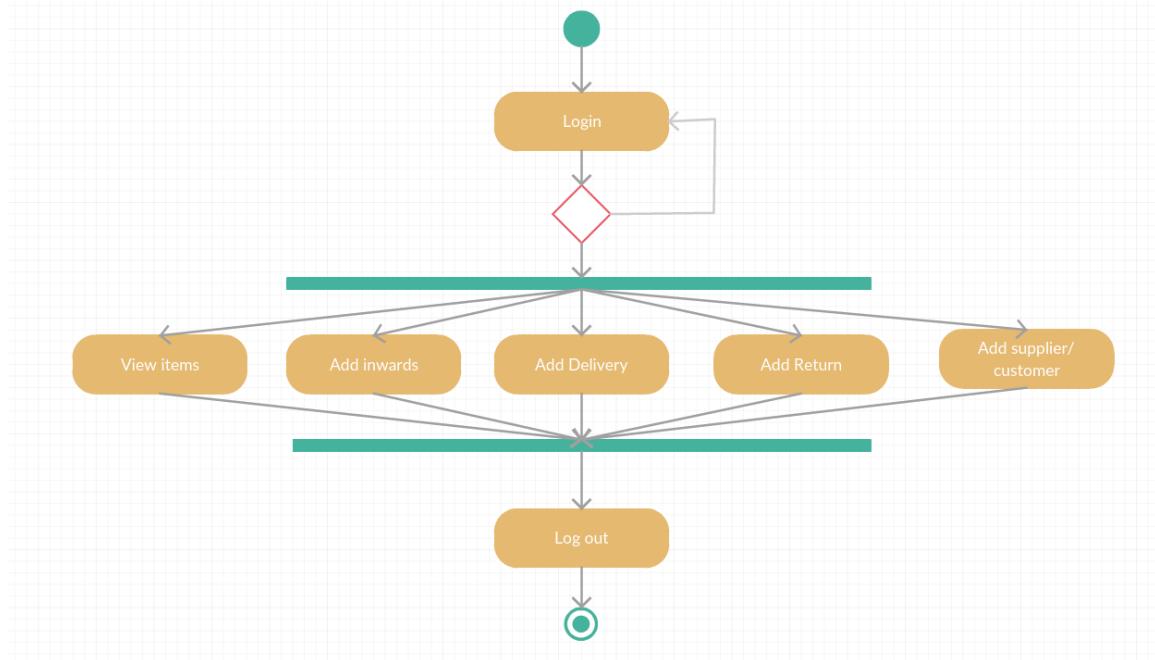
### Use Case Diagram (User):



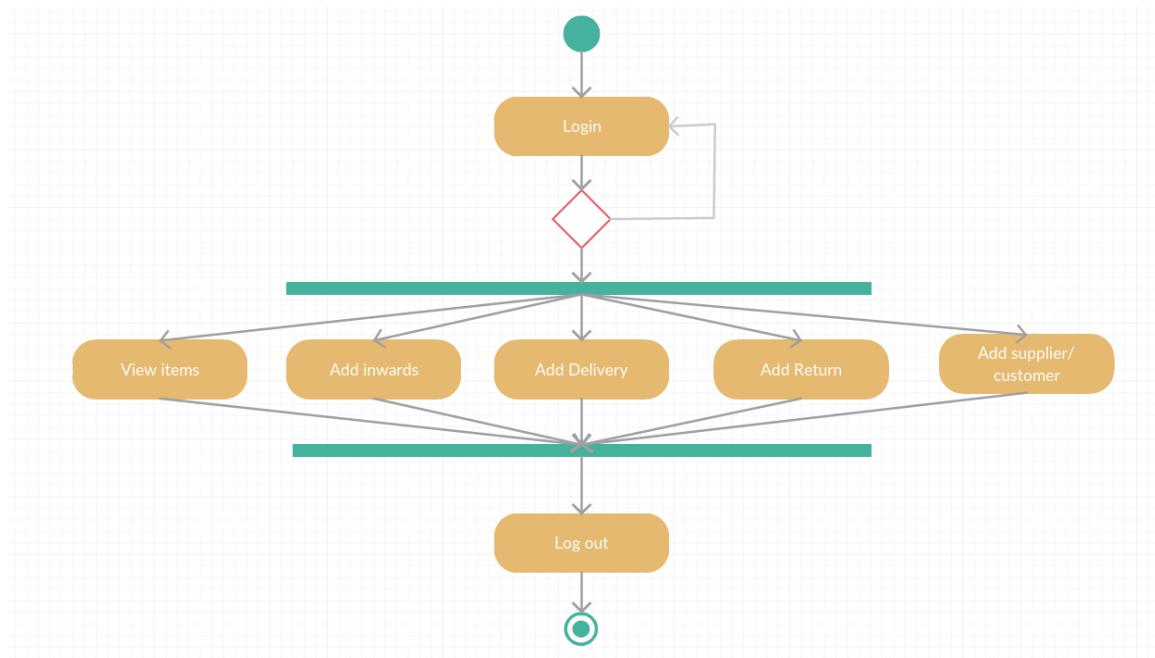
### Use case Diagram (Admin):



## Activity Diagram (Admin):



## Activity Diagram (User):



## 7. IMPLEMENTATION

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Register</title>

</head>

<body>
    <ul class="topnav">
        <li><a href="index.jsp">Home</a></li>
        <li><a href="Login.html">Login</a></li>
        <li><a class="active"
href="Register.jsp">Register</a></li>
        <li><a href="about.jsp">About</a></li>
    </ul>
    <div class="content">
        <font face="Tw Cen MT" color="1234"><center>
            <h1>Register Here</h1>
        </center>
        <form align="center" action="Register.jsp">
            <input type="text" name="empID"
pattern="[1-9][0-9]{4}" title="5 Digit employee ID"
placeholder="Enter your employee ID" required /><br>
            <input type="text" name="empName"
pattern="[A-Za-z].{3,}" title="atleast 3 letters" placeholder="Enter
your Full Name" /><br>
            Gender: <input
type="radio" name="gender"
value="male" checked> Male <input
type="radio" name="gender"
value="female"> Female<br>
            <select name="JLocation">
                <option>Select your Job
Location</option>
                <option value="hyd">hyd</option>
                <option
value="mumbai">mumbai</option>
                <option value="delhi">delhi</option>
                <option
value="chennai">chennai</option>
                <option
value="banglore">banglore</option>
            </select><br> Join Date : <input
type="date" name="JDate" /><br>
            <select
name="role" required>
                <option>Select your Role</option>
                <option
value="manager">manager</option>
                <option value="clerk">clerk</option>
                <option
value="accountant">accountant</option>
                <option
value="security">security</option>
                <option value="other">Other</option>
            </select><br>
            <input type="password"
name="password" id="password" pattern=".{5,}" title="atleast 5
characters" placeholder="Enter Password" required />
        </form>
    </div>
</body>
```

```

        <input type="password"
name="rpassword" id="rpassword" pattern=".{5,}" title="atleast 5
characters" placeholder="Re Enter Password"
onkeyup="validatePassword()" required /><br>
        <input type="submit" name="register"
value="Register" /> <input type="reset" value="cancel" />
    </form>
    <style>
</style>

</style>

<center>
<%@page language="java"
contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@page import="java.sql.* , java.util.*"%>
<%
String empID =
request.getParameter("empID");
String empName =
request.getParameter("empName");
String gender =
request.getParameter("gender");
String JLocation =
request.getParameter("JLocation");
String role =
request.getParameter("role");
String password =
request.getParameter("password");
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root",
"2513218");
Statement st =
con.createStatement();
String query = "insert into
emp(empID,empName,gender,JLocation,role,password) values ('" + empID
+ "','" +
+ empName +
"', '" + gender + "','" + JLocation + "','" + role + "','" + password
+ "')";
System.out.println(query);
int i =
st.executeUpdate(query);
out.println("Successfully
Registered your details");
%>
<a href="Login.html">Login here...</a>
<%
} catch (Exception e) {
System.out.print(e);
e.printStackTrace();
}
%>
</center> </font>
</div>
<script type="text/javascript">
function mouseoverPass(obj) {
    var obj = document.getElementById('Password');
    obj.type = "text";
}

```

```

        function mouseoutPass(obj) {
            var obj = document.getElementById('Password');
            obj.type = "password";
        }
    </script>
</body>
<footer>
    <small>&copy; Copyright 2017, Stock Analyser NEC</small>
</footer>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>godown Add</title>
</head>
<body onload="history.forward(1)">
    <ul class="sidenav">
        <li><a class="active" href="godownAdd.jsp">godown
add</a></li>
        <li><a href="godownlist.jsp">List Godown</a></li>
        <li><a href="godownView.jsp">View Godown</a></li>
        <li><a href="godownDelete.jsp">Delete Godown</a>
        <li><a href="user.jsp">Back</a></li>
        <li class="logout"><a href="Logout.jsp">Logout</a></li>
    </ul>
    <div class="content">
        <font face="Tw Cen MT" color="1234"><center>
        <%
            String admin =
(String)session.getAttribute("userid");
            if ((session.getAttribute("userid") ==
null) || (session.getAttribute("userid") == ""))
                %
            You are not logged in<br /> <a
href="Login.html">Please Login</a>
            <%
                }
            else {
                %
        <h1 align="center">
            <u>Add New Godown</u>
        </h1>
        </center>
        <form align="center" action="godownAdd.jsp">
            <input type="text" name="godownID"
placeholder="Enter the godown ID"
                    required /> <input type="text"
name="GLocation"
                    placeholder="Location of your
godown" required />
            <%
                Connection con1 = null;
                Statement st = null;
                try {

```

```

        Class.forName("com.mysql.jdbc.Driver");
        con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
                st = con1.createStatement();
                ResultSet rs =
st.executeQuery("Select empName from emp where role='manager'");
                %>
                <select name="items">
                    <option>select godown
Manager</option>
                <%
                while (rs.next()) {
                    String GManager
= rs.getString("empName");
                    %>
                    <option
value=<%out.println(GManager);%>>
                    <%
                    out.println(GManager);
                    %>
                    </option>
                    <%
                }
                } catch (Exception e) {
                    System.out.print(e);
                    e.printStackTrace();
                }
                %>
                </select> Godown Start Date <input
type="date" name="GSDate" required /> <input
                    type="text" name="GCapacity"
placeholder="Capacity in quintals"
                    required /> <input type="text"
name="GStock"
                    placeholder="Present Stock in
godown" required /> <input
                    type="submit" value="submit" />
<input type="reset" value="cancel" />
                </form>
            </font>
            <%
            String godownID =
request.getParameter("godownID");
            String GLocation =
request.getParameter("GLocation");
            String GCapacity =
request.getParameter("GCapacity");
            String GManager =
request.getParameter("GManager");
            String GSDate = request.getParameter("GSDate");
            String GStock = request.getParameter("GStock");
            con1 = null;
            st = null;
            try {
                Class.forName("com.mysql.jdbc.Driver");
                con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
                st = con1.createStatement();

```

```

        String query = "insert into
godown(godownID,GLocation,GManager,GSDate,GCapacity,GStock) values
('"
                    + godownID + "','" +
GLocation + "','" + GManager + "','" + GSDate + "','" + GCapacity + 
"',"
                    + GStock + "')";
System.out.println(query);
int i = st.executeUpdate(query);
out.println("Successfully Registered your
details");
} catch (Exception e) {
    System.out.print(e);
    e.printStackTrace();
}
}
%>
<style>
form {
    width: 300px;
    margin: 0 auto;
    padding: 10px;
}

input[type=text], select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

input[type=submit] {
    width: 45%;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type=reset] {
    width: 45%;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type=submit]:hover {
    background-color: #111;
}

input[type=reset]:hover {
    background-color: #111;
}

```

```

body {
    margin: 0;
    background-image:
        url("images/yellow-gold-scratched-paint-wall-
background.jpg");
    background-size: cover;
    width: 100%;
    height: 400px;
}

ul.sidenav {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 10%;
    background-color: #333;
    position: fixed;
    height: 100%;
    overflow: hidden;
}

ul.sidenav li a {
    display: block;
    color: #fff;
    padding: 14px 16px;
    text-decoration: none;
}

ul.sidenav li a.active {
    background-color: #4CAF50;
    color: white;
}

ul.sidenav li a:hover {
    background-color: #111;
}

div.content {
    margin-left: 10%;
    padding: 1px 16px;
}

@media screen and (max-width: 1050px) {
    ul.sidenav {
        width: 100%;
        height: auto;
        position: relative;
    }
    ul.sidenav li a {
        float: left;
        padding: 15px;
    }
    div.content {
        margin-left: 0;
    }
}

@media screen and (max-width: 400px) {
    ul.sidenav li a {
        text-align: center;
        float: none;
    }
}

```

```

        }
    footer {
        position: fixed;
        bottom:0;
        color: white;
        margin-left: 43%;
    }

```

```

    </style>
    </div>
</body>
<footer>
    <small>&copy; Copyright 2017, Stock Analyser NEC</small>
</footer>
</html>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>inwards Add </title>
</head>
<body onload="history.forward(1)">
<ul class="sidenav">
    <li><a class="active" href="inwardsAdd.jsp">Add
Inwards</a></li>
    <li><a href="inwardsReport.jsp">Inwards Report</a></li>
    <li><a href="user.jsp">Back</a></li>
    <li class="Logout"><a href="Logout.jsp">Logout</a></li>

```

```
</ul>
<div class="content">
```

```

<font face="Tw Cen MT" color="1234">
<center>
    <form align="center" action="inwardsAdd.jsp">
        <%
            String admin = (String)session.getAttribute("userid");
            if ((session.getAttribute("userid") == null) ||
                (session.getAttribute("userid") == ""))
        %>
        You are not logged in<br /> <a href="Login.html">Please Login</a>
        <%
    }
    else {
        Connection con1=null;
        Statement st=null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
            st=con1.createStatement();
            ResultSet rs = st.executeQuery("Select godownID from godown");
        %>
            <h1 align="center" ><u>Add New inwards</u></h1>
            <select name="godown">
                <option>select godown</option>
            <%
                while(rs.next())
                {
                    String godownID=rs.getString("godownID");

```

```

        %>
        <option
value=<%out.println(godownID);%>><%out.println(godownID);%></option>
        <%
        }
        %>
        </select>
        <%
        rs = st.executeQuery("Select SID from supplier"); %>
        <select name="supplier">
        <option>select supplier</option>
        <%
        while(rs.next())
        {
            String SID=rs.getString("SID");
            %>
            <option
value=<%out.println(SID);%>><%out.println(SID);%></option>

            <%
            }%>
            </select>
            <%
            rs = st.executeQuery("Select itemID from item"); %>
            <select name="item">
            <option>select item</option>
            <%
            while(rs.next())
            {
                String itemID=rs.getString("itemID");
                %>
                <option
value=<%out.println(itemID);%>><%out.println(itemID);%></option>
                <%
                }
                %>
                </select>
                Date of Supply <input type="date" name="dateOfSupply"
required/>
                <input type="text" name="quantity" placeholder="Enter
quantity of stock" required/>
                <input type="text" name="receiptNo" placeholder="Enter
the receipt number" required/>
                <%
                rs = st.executeQuery("Select empID from emp"); %>
                <select name="remp">
                <option>select received employee</option>
                <%
                while(rs.next())
                {
                    String empID=rs.getString("empID");
                    %>
                    <option
value=<%out.println(empID);%>><%out.println(empID);%></option>

                    <%
                    }%>
                    </select>
                    <input type="text" name="bill" placeholder="bill"
required/>
                    <%
                    rs = st.executeQuery("Select empID from emp"); %>
                    <select name="cemp">

```

```

<option>select received employee</option>
<%
while(rs.next())
{
    String empID=rs.getString("empID");
    %>
    <option
value=<%out.println(empID);%><%out.println(empID);%></option>

    <%
}>
</select>
<input type="text" name="invoice" placeholder="invoice"
required/>
    <input type="submit" value="submit"/>
    <input type="reset" value="cancel"/>
</form>
<%
}
catch(Exception e){
    System.out.print(e);
    e.printStackTrace();
}
String godownID = request.getParameter("godown");
String SID = request.getParameter("supplier");
String itemID=request.getParameter("item");
String dateOfSupply=request.getParameter("dateOfSupply");
String quantity=request.getParameter("quantity");
String receiptNo=request.getParameter("receiptNo");
String REmpID=request.getParameter("remp");
String bill=request.getParameter("bill");
String CEmpID=request.getParameter("cemp");
String invoice=request.getParameter("invoice");

try {
Class.forName("com.mysql.jdbc.Driver");
con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
st=con1.createStatement();
String query="insert into
inwards(godownID,SID,itemID,dateOfSupply,quantity,receiptNo,REmpID,bi
ll,CEmpID,invoice) values
('"+godownID+"','"+SID+"','"+itemID+"','"+dateOfSupply+"','"+quantity
+"','"+receiptNo+"','"+REmpID+"','"+bill+"','"+CEmpID+"','"+invoice+
')";
System.out.println(query);
int i=st.executeUpdate(query);
Integer q2 = Integer.parseInt(quantity);
ResultSet rs = st.executeQuery("Select * from godown where
godownID='"+ godownID + "'");
if(rs.next())
{
Integer stock=rs.getInt("GStock");
stock=stock + q2;
String query2="update godown set GStock = '"+stock+"' ";
System.out.println(query2);
int j=st.executeUpdate(query2);
out.println("Successfully Registered your details");
}
}
catch(Exception e){
System.out.print(e);
}

```

```

e.printStackTrace();
}
}
%>

<style>
form {
    width: 300px;
    margin: 0 auto;
    padding: 10px;
}
input[type=text], select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}
input[type=submit] {
    width: 45%;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
input[type=reset] {
    width: 45%;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
input[type=date]{
    width: 60%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}
input[type=submit]:hover {
    background-color: #111;
}
input[type=reset]:hover {
    background-color: #111;
}
body {
    margin: 0;
    background-image:
        url("images/yellow-gold-scratched-paint-wall-
background.jpg");
    background-size: cover;
    width: 100%;
    height: 400px;
}

```

```

        }

    ul.sidenav {
        list-style-type: none;
        margin: 0;
        padding: 0;
        width: 10%;
        background-color: #333;
        position: fixed;
        height: 100%;
        overflow: hidden;
    }

    ul.sidenav li a {
        display: block;
        color: #fff;
        padding: 14px 16px;
        text-decoration: none;
    }

    ul.sidenav li a.active {
        background-color: #4CAF50;
        color: white;
    }

    ul.sidenav li a:hover {
        background-color: #111;
    }

    div.content {
        margin-left: 10%;
        padding: 1px 16px;
    }

    @media screen and (max-width: 1050px) {
        ul.sidenav {
            width: 100%;
            height: auto;
            position: relative;
        }
        ul.sidenav li a {
            float: left;
            padding: 15px;
        }
        div.content {
            margin-left: 0;
        }
    }
    footer {
        position: fixed;
        bottom: 0;
        color: white;
        margin-left: 43%;
    }
</style>

</center>
</font>
</div>
</body>
<footer>
    <small>&copy; Copyright 2017, Stock Analyser NEC</small>
</footer>
```

```

</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>customer List</title>
</head>
<body onload="history.forward(1)">
    <ul class="sidenav">
        <li><a href="customerAdd.jsp">Add Customer</a></li>
        <li><a class="active" href="customerlist.jsp">List
Customer</a></li>
        <li><a href="customerDelete.jsp">Delete
Customer</a></li>
        <li><a href="user.jsp">Back</a></li>
        <li class="Logout"><a href="Logout.jsp">Logout</a></li>
    </ul>
    <div class="content">
        <font face="Tw Cen MT" color="1234">
            <center>
                <%
                    String admin = (String)session.getAttribute("userid");
                    if ((session.getAttribute("userid") == null) ||
(session.getAttribute("userid") == ""))
                %>
                    You are not logged in<br /> <a href="Login.html">Please
Login</a>
                <%
                }
            else {
                Connection con1 = null;
                Statement st = null;
                Statement st2 = null;
                try {
                    Class.forName("com.mysql.jdbc.Driver");
                    con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
                    st = con1.createStatement();
                    ResultSet rs = st.executeQuery("Select *
from customer");
                %>

```

CID	CName	itemID

```

                <h2 align="center">
                    <u>Customer List</u>
                </h2>
                <table cellpadding="15">
                    <tr>
                        <th>CID</th>
                        <th>CName</th>
                        <th>itemID</th>
                    </tr>
                    <%
                        while (rs.next()) {
                    %>
                    <tr>
                        <td>
```

```

        <%
        out.println(rs.getString("CID"));
        %>
        </td>
        <td>
        <%
        out.println(rs.getString("CName"));
        %>
        </td>
        <td>
        <%
        out.println(rs.getString("itemID"));
        %>
        </td>
        </tr>
        <%
        }
    }

} catch (Exception e) {
    System.out.print(e);
    e.printStackTrace();
}
}

%>
</table>
</center>
</font>
</div>
</body>
<style>
th, td {
    border-bottom: 1px solid #ddd;
    text-align: left;
}
body {
    margin: 0;
    background-image:
        url("images/yellow-gold-scratched-paint-wall-
background.jpg");
    background-size: cover;
    width: 100%;
    height: 400px;
}
ul.sidenav {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 10%;
    background-color: #333;
    position: fixed;
    height: 100%;
    overflow: hidden;
}
ul.sidenav li a {
    display: block;
    color: #fff;
    padding: 14px 16px;
}

```

```

        text-decoration: none;
    }

    ul.sidenav li a.active {
        background-color: #4CAF50;
        color: white;
    }

    ul.sidenav li a:hover {
        background-color: #111;
    }

    div.content {
        margin-left: 10%;
        padding: 1px 16px;
    }

    @media screen and (max-width: 1050px) {
        ul.sidenav {
            width: 100%;
            height: auto;
            position: relative;
        }
        ul.sidenav li a {
            float: left;
            padding: 15px;
        }
        div.content {
            margin-left: 0;
        }
    }
    footer {
        position: fixed ;
        bottom:0;
        color: white;
        margin-left: 40%;
    }

```

</style>

<footer>

<small>&copy; Copyright 2017, Stock Analyser NEC</small>

</footer>

</html>

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>User Delete</title>
</head>
<body onload="history.forward(1)">
    <ul class="sidenav">
        <li><a href="userView.jsp">View Users</a></li>
        <li><a class="active" href="userDelete.jsp">Delete
User</a></li>
            <li><a href="user.jsp">Back</a></li>
            <li class="Logout"><a href="Logout.jsp">Logout</a></li>
    </ul>
    <div class="content">

```

```

<center>
    <font face="Tw Cen MT" color="1234"> <%
        String admin =
(String)session.getAttribute("userid");
        if ((session.getAttribute("userid") == null) ||
(session.getAttribute("userid") == ""))
            %>
        You are not logged in<br /> <a href="Login.html">Please
Login</a>
    <%
}
else if(!admin.substring(0,1).equals("9")) {
    %>
    You are not Admin<br /> <a
href="Login.html">Please Login as Admin to access this page</a>
    <%
}
else {
Connection con1 = null;
Statement st = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
    st = con1.createStatement();
    ResultSet rs = st.executeQuery("Select empName from
emp");
    %>

        <h1 align="center">
            <u>Delete User</u>
        </h1>
        <form align="center"
action="userDelete.jsp">
            <select name="users">
                <option>select user</option>
                <%
                    while (rs.next()) {
                        String
empName = rs.getString("empName");
                %>
                <option
value=<%out.println(empName);%>>
                <%
                    out.println(empName);
                %>
                </option>
                <%
                    }
                %>
            </select> <br> <input type="submit"
value="delete" />
            </form> <%--// --%> <%
} catch (Exception e) {
    System.out.print(e);
    e.printStackTrace();
}
String emp = request.getParameter("users");
try {
    Class.forName("com.mysql.jdbc.Driver");

```

```

        con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
        st = con1.createStatement();
        String query = "delete from emp where empName=' " + emp +
"" ;
        System.out.println(query);
        int i = st.executeUpdate(query);
        if (i > 0)
            out.println("Successfully deleted user from
database");
        else
            out.println("");
    } catch (Exception e) {
        System.out.print(e);
        e.printStackTrace();
    }
}
}

%>
        </font>
    </center>
</body>
<style>
input[type=submit], select {
    width: 300px;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

input[type=submit] {
    width: 100px;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

body {
    margin: 0;
    background-image:
        url("images/yellow-gold-scratched-paint-wall-
background.jpg");
    background-size: cover;
    width: 100%;
    height: 400px;
}

ul.sidenav {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 10%;
    background-color: #333;
    position: fixed;
    height: 100%;
    overflow: hidden;
}

```

```

}

ul.sidenav li a {
    display: block;
    color: #fff;
    padding: 14px 16px;
    text-decoration: none;
}

ul.sidenav li a.active {
    background-color: #4CAF50;
    color: white;
}

ul.sidenav li a:hover {
    background-color: #111;
}

div.content {
    margin-left: 10%;
    padding: 1px 16px;
}

@media screen and (max-width: 1050px) {
    ul.sidenav {
        width: 100%;
        height: auto;
        position: relative;
    }
    ul.sidenav li a {
        float: left;
        padding: 15px;
    }
    div.content {
        margin-left: 0;
    }
}
footer {
    position: fixed;
    bottom:0;
    color: white;
    margin-left: 33%;
}

</style>
<footer>
    <small>&copy; Copyright 2017, Stock Analyser NEC</small>
</footer>
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.sql.*"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-
1">
<title>Returns Report</title>
</head>
<body onload="history.forward(1)">
<ul class="sidenav">
    <li><a href="returnsAdd.jsp">Add Returns</a></li>

```

```

        <li><a class="active" href="returnsReport.jsp">Returns
Report</a></li>
        <li><a href="user.jsp">Back</a></li>
        <li class="Logout"><a href="Logout.jsp">Logout</a></li>
    </ul>
    <div class="content">
        <font face="Tw Cen MT" color="1234">
            <center>

        </center>
        <form align="center" action="returnsReport.jsp">
            <%
                String admin =
(String)session.getAttribute("userid");
                if ((session.getAttribute("userid") == null) ||
(session.getAttribute("userid") == ""))
                    %>
                You are not logged in<br /> <a href="Login.html">Please
Login</a>
            <%
                }
            else {
                Connection con1 = null;
                Statement st = null;
                ResultSet rs = null;
                try {
                    Class.forName("com.mysql.jdbc.Driver");
                    con1 =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
                    st = con1.createStatement();
                    rs = st.executeQuery("Select
godownID from godown");
                    %>
                    <h1 align="center">
                        <u>Returns Report</u>
                    </h1>
                    <select name="godownID">
                        <option>select godown</option>
                        <%
                            while (rs.next()) {
                                String godownID =
rs.getString("godownID");
                                %>
                                <option value=<%out.println(godownID);%>>
                                    <%
                                        out.println(godownID);
                                    %>
                                </option>
                                <%
                            }
                        <%
                    </select>
                    <%
                } catch (Exception e) {
                    System.out.print(e);
                    e.printStackTrace();
                }
            <%
        Starting Date <input type="date" name="startDate"
required /><br>

```

```

        Ending Date<input type="date" name="endDate"
required />
            <input type="submit" value="get details"
onClick="show()" />
        </form>
    </font>

<%
    String godownID = request.getParameter("godownID");
    String startDate = request.getParameter("startDate");
    String endDate = request.getParameter("endDate");
    Connection con = null;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/stockanalyse
r", "root", "2513218");
        st = con.createStatement();
        rs = st.executeQuery("Select * from returns
where (dateOfreturn between ('" + startDate + "') and ('"
                + endDate + "'')) and GodownID='"
                + godownID + "'");
    %><font face="Tw Cen MT" color="1234">
        <center>
    <%
        %>
        <table border="0" cellpadding="1">
            <tr>
                <th>Item ID</th>
                <th>invoice</th>
                <th>quantity</th>
                <th>Bill</th>
            </tr>
        <%
    } catch (Exception e) {
        System.out.print(e);
        e.printStackTrace();
    }
        while (rs.next()) {
    %>
            <tr>
                <td>
    <%
        out.println(rs.getString("itemID"));
    %>
                </td>
                <td>
    <%
        out.println(rs.getString("invoice"));
    %>
                </td>
                <td>
    <%
        out.println(rs.getString("quantity"));
    %>
                </td>
                <td>
    <%

```

```

        out.println(rs.getString("bill"));
        %>
    </td>
</tr>

<%
}
%>
</table>

</center>
</font>
</body>
<style>
form {
    width: 300px;
    margin: 0 auto;
    padding: 10px;
}

input[type=text], select {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

input[type=submit] {
    width: 45%;
    background-color: #333;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type=submit]:hover {
    background-color: #111;
}

td {
    text-align: left;
    width: 50px;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

th, td {
    border-bottom: 1px solid #ddd;
    text-align: left;
}

td:hover {

```

```

        background-color: #f5f5f5
    }

    th:hover {
        background-color: #f5f5f5
    }
    body {
        margin: 0;
        background-image:
            url("images/yellow-gold-scratched-paint-wall-
background.jpg");
        background-size: cover;
        width: 100%;
        height: 400px;
    }

    ul.sidenav {
        list-style-type: none;
        margin: 0;
        padding: 0;
        width: 10%;
        background-color: #333;
        position: fixed;
        height: 100%;
        overflow: hidden;
    }

    ul.sidenav li a {
        display: block;
        color: #fff;
        padding: 14px 16px;
        text-decoration: none;
    }

    ul.sidenav li a.active {
        background-color: #4CAF50;
        color: white;
    }

    ul.sidenav li a:hover {
        background-color: #111;
    }

    div.content {
        margin-left: 10%;
        padding: 1px 16px;
    }

    @media screen and (max-width: 1050px) {
        ul.sidenav {
            width: 100%;
            height: auto;
            position: relative;
        }
        ul.sidenav li a {
            float: left;
            padding: 15px;
        }
        div.content {
            margin-left: 0;
        }
    }
    footer {

```

```
        position: fixed;
        bottom: 0;
        color: white;
        margin-left: 3%;
    }

</style>
<footer>
    <small>&copy; Copyright 2017, Stock Analyser NEC</small>
</footer>
</html>
```

## **8 TESTING AND TEST CASES**

### **8.1 Testing Survey**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **8.2 Types of Tests**

#### **8.2.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

#### **8.2.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration test demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### **8.2.3 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

#### **Test Strategy and Approach**

Field testing will be performed manually and functional tests will be written in detail.

#### **Test Objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

#### **Features to Be Tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### **Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

## **Test Results**

All the test cases mentioned above passed successfully. No defects.

### **8.2.4 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

## **Test Results**

All the test cases mentioned above passed successfully. No defects encountered.

### 8.3 Test Cases

Table 7: Test cases table for Student information system

S no.	Input	Expected Behavior	Actual Behavior	Output
Test case 1	User login	Valid data	Invalid username /password	Failed
Test case 2	User login	Valid data	valid username /password	Success
Test case 3	Registration	Valid data	Invalid data	Failed
Test case 4	Registration	Valid data	Valid data	Success
Test case 5	User page	Open page	Page open not	Failed
Test case 6	Inwards add	Valid data	Valid data	Success

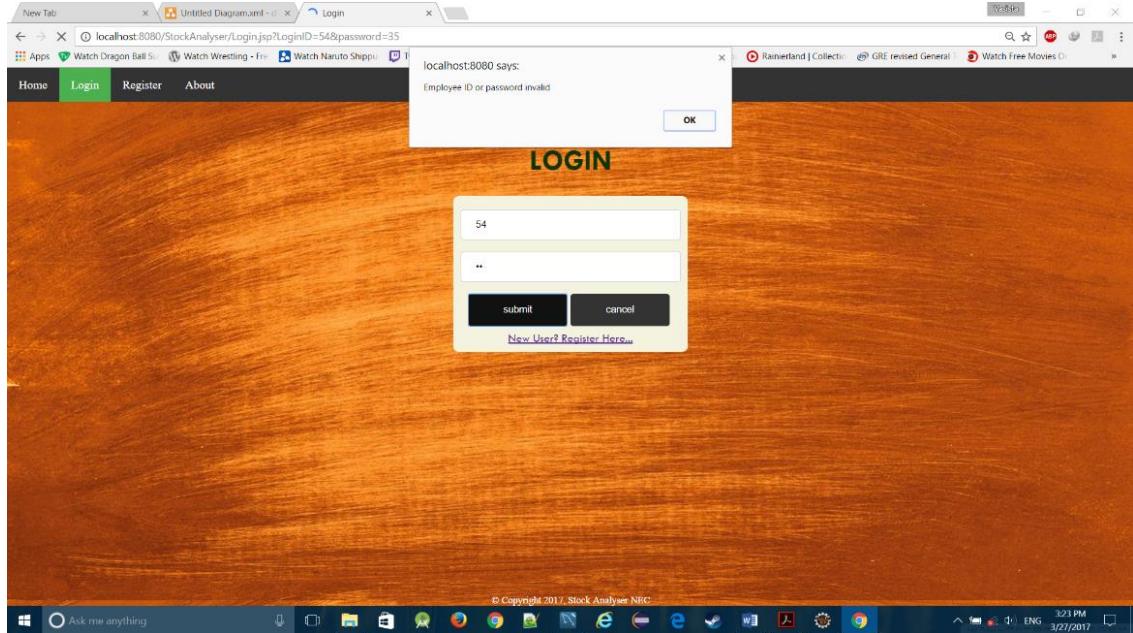
## Test Case 1

Input: User Login

Expected Behaviour: **Valid data**

Actual Behaviour: **Invalid Username/Password**

Result: **Failed**



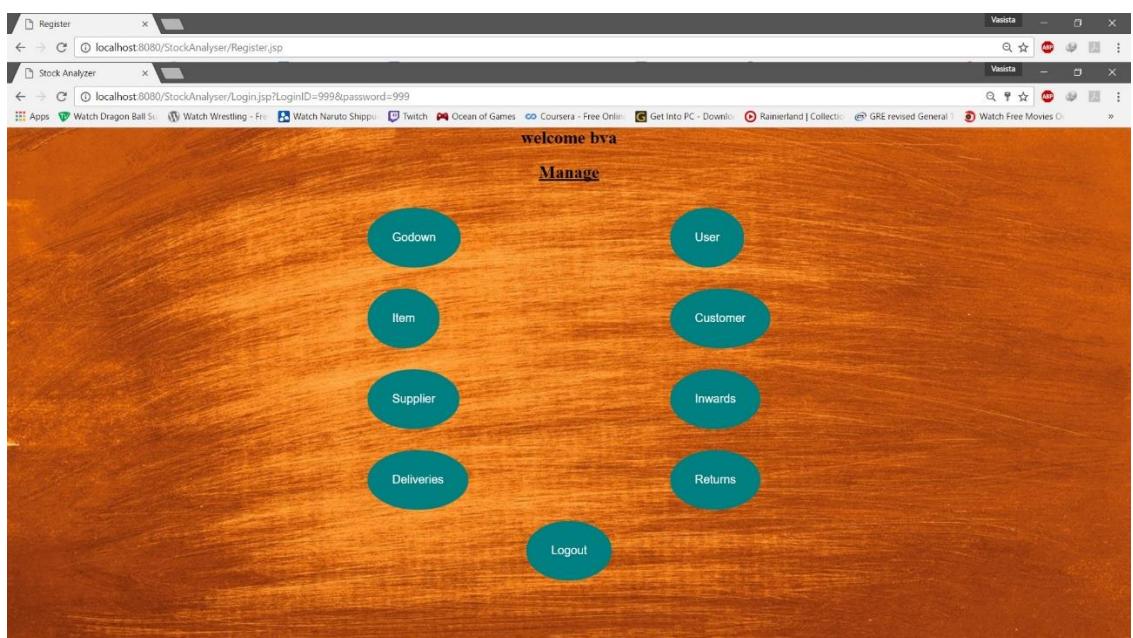
## Test Case 2

Input: User Login

Expected Behaviour: **Valid data**

Actual Behaviour: **Valid Username/Password**

Result: **Success**



## Test Case 3

Input: **user Registration**

Expected Behaviour: **Valid employee id**

Actual Behaviour: **Invalid employee id**

Result: **Failed**

The screenshot shows a 'Register Here' form on a website. The 'Employee ID' field contains '123', which is highlighted in red and has a tooltip: 'Please match the requested format. 5 Digit employee ID'. The 'Gender' field is set to 'Male'. Other fields include 'Full Name' (vasista), 'Job Location' (mumbai), 'Join Date' (12/30/2015), and 'Role' (manager). Buttons for 'Register' and 'cancel' are at the bottom.

## Test Case 4

Input: **Registration**

Expected Behaviour: **Valid details**

Actual Behaviour: **Valid details (Registration successful)**

Result: **Success**

click here'."/>

The screenshot shows a 'Register Here' form on a website. The 'Employee ID' field is empty. The 'Password' field contains '12345', with a validation message: 'atleast 5 characters'. The 'Gender' field is set to 'Male'. Other fields include 'Full Name', 'Job Location', 'Join Date', 'Role', 'Enter Password', and 'Re Enter Password'. A success message at the bottom states 'Successfully Registered your details [click here](#)'.

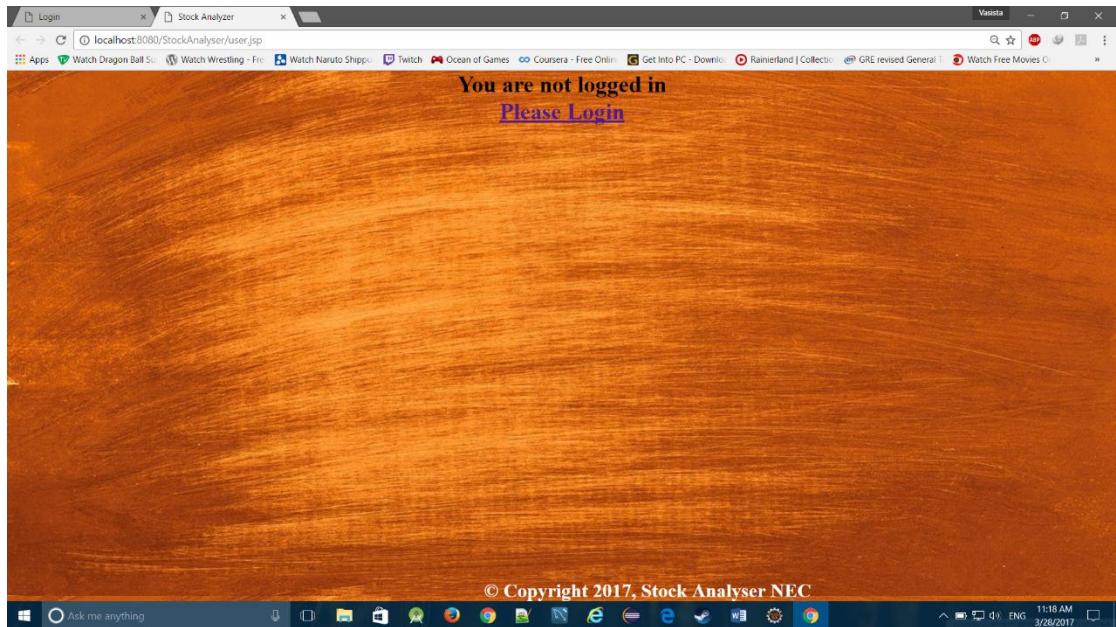
## Test Case 5

## Input: user page

## Expected Behaviour: **page Open**

Actual Behaviour: **page not open (not logged in)**

## Result: Failure



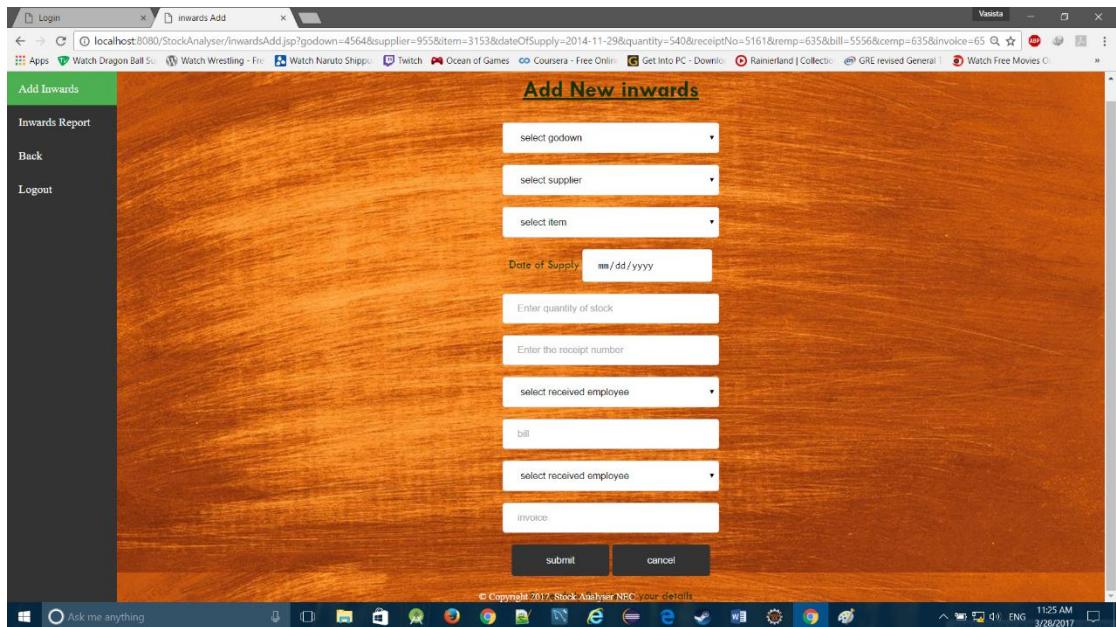
## Test Case 6

## **Input: Inwards Details**

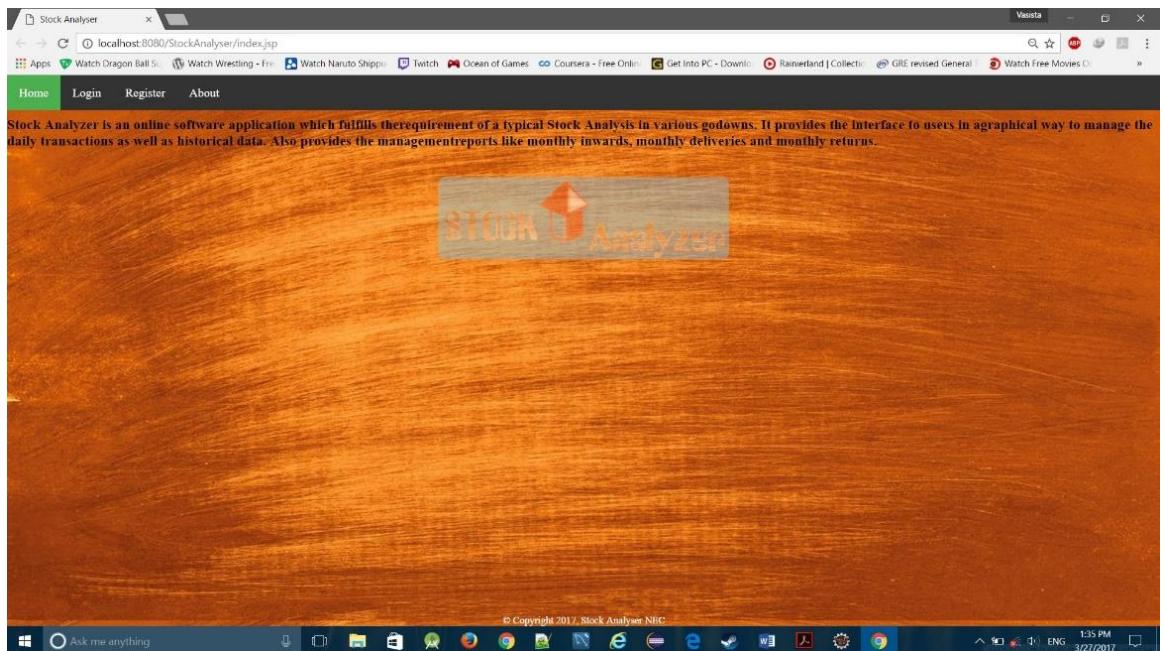
## Expected Behaviour: **valid details**

## Actual Behaviour: **valid details**

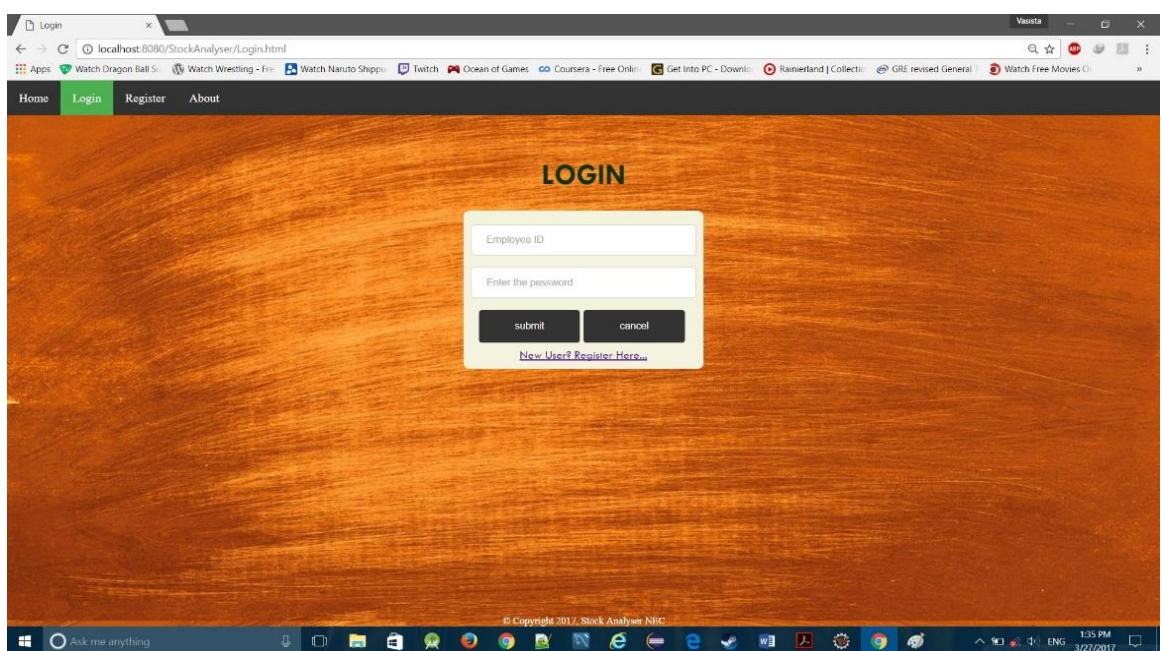
## Result: Success



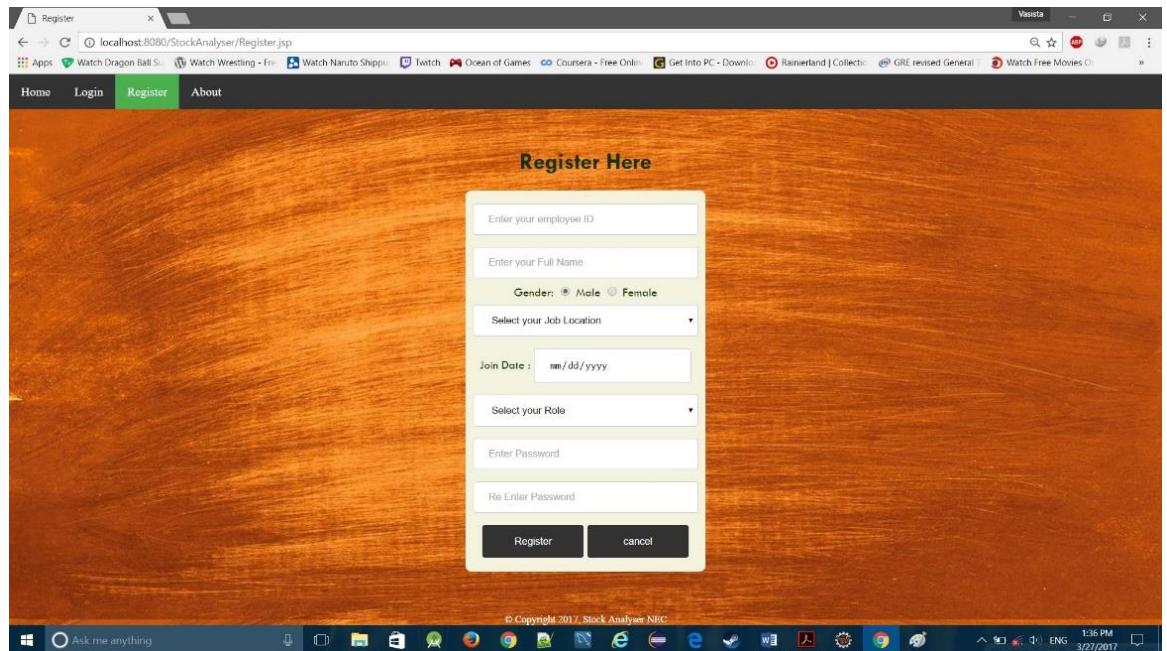
## 9 OUTPUT SCREENS



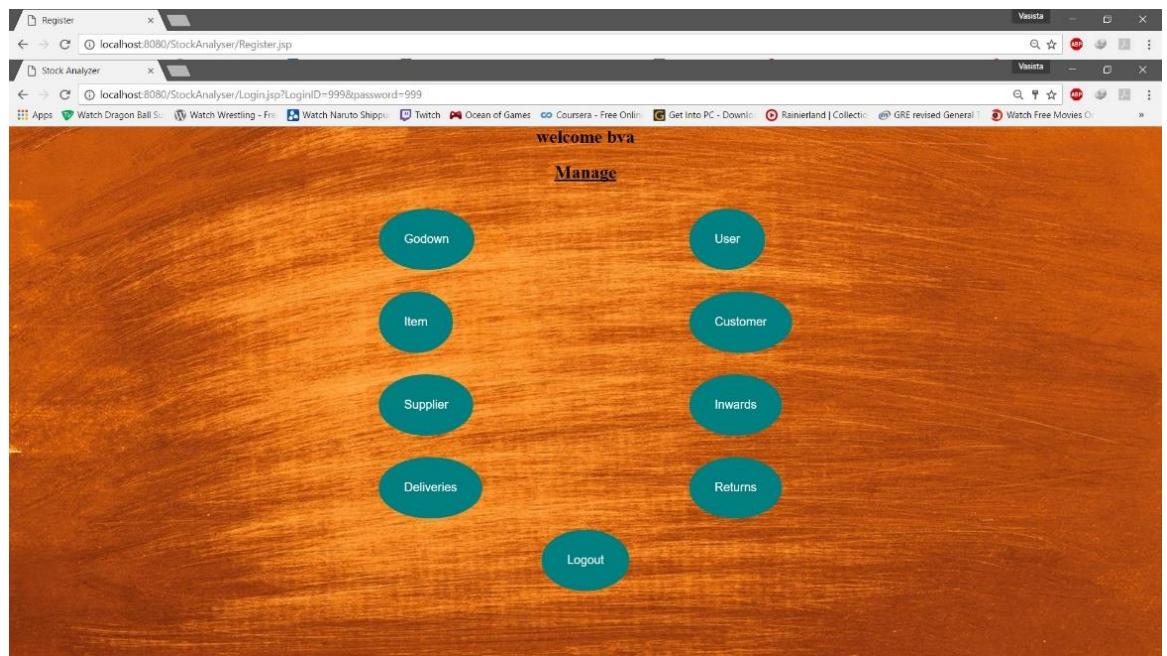
**Fig: Index Page**



**Fig 9.1: Login Page**



**Fig 9.2: Register Page**



**Fig 9.3: User Page**

**Add New Godown**

Enter the godown ID

Location of your godown

select godown Manager

Godown Start Date mm/dd/yyyy

Capacity in quintals

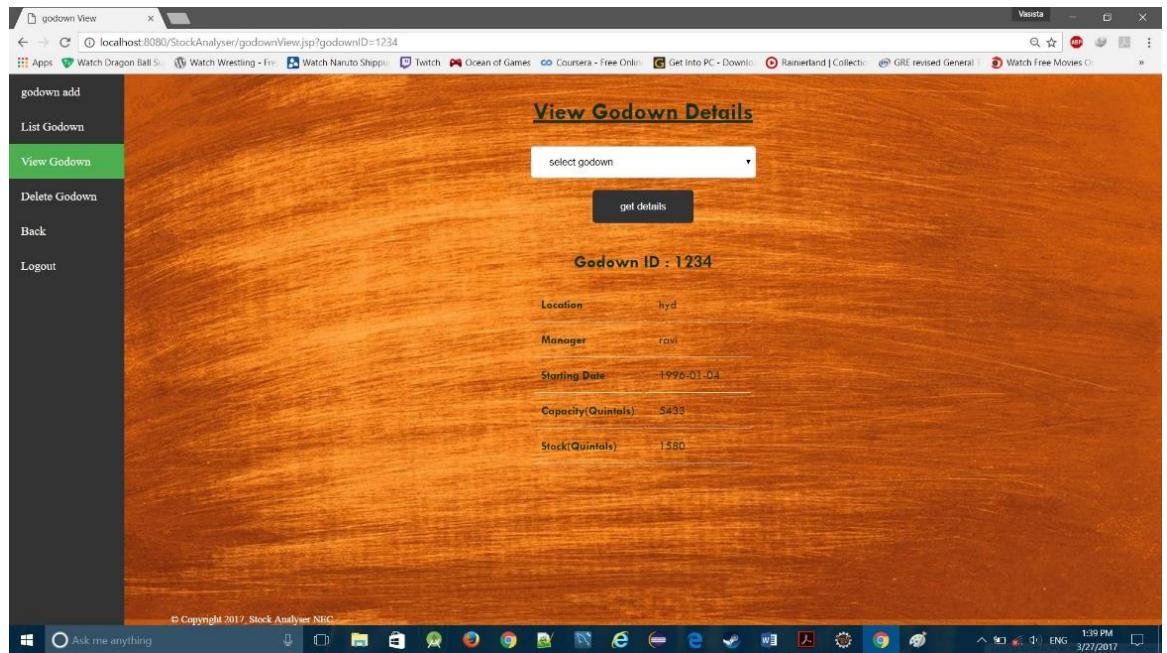
Present Stock in godown

submit cancel

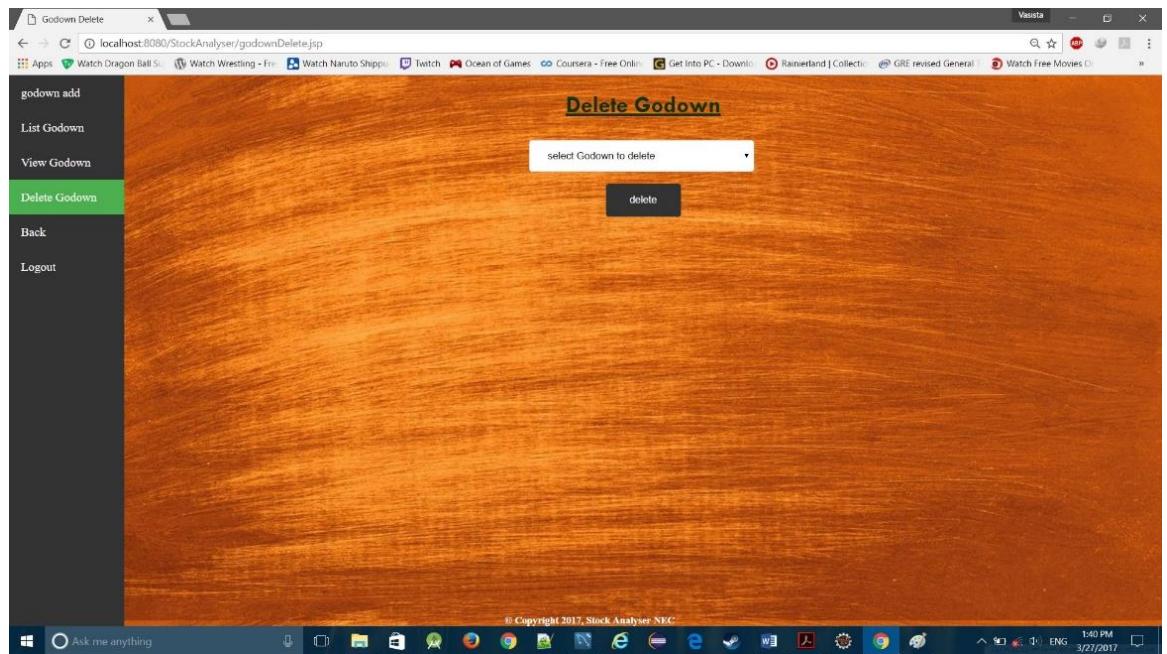
**Fig 9.4: Add Godown Page**

GodownID	Location	Manager
1234	hyd	ravi
4564	bangalore	vistara

**Fig 9.5: List Godown Page**



**Fig 9.6: View Godown Page**



**Fig 9.7: Delete Godown Page**

The screenshot shows a Windows desktop environment with a Java application window titled "User View". The URL in the address bar is "localhost:8080/StockAnalyser/userView.jsp". The main content area is titled "User Details" and displays a table of user data. The columns are labeled "emplID", "empName", "gender", "Jlocation", and "role". The data rows are as follows:

emplID	empName	gender	Jlocation	role
555	lava	male	mumbai	clerk
635	tarak	male	hyd	security
987	Srinivasarao	male	hyd	manager
999	lava	male	chennai	accountant
2222	varista	male	delhi	clerk
8654	varista arinath b	male	chennai	security
23156	revi	male	delhi	clerk
55555	vomar	male	hyd	security
65432	krishna	male	chennai	accountant
98765	sivo	male	hyd	manager
7854521	sedik	male	mumbai	clerk

At the bottom of the window, it says "Copyright 2017 Stock Analyser NBC". The taskbar at the bottom shows various icons for applications like File Explorer, Google Chrome, and Microsoft Edge.

**Fig 9.8: User Details Page**

The screenshot shows a Windows desktop environment with a Java application window titled "User Delete". The URL in the address bar is "localhost:8080/StockAnalysers/userDelete.jsp". The main content area is titled "Delete User" and contains a single input field with the placeholder "select user" and a "delete" button below it.

On the left side of the window, there is a sidebar with the following navigation links: "View Users", "Delete User", "Back", and "Logout".

At the bottom of the window, it says "Copyright 2017 Stock Analyser NBC". The taskbar at the bottom shows various icons for applications like File Explorer, Google Chrome, and Microsoft Edge.

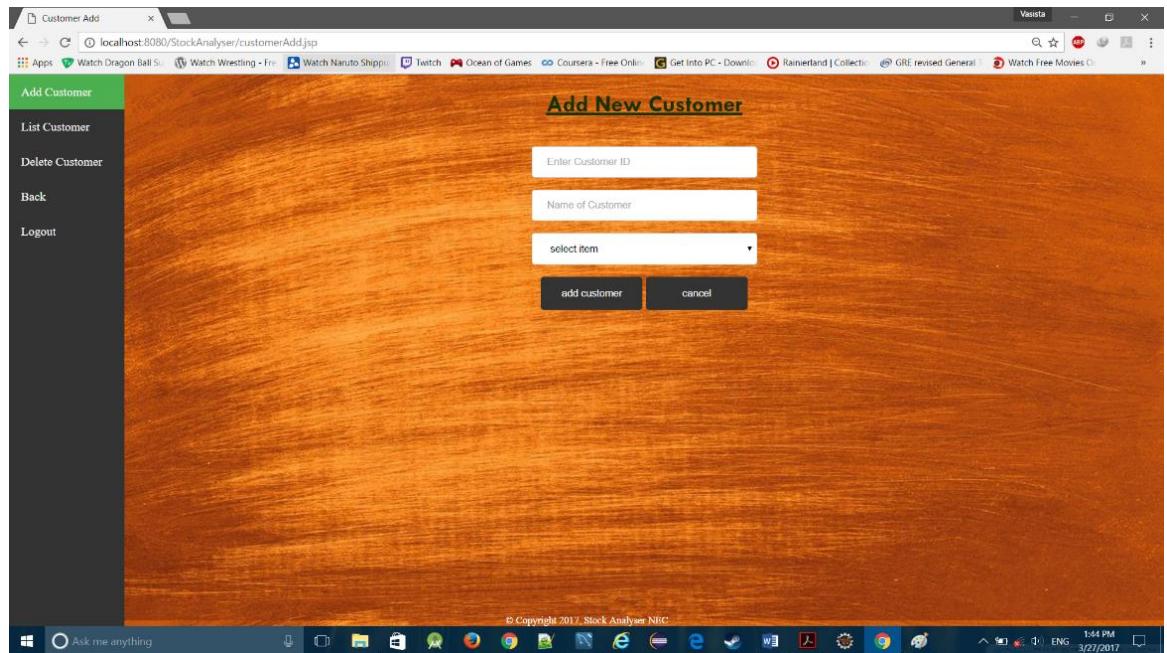
**Fig 9.9: Delete User Page**

Item List			
	ItemID	ItemName	ItemDetails
	2520	eggs	rdgfhdfifgffofkdkmgmnikd
	3153	sdqdsq	dsgfdhnddjjgandlgnodlgsdg
	56221	rice	15
	632541	wheat	wheat quantities of bags

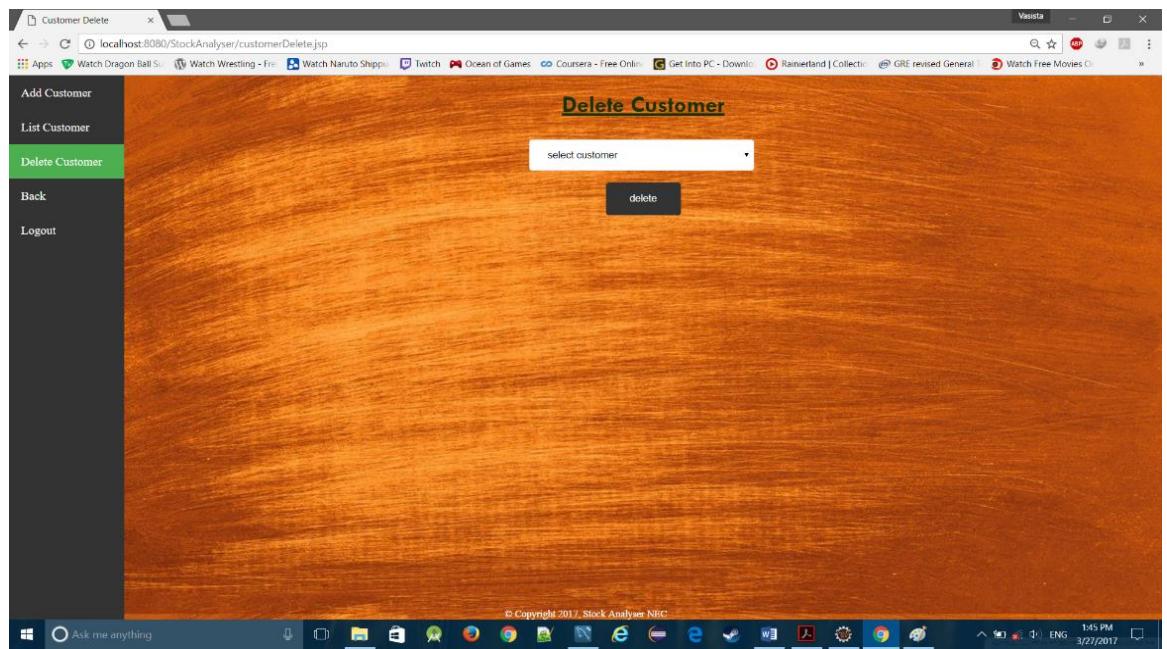
**Fig 9.10: Item List Page**

Delete item			
	select item to delete	delete	
		<input type="button" value="delete"/>	

**Fig 9.11: Item Delete Page**



**Fig 9.12: Add Customer Page**



**Fig 9.13: Delete Customer Page**

The screenshot shows a Windows desktop environment with a Java application window titled "customer List". The URL in the address bar is "localhost:8080/StockAnalyser/customerlist.jsp". The application has a sidebar with links: "Add Customer", "List Customer" (which is highlighted in green), "Delete Customer", "Back", and "Logout". The main content area is titled "Customer List" and displays a table with columns: CID, CName, and itemID. The data in the table is as follows:

CID	CName	itemID
325	drgsdg	56221
3216	arjun	3153
32156	nomed	2520
58626	hori	2520

**Fig 9.14: List Customer Page**

The screenshot shows a Windows desktop environment with a Java application window titled "supplier Add". The URL in the address bar is "localhost:8080/StockAnalyser/supplier/Add.jsp". The application has a sidebar with links: "Add Supplier" (highlighted in green), "List Supplier", "Delete Supplier", "Back", and "Logout". The main content area is titled "Add New Supplier" and contains the following form fields:

- Enter Supplier ID
- Name of Supplier
- select item
- add supplier
- cancel

**Fig 9.15: New Supplier Page**

A screenshot of a Windows desktop showing a web browser window titled "Supplier List". The URL is "localhost:8080/StockAnalyser/supplierlist.jsp". The page contains a sidebar with links: "Add Supplier", "List Supplier" (which is highlighted in green), "Delete Supplier", "Back", and "Logout". The main content area is titled "Supplier List" and displays a table with four rows of data:

SID	SName	itemID
955	anfdg	2520
57465	krishna	2520
323536	varun	65654

**Fig 9.16: List Supplier Page**

A screenshot of a Windows desktop showing a web browser window titled "Supplier Delete". The URL is "localhost:8080/StockAnalyser/supplierDelete.jsp". The page contains a sidebar with links: "Add Supplier", "List Supplier", "Delete Supplier" (which is highlighted in green), "Back", and "Logout". The main content area is titled "Delete Supplier" and features a dropdown menu labeled "select supplier" and a "delete" button.

**Fig 9.17: Delete Supplier Page**

Add New Inwards

select godown

select supplier

select item

Date of Supply mm/dd/yyyy

Enter quantity of stock

Enter the receipt number

select received employee

bill

select received employee

INVOICE

submit cancel

**Fig 9.18: Add Inwards Page**

Inwards Report

select godown

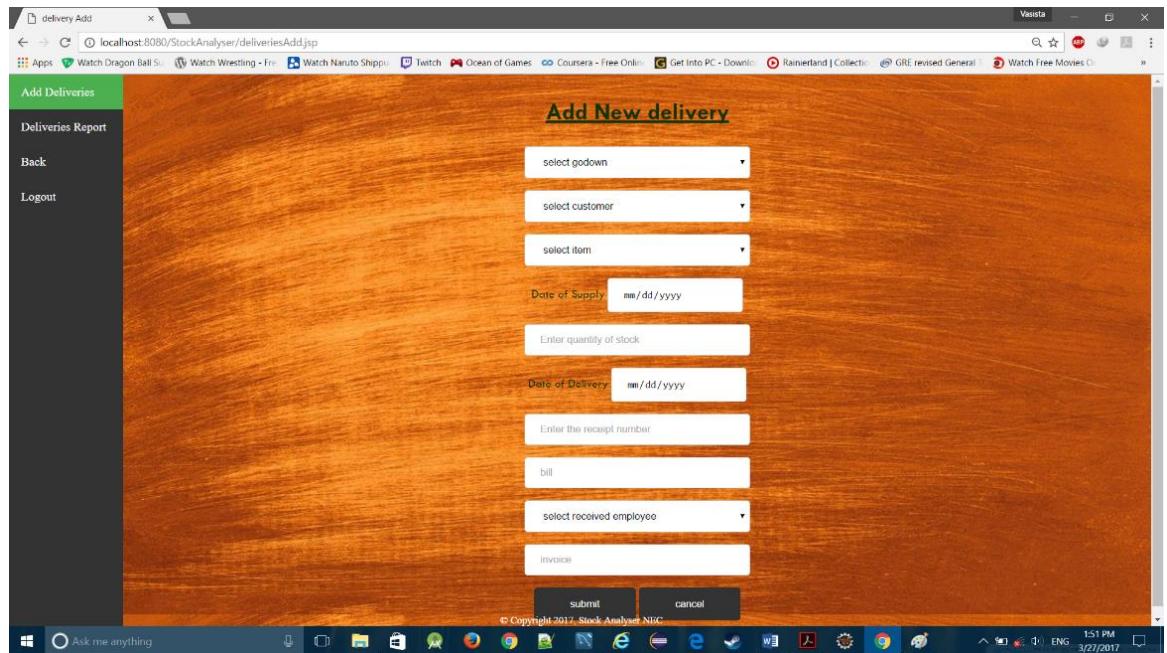
Starting Date mm/dd/yyyy

Ending Date mm/dd/yyyy

get details

Item ID	Invoice	Quantity	Bill
2520	50165	800	85321
3153	513113	50	56622
3153	987542	800	665153

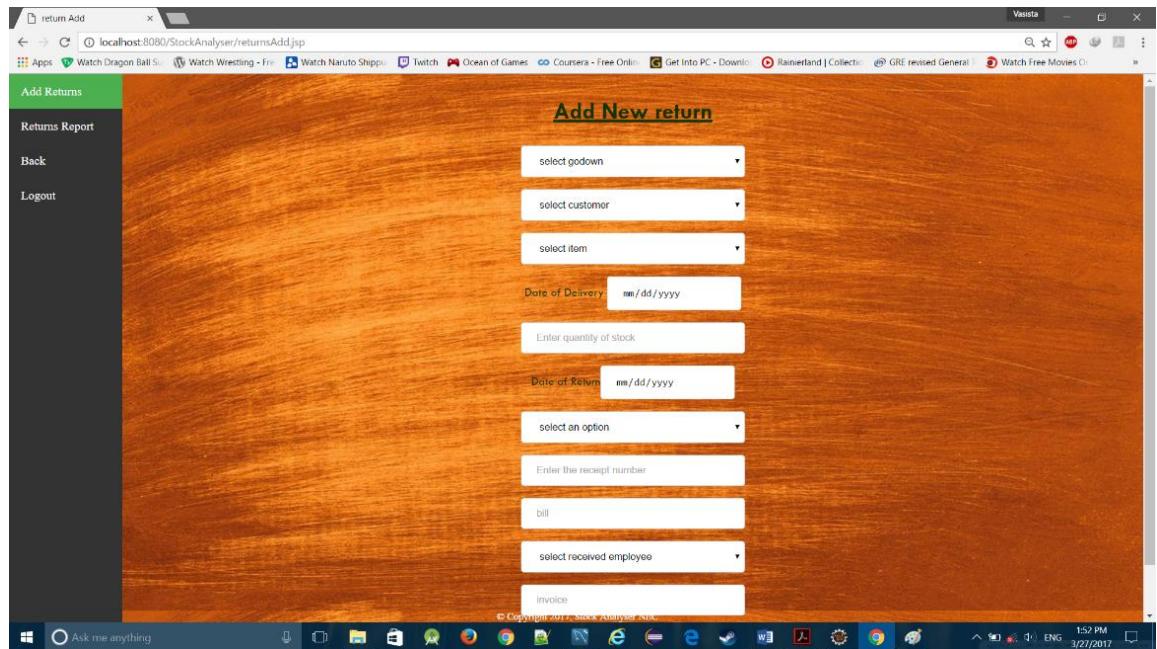
**Fig 9.19: Inwards Report Page**



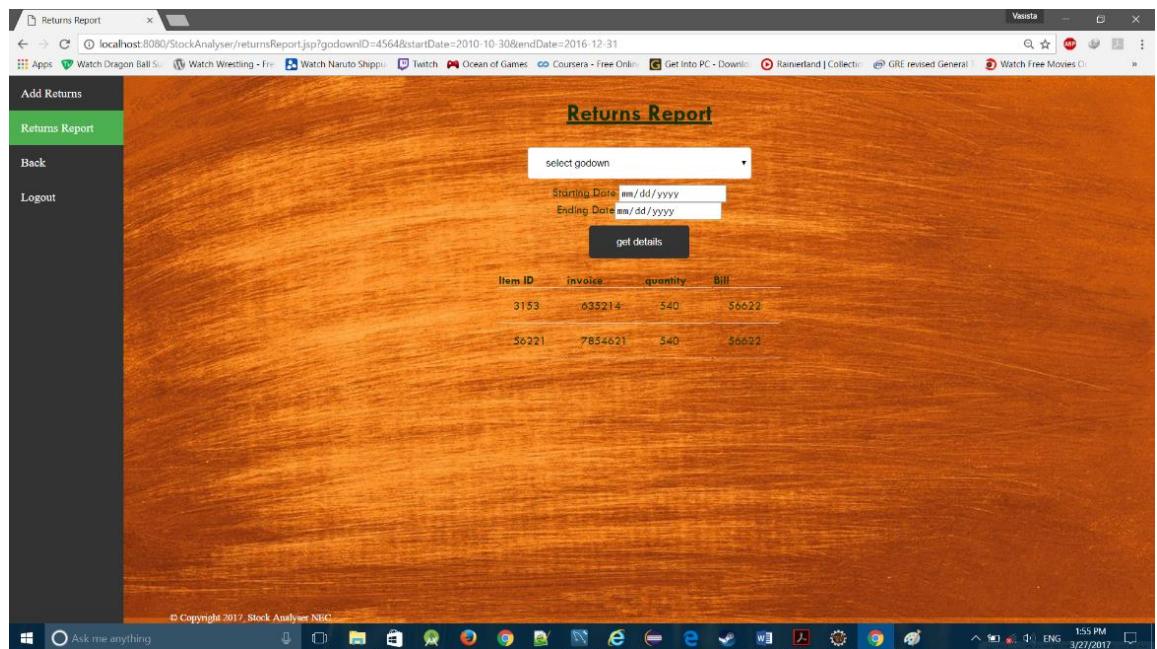
**Fig 9.20: Add Delivery Page**

Deliveries Report			
<input type="button" value="select godown"/> Starting Date: <input type="text" value="mm/dd/yyyy"/> Ending Date: <input type="text" value="mm/dd/yyyy"/>			
Item ID	Invoice	Quantity	Bill
3153	66516	800	5556
2520	654231	560	56622
3153	663521	650	65321

**Fig 9.21: Delivery Report Page**



**Fig 9.22: Add Return Page**



**Fig 9.23: Returns Report Page**

## **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 FUTURE ENHANCEMENT**

Security measures will be taken to make sure user, customer and supplier data cannot be gained by unauthorized users. Modifications for the project will be done and some new features like showing godowns in map will be added.

## **12 BIBILOGRAPHY & REFERENCES**

### **References:**

- <http://www.google.com>
- <http://www.codeprojct.com>
- <http://www.studentdeveloper.com>
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [www.w3schools.com](http://www.w3schools.com)
- <http://www.c#tutorials.com>

