**Chapter 1: Problem Definition**

**I. Description about Inventory Management System**

Today, applications to life more than. And computing and informatics is being developed so fast. Management a shop or a inventory with a large amount data stored. Management bills, Tran tract and information of customer in a day will very difficult if you done by hand.

That is objectives of Inventory Management System, which we provided here. This is application software has functional help sales, inventory manager, sales manager and administrator easy view, validate and manipulation information of customer, bills and categories product. Beyond, systems support administrator statistics numeric about

Functional of application has been developed to provide a lot of benefit to user. Help to management inventory easier. All business process faster and correctly.

Base Java programming language can run on any platform. Your application can execute even on many platform differences if they setup Java Runtime Environment .Therefore system will implemented widely and reality higher.

**II. Goals of project**

Inventory Management System has been developed to achieve these goals:

Management easier information of product and customers.

Management bills details, Tran tract and generate report to administrator and sales manager.

**System built with functions below:**

Store information abound administrator, sales, inventory manager in database.

Sales person:

Login to the system through the initial screen of the system



Change the password after logging into the system



See status of the inventory, i.e., whether a particular item is available or not etc



Search for a particular item by typing the initial.



Enter the items purchased by a customer and produce a bill for the same



Cancel the produced bill, in case of error in entering the details, and produce a new one



Take back an already sold Items that is not satisfactory to the customer (the shop allows that) and produce a receipt for the same



Inform the inventory manager about the Items that are not available, so that they can be stocked



Inform the sales manager about any exceptions (such as an



Items are being purchased, but it is not available in the system. Get help about the system on how to use the different features



of the system

Inventory manager

Login to the system/change his password after logging in etc check the status of the inventory



Veer the reports from sales people about items that are not available and need to be kept in the inventory



Validate the inventory against the sales done in the previous day (whether the items that are sold and the change in the inventory for the corresponding items match etc)



Generate inventory reports of the items category-wise, price-wise etc



Generate inventory-trends (like how is coffee powder stock changing over the month)



Sales Manager

Login to the system/change his password after logging in etc



Check how many bills have been generated in the day so far



Check how much money is transacted in the day so far



Check how many bills have been cancelled so far in the day(due to wrong entry by the sales persons)



Check how many items have been returned so far in the day



Check for any exception reports from the sales persons and correct it by contacting the inventory manager



Generate sales-trend graphs for each of the item (like how is apple selling in the last one month)



Administrator

Login to the system and change his password after logging in



Add new users to the system



Add new items/categories to the inventory (‘grape’ may be added to the category ‘fruits’ or a new category such as ‘vegetables’ may be added)



Change the price of an item



**Chapter 2: Customer Requirement**

**Specification (CRS)**

**Client:** User

**Business/Project Objective**

Camp connection is general stores selling various fast moving consumer goods (FMCG). The store is existence from almost 5 years and is situated in the heart of the city.

Since last 2 – 3 years is has been doing exceptionally well because of the huge complex built around the area and people prefer to visit the nearest place to buy various FMCG.

The store is able to cater to the need of people but at the same time finding

it difficult to manage the sales and stock details and to manage the large amount of transaction which is causing discrepancies in the data.

Now a day use of computer is the best way to manage the large amount of data easily and effectively so the owner of the store has decided make use of computer to maintain the data. He has thought of development of Sales and Inventory Management System (SIMS) which will store all the information and required data can be retrieved with ease. SIMS is the proposed solution for his store problems.

Create a database for the store. Add tables to store information about the different Inventory, Sales and Users. The client-server application authenticates the existing user by his username and password. The application has a registration page where a new user can enter his personal details in the given fields. The user details get stored in the database containing details of existing user gets updated. After the user is authenticated, the user can perform the task of sales or inventory manage or sales manage operation which the user has permission.

**Input provided by the Client:**

Inputs for the application :

The information of users

The information of products.

The information of bills.

Outputs from the application :

The information of goods when a user searches with their input information.

Status of the inventory

Produce bills.

Reports about the inventory problems

Process Involved in the application :

Log in or log out the system :

Add a new record

Update records

Search for a particular item in the store

Generate a bill

Manage bills

Generate reports

**H**a**rdware Requirements:**

A minimum computer system that will help you access the entire tool in the course is a Pentium 200 or a higher capability CPU.

256 Megabyte of RAM or Higher.

**Software Requirements:**

Any platform you can have (window, Linux…). jdk 1.4\_02 or later.

Net Bean 6.5.

MS SQL Server 2005.

**Scope of the Work (in brief):**

Depending on the decision taken by the store owner, Sale and Inventory

Management System need to be developed with the following requirements: Registration of new client and authentication of existing client.



Provide a user friendly interface to the user to work with application



Allow user to search for item information in the store, user information. Add new goods and manage the entire item in the store.



Generation of reports related to all the store problems.



**Chapter 3: Architecture and Design of the**

**Project**

Form No.3/eProjects/Design/Version1

Tier 1

Presentation Tier

Tier 2



Database Tier

**User interface – GUI Components**

**Database Access**

**Tow-Tier Architecture of the Project**

**Design Flowcharts**

Administrator Login:

**Start**

**Input user Name and password**

**Display error**

**Check user name and password**

**No**

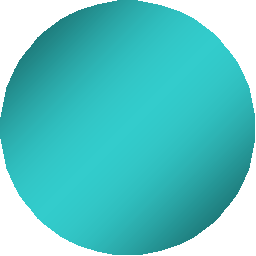
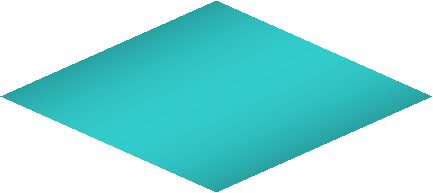
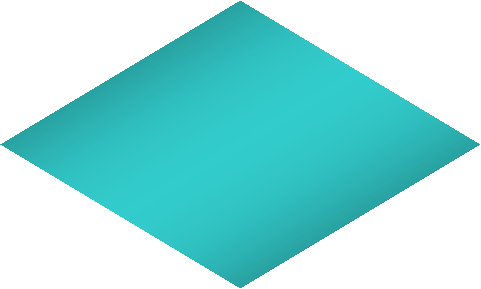
**Check status the user name on the**

**database**

**Display window main**

Yes

**Status =0**



**Stop**

No

**Registr ation**

**Function Add:**

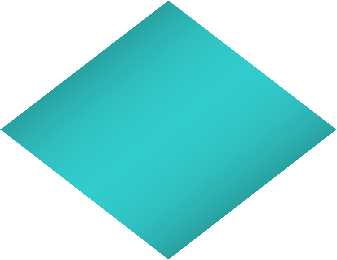
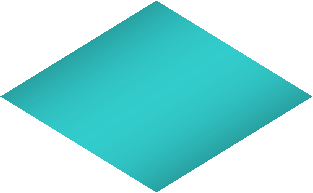
**Start**

**Input information**

**Display error**

**Validation of information**

**And insert to database**



**No**

**No**

**Validate successful**

**Yes**

**Insert successful**

**No**

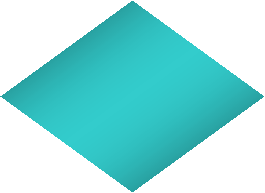
**A > 0**

**Yes**

**a = account record added**

**Yes**

**Display successful**



**Stop**

**Function Edit**

**Start**

**Input information**

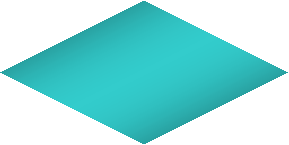
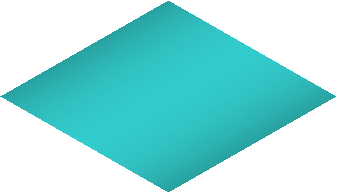
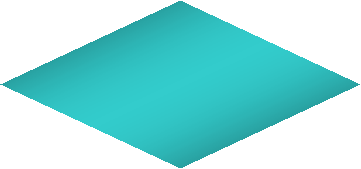
**Validation data on the form**

**Display error**

**Validate successful**

**Yes**

**Update No successful**



**Yes**

**No** **a = count record**

**updated**

**No**

**A > 0**

**Yes**

**Display successful**

**Stop**

**Function Delete**

**Start**

**Input information**

**Validation data on the form**

**Display error**

**Validate successful**

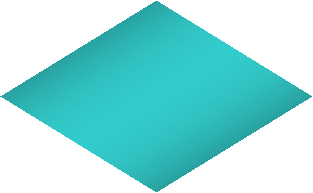
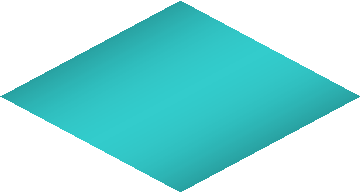
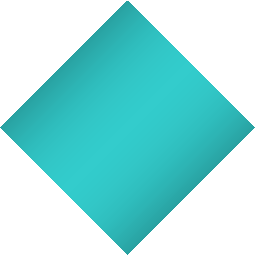
**Yes**

**No**

**Update successful**

**No**

**Yes**



**a = count record deleted**

**No a > 0**

**Yes**

**Display successful**

**Stop**

**Function Search**

**Start**

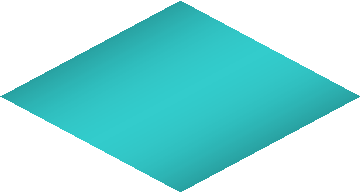


**Input information**

**Validation data on the form**

**Display error**

**No**



**Validate successful**

**Yes**

**Display information**

**Stop**

**Chapter 4: Data Flow Diagram**

**Define:** Data Flows Diagram (DFD) describes the information flow in the system. The next step of system analysis is to consider in detail the information necessary for the implementation for functions discussed above and the one necessary for the improvement of the functions. Modeling tool frequently used for this purpose is DFD. DFD will support 4 main

activities:

**Analysis:** DFD is used to determine requirement of users.

**Design:** DFD is used to map out plan and illustrate solution to analysis and users while designing a new system.

**Communication:** one of the strength of DFD is its simplicity and ease to understand to analysts and users;

**Document:** DFD is used to provide special description of requirement and system design. DFD provide an overview of key functional components of the system but it does not provide any detail on these components. We have to use other tools like database dictionary, process specification to get an idea of which information will be exchanged and how.

**The main components of Context Diagram:**

**Process**

**Data flow**

**External**

**Data store**

The process: Shows the common function of system

The external factors: External factors can be a person, a group of persons or an organization that are sources of information for the systems and are where system products are transferred to.

The data flow: Describe the movement of information from one part of the system to another.

The data store: The Data Store is used to model a collection of data packets at rest. A store is represented graphically by two parallel lines. The name of a Data Store that identifies the store is the plural of the name of the packets that are carried by flows into and out of the Data Store

**Sales**

**Transaction**

**Inventory detail**

**Inventory**

**Manager**

**Inventory report**

**Feedback**

**Feedback**

**Feedback**

**Feedback**

**Manage Sales and inventory**

**Feedback**

**Feedback**

**Transaction checking**

**Admin**

**Authentication details**

**Account details**

**Feedback**

**Feedback**

**Sales Manager**

***Figure 2: Level 0 Context Diagram***

**Sales manager**

**Report**

**User**

**feedback**

**Check, change pass, id**

**3. Transaction details**

**Require**

**Products**

**Relay id, date, time, Item/quantity/price**

**1. Account process**

**Relay information**

**Require**

**Relay information**

**User id, pass**

**Sales**

**Require bills**

**4.**

**Create bill**

**Input bills**

**Relay information**

**Bills**

**Feedback**

**Require**

**Feedback**

**2. Check information**

**Check**

**Relay information**

**Products**

**Report**

**Inventory manager**

***Figure: DFD Level 1 For Sale***

**User**

**Retrieve information**

**Saving account**

**Products**

**Retrieve information**

**Save**

**1. Account management**

**3. Change price**

**Retrieve information**

**Require**

**show**

**Require**

**Require permission**

**2.**

**Authentification**

**Input**

**show**

**Roles**

**Admin**

**Feedback**

**Require**

**Show**

**4. Input categories**

**& products**

**Input show**

**Categories**

**Input**

**show**

**Products**

***Figure 3: DFD Level 1 for Administrator***

**Products**

**Retrieve information**

**Require**

**2. Check status**

**Retrieve**

**bills**

**Products**

**Retrieve**

**require**

**Require**

**User**

**Relay**

**Require**

**5. Validate sale**

**feedback**

**Check, change pass, id**

**3. Create report**

**Require**

**Relay information**

**1. Account process**

**Retriev**

**Require**

**relay information**

**User id, pass**

**4. Make inventory trends**

**Inventory manager**

**Require**

**Relay information**

***Figure 4: DFD Level 1 For Inventory management***

**Bills**

**Require bills Show**

**2. Check transaction**

**Distingue**

**Feedback**

**User**

**feedback**

**Check, change pass, id**

**3. Check exception**

**1. Account process**

**Check**

**exception Feedback**

**relay information**

**User id, pass**

**4.**

**Create**

**Sale-trends**

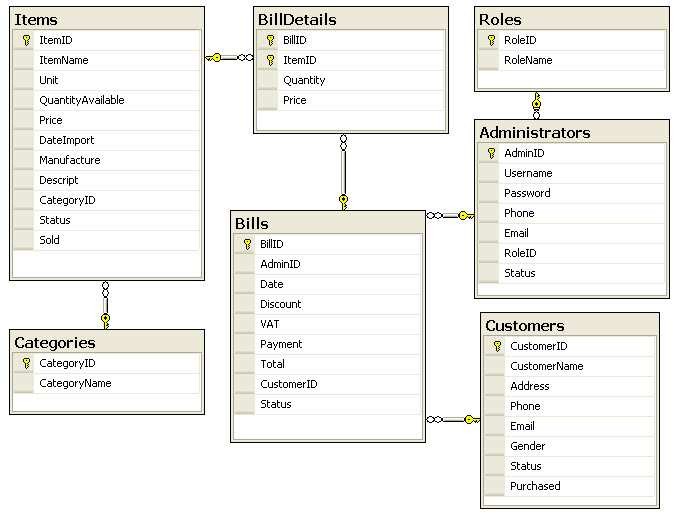
**Sales manager**

**Require trends**

**Feedback**

***Figure 5: DFD Level 1 for Sale Manager***

**Chapter 5: Entity Relationships(ER) Diagram**



**Chapter 6: Database Design/Structure**

**Table: Administrator**

**Field Name Date Type Null Key Description**

AdminID Varchar(5) No PK Identifier of administrator

Username Varchar(30) No Username for login of admin

Password Varchar(20) No Password for login of admin

Phone Nvarchar(20) Yes Phone number of admin

Email Nvarchar(50) Yes Email of admin

RoleID Varchar(5) No Role ID of admin

Status Bit Yes Status of admin

**Table: Roles**

**Field Name Data Type Null Key Description**

RollID Varchar(5) No PK Stores the RollId of User

RollName Nvarchar(30) No Stores RollName of User

**Table: Categories**

**Field Name Data Type Null Key Description**

Stores the CategoryId of

CategoryID Varchar(5) No PK CategoryName Nvarchar(30) No

Item

Stores the CategoryName of the ITem

**Table: Items**

**Field Name Data Type Null Key Description**

ItemID Varchar(5) No PK Stores ItemId of the Items

Stores Item Name of the

ItemName Nvarchar(25) No

Items

Unit Nvarchar(20) No Stores unit of the Items

Quantity

Available Int No CHK Stores quantity of items

Price Float No Stores price of Items

DateImport Datetime No Stores Date import of Items

Stores Manufacture of

Manufacture Nvarchar(50) No

Items

Descript Nvarchar(200) No Store descript of Items

CategoryID Varchar(5) No FK Stores Category Id of Items

Status Bit No Stores status of items

Sold Int Yes Quantity sold of item

**Table: Customer**

**Field Name Data Type Null Key Description**

CustomerID Varchar(5) No PK Stores Customer Id

CustomerName Nvarchar(50) No Stores customer Name

Address Nvarchar(200) No Store address of customer

Stores Identity Card of

IdentityCard Varchar(20) No

DateImport Datetime No

customer

Stores date import of customer

Status Bit No Stores status of customer

Store amount money this

Purchased Float Yes