INVENTORY MANAGEMENT SYSTEM FOR SRI SAI THIRUMALA WINES

A
Major Project (Stage-1) Thesis submitted
in the partial fulfillment of the requirements for the
award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

by

K. SATHWIKA	**19281A0588**
K. POOJITHA	**19281A05A1**
Y. VASANTH	**19281A0589**
Ch. DIVYA VANI	**19281A05B3**

PROJECT GUIDE

Mr. DEEPAK GUJJULA

(Assistant Professor, CSE)

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad, Accredited by NAAC with B++ and NBA (CSE,ECE,EEE))Singapur, Huzurabad, Karimnagar,Telangana

2022 – 2023

KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Sponsored by Kamala Education Society, Approved by AICTE, New Delhi,
Affiliated to JNTU, Hyderabad, Accredited by NAAC with B++ and NBA
(CSE,ECE,EEE))Singapur, Huzurabad, Karimnagar, Telangana-505468.

CERTIFICATE

This is to certify that **K. SATHWIKA(19281A0588), K.POOJITHA(19281A05A1), Y.VASANTH (19281A0589), Ch. DIVYA VANI (19281A05B3)**of the B.Tech Computer Science & Engineering has satisfactorily completed the dissertation work for Mini project entitled**“INVENTORY MANAGEMENT SYSTEM FOR SRI SAI THIRUMALA WINES”** towards the partial fulfillmentofB.Tech degree in this academic year i.e:2022-2023.

Project Guide Head of the Department

Principal External Examiner

INDEX

ACKNOWLEDGEMENT

The success of any course depends mostly on the teachers who teach me. Only good teaching can interpret the syllabus and produce desirable changes and competent citizens. This one was a team effort and many people whose names do not appear on the cover deserve credit. First, I thank God almighty for his manifold mercies in carrying out our project successfully.

We heart fully thank our principal, **Dr. K. SHANKER,** for providing all the resources in completing my project.

We thank our Head of the Department of Computer Science and Engineering **Dr. Capt. RAVINDRA BABU KALLAM**, for encouraging me in doing real time projects.

We thank our project Coordinator **Mr. J. PAVAN KUMAR,** Assistant Professor in CSE for their continuous support in completion of this project.

We would like to thank our project guide **MR. G. DEEPAK,** Assistant Professor in CSE for the valuable guidance and constant encouragement at each stage of this project

We also thank other teaching and non-teaching staff of Kamala Institute of Technology and Science, for supporting me in every stage of my mini project, entitled **“INVENTORY MANAGEMENT SYSTEM FOR SRI SAI THIRUMALA WINES.”**

We thank my Parents and Friends for their moral support throughout the project work that helped to strengthen my will.

K. SATHWIKA	**19281A0588**
K. POOJITHA	**19281A05A1**
Y.VASANTH	**19281A0589**
CH. DIVYAVANI	**19281A05B3**

ABSTRACT

The aim of this project is to develop a web application for managing the inventory in which all the information regarding the stock will be presented. Here management currently uses a manual system that is natively recorded by hand by their employees. This may sometimes lead to the situation where the worker let slip to update the stock and will cause time waste as a result it may lead to risk in maintaining data, documents and anything related to the daily transaction. The proposed system is computerized and uses HTML, CSS, and python therefore it gives more facilities and is user friendly. To overcome the problems in our application we have an admin, employee and accountant module. Where, in the admin module he can manage by updating new brands, price and adding banking details, manages employees etc and in the employee module he manages the daily sale of stock, available goods, update with the new stock details and in the accountant module he can view the daily sheet and bill details for his respective work.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 ABOUT PROJECT:

Inventory Management System is a software application which fulfills the requirement of a typical Stock Analysis in a warehouse. 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 reflect immediately. This is an online tool so more than one user can login into the 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. Also this application provides an interface to users to view the details like the daily Stock Statements of the shop.

1.2 EXISTING SYSTEM WITH DRAWBACKS:

Current system is a manual one in which employees are maintaining ledgers, books etc to store the information like supplier details, inwards, deliveries and returns of items.

Drawbacks (Problems):

In the case of the existing system, the management has to face a lot of problems. Few of them as follows:

- Daily transactions are to be entered into different books immediately to avoid conflicts which are very difficult.
- It is tedious to manage previous data which needs much space to keep all the previous year’s ledgers, books etc.
- More manual hours need to generate required reports.

1.3 PROPOSED SYSTEM WITH FEATURES:

Web application for the Sri Sai Thirumala Wines has been designed in which details regarding different categories such as maintaining stock details, employee details, daily transaction details etc.

Features:

- This web application consists of the details and description about the different brands and stock.
- Minimize administration efforts.
- This inventory eliminates paperwork, human faults, manual delay and speed up process.
- Easy to manage all the daily transactions.
- Can generate required reports easily.
- Easy to use GUI that does not require specific training.
- Easy to manage previous data in a secure manner.
- Centralized database helps in avoiding conflicts.

CHAPTER 2

LITERATURE SURVEY

Inventory management practice is the type of activities that an organization follows in the process of ordering, issuing, storing and trading inventory products held by that organization.Inventory is materials, parts provided, and materials in the process that are contained in the company for the production process and finished goods or products provided to meet demands from consumers or customers every time they are stored and maintained. It is considered a specific rule in the Inventory so that it is always ready to use and recorded in the form of company books.

CHAPTER 3

ANALYSIS

The goal of system analysis is to determine where the problem is in an attempt to fix the system. This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

3.1 HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements (Preferable):

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

PROCESSOR	i3 or above
RAM CAPACITY	2GB or above
HARD DISK	128GB or above
PROCESSOR SPEED	1.1 GHz
SYSTEM TYPE	32-bit (or)64-bit

3.1.1 Hardware Requirements

Software Requirements:

One of the most difficult tasks is selecting the software, once a system requirement is known that is determining whether a particular software package fits the requirements.

FRONT END	HTML, CSS, JavaScript
OPERATING SYSTEM	Windows 7 or Higher
DATABASE	MySQL
UML PLATFORM	STAR UML
WEB SERVER	XAMPP
SCRIPTING LANGUAGE	Python (Django)

3.1.2 Software Requirements

3.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS:

Functional Requirements:

Functional Requirements are associated with specific functions, tasks or behaviors of the system. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements. A task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement.

The following are the functional requirements of our project. They are:

- A database has to be created to maintain the details of employees, accountant, daily sheet, challan, warehouse and price details.
- The information should be updated whenever required.
- It should be very efficient in retrieval and updating of information.
- It shouldn't accept null values whenever validated.

Non-Functional Requirements:

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions.

Following are the non-functional requirements:

- **Maintainability**: Maintainability is used to make future maintenance easier, meet new requirements.
- **Robustness:** Robustness is the quality of being able to withstand stress, pressures or changes in procedure or circumstance.
- **Reliability:** Reliability is an ability of a person or system to perform and maintain its functions in circumstances.
- **Size:** The size of a particular application plays a major role, if the size is less then efficiency will be high.
- **Speed:** If the speed is high then it is good. Since the number of lines in our code is less, hence the speed is high.

3.3 MODULE DESCRIPTION:

The project has been divided into these modules:

- Admin
- Employee
- Accountant

Admin: Admin should login by providing the valid username and password which is predefined. The admin role is to add employees and accountants. He can also update the price and new brands, and can add the challan details to the website.

Employee: Employee should login by providing the valid username and password. Employees can view the price updates, new brand details, warehouse stock, and he enters the daily stock details in the daily sheet.

Accountant: Accountant can login by providing the valid username and password. He can perform operations such as viewing the daily sheet and challan details.

CHAPTER 4

DESIGN

4.1 BLOCK DIAGRAM:

A block diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering in hardware design, electronic design, software design, and process flow diagrams.

Block diagrams are typically used for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for details of implementation. Contrast this with the schematic diagrams and layout diagrams used in electrical engineering, which show the implementation details of electrical components and physical construction.

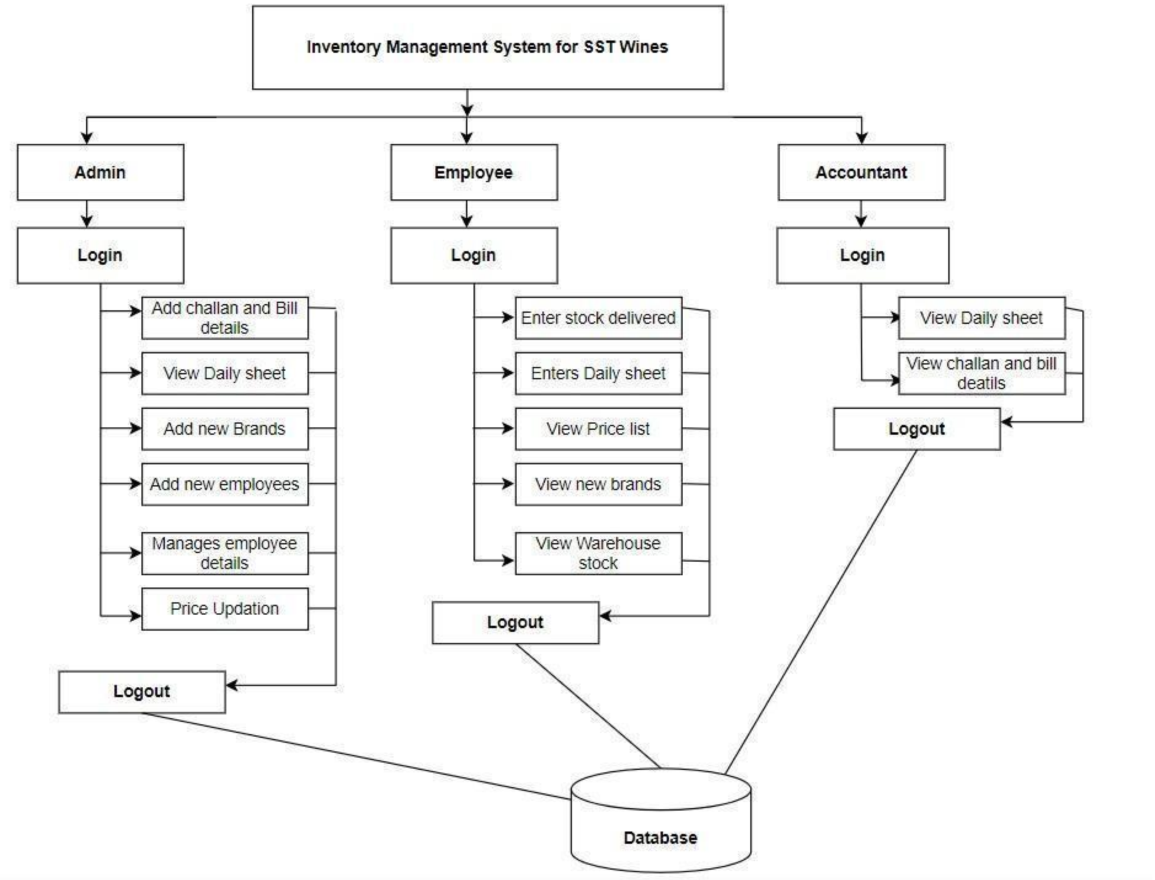


Figure 4.1 Block diagram

4.2 DATA-FLOW DIAGRAMS

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD [1] provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.

It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored. It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modeled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled [2]. The Level 1 DFD shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system. Data-flow diagrams were invented by Larry Constantine, the original developer of structured design, based on Martin and Estrin's "data-flow graph" model of computation. Data-flow diagrams (DFDs) are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system's data flow diagrams can be drawn up, compared with the new system's data-flow diagrams to draw comparisons to implement a more efficient system. Data-flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report.

There are different notations to draw data-flow diagrams, defining different visual representations for processes, data stores, data flow, and external entities. Data flow diagrams ("bubble charts") are directed graphs [3] in which the nodes specify processing activities and the arcs specify data items transmitted between processing nodes.

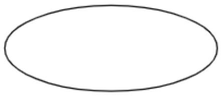
DFD Symbols :

In the DFD, there are four symbols

- A square defines a source (originator) or destination of system data.
- An arrow identifies data flow. It is the pipeline through which the information flows.
- A circle or a bubble represents a process that transforms incoming data flowinto outgoing data flows.
- An open rectangle is a data store, data at rest or a temporary repository of data.

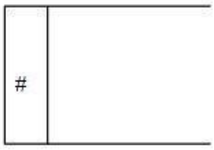
Dataflow: Data moves in a specific direction from an origin to a destination.

Process: People, procedures or devices that use or produce (transform) data. The physical component is not identified.



Sources: External sources or destination of data,which may be programs, organizations or other entities.

Data stores: Here data is stored or referenced by a process in the system.



Constructing a DFD:

Several rules of thumb are used in drawing DFD'S:

- Processes should be named and numbered for an easy interface. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flows from source to the destination although they may flow back to the source. One way to indicate this is to draw a long flow line back to a source.
- An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

4.2.1 Context level DFD

Context level DFD A Context level Data flow diagram created using select structured systems analysis and design method (SSADM). This level shows the overall context of the system and its operating environment and shows the whole system as just one process. It does not usually show data stores, unless they are owned by external systems, e.g. are accessed by but not maintained by this system, however, these are often shown as external entities. The Context level DFD is shown in fig.4.2.1

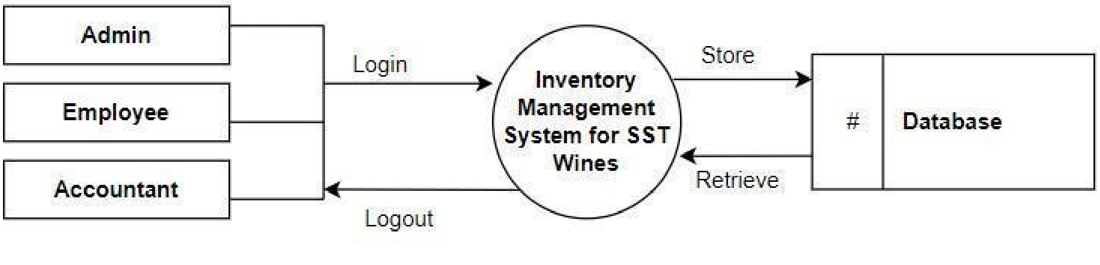


Figure 4.2.1 Context level Data Flow Diagram(DFD)

4.2.2 Top level DFD

A data flow diagram is that which can be used to indicate the clear progress of a business venture. In the process of coming up with a data flow diagram, the level one provides an overview of the major functional areas of the undertaking. After presenting the values for most important fields of discussion, it gives room for level two to be drawn.

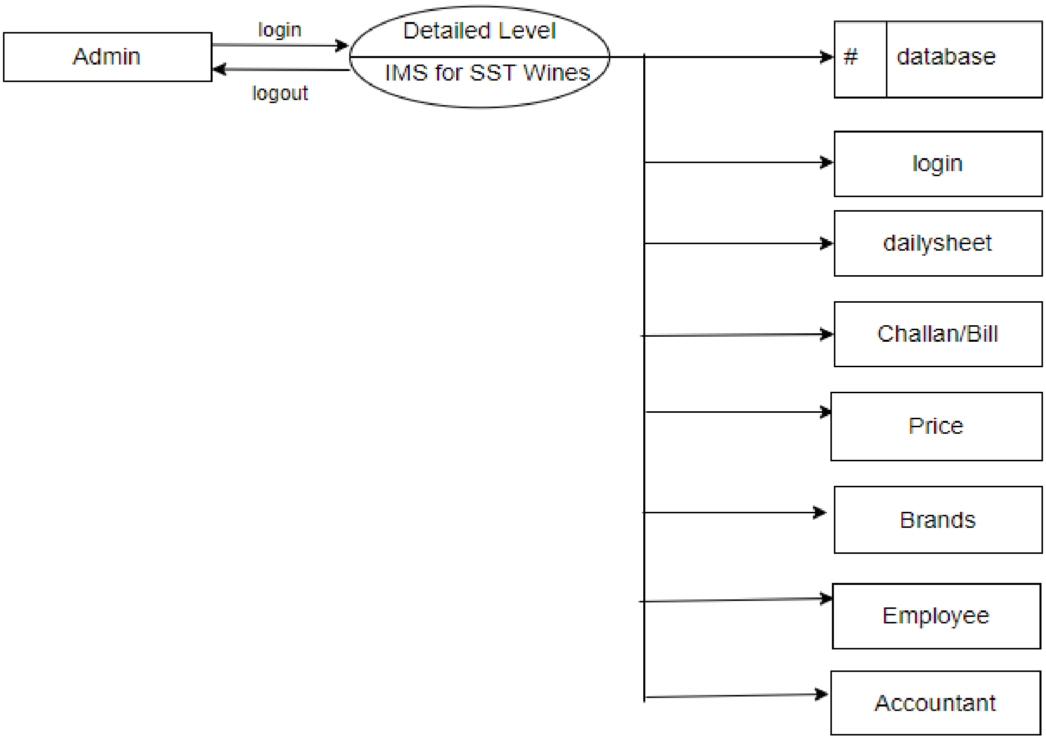


Figure 4.2.2.1 Top level data flow diagram for admin

Admin should login by providing the valid username and password. All the operations which are displayed in the above figure such as adding, viewing and deleting can be performed and all these are linked to the database. It is shown in the above figure 4.2.2.1

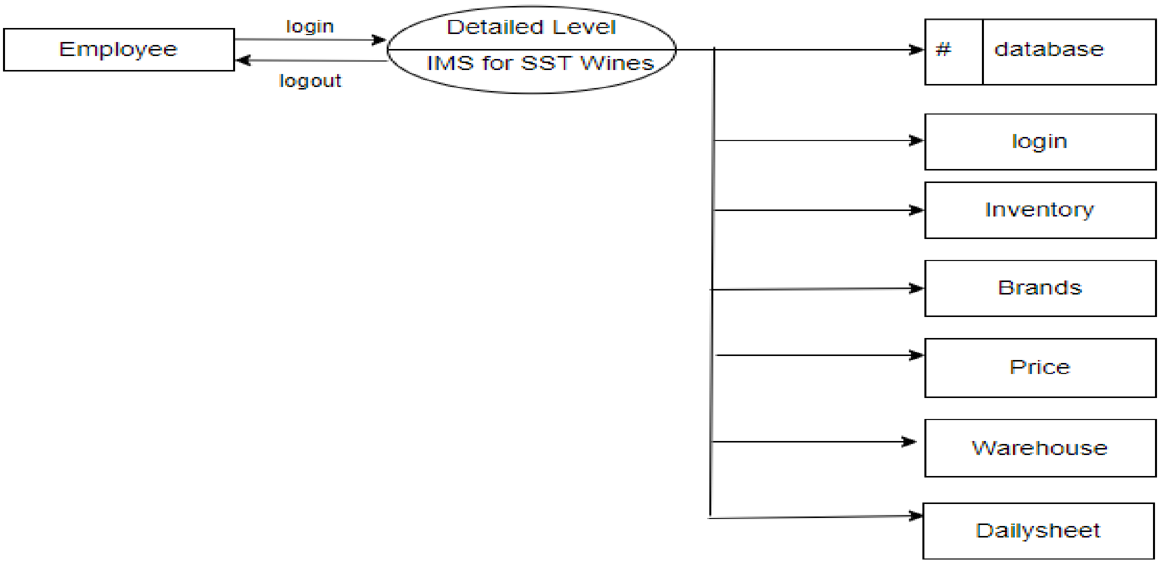


Figure 4.2.2.2 Top level data flow diagram for employee

Employees should login by providing the valid username and password. All the operations which are displayed in the above figure such as adding ,viewing can be performed and all these are linked to the database. It is shown in the above figure 4.2.2.2

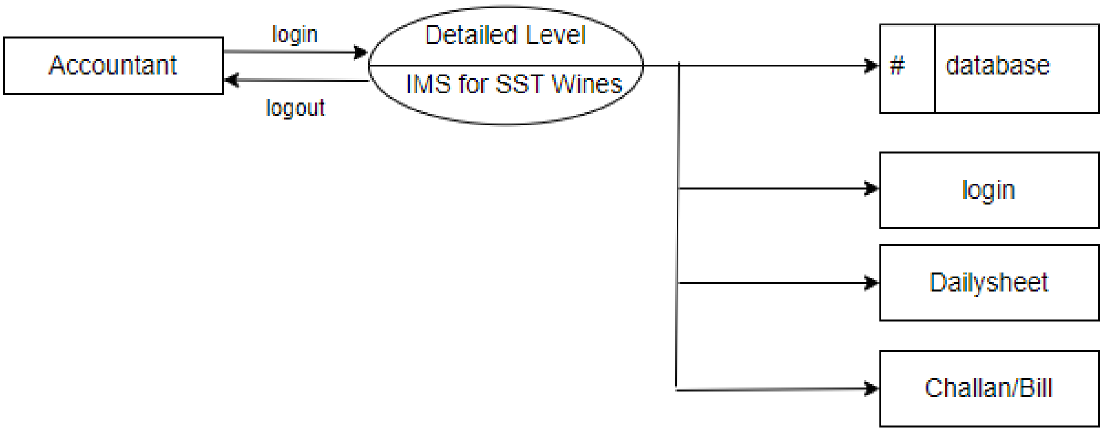


Figure 4.2.2.3 Top level data flow diagram for customer

Accountants should login by providing the valid username and password. All the operations which are displayed in the above figure such as viewing daily sheet and challan details can be viewed and all these are linked to the database. It is shown in the above figure 4.2.2.3

4.2.3 Detailed level DFD

A detailed level data flow diagram shows the overall context of the system and its operating environment and shows the whole system in detailed form, i.e. by seeing the DFD we can say what happens after each and every step. The purpose of this level is to show the major and high-level processes of the system and their interrelation. A process model will have one, and only one, decomposed in lower-level DFD. A decomposed in lower-level DFD diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the decomposed in lower-level DFD.

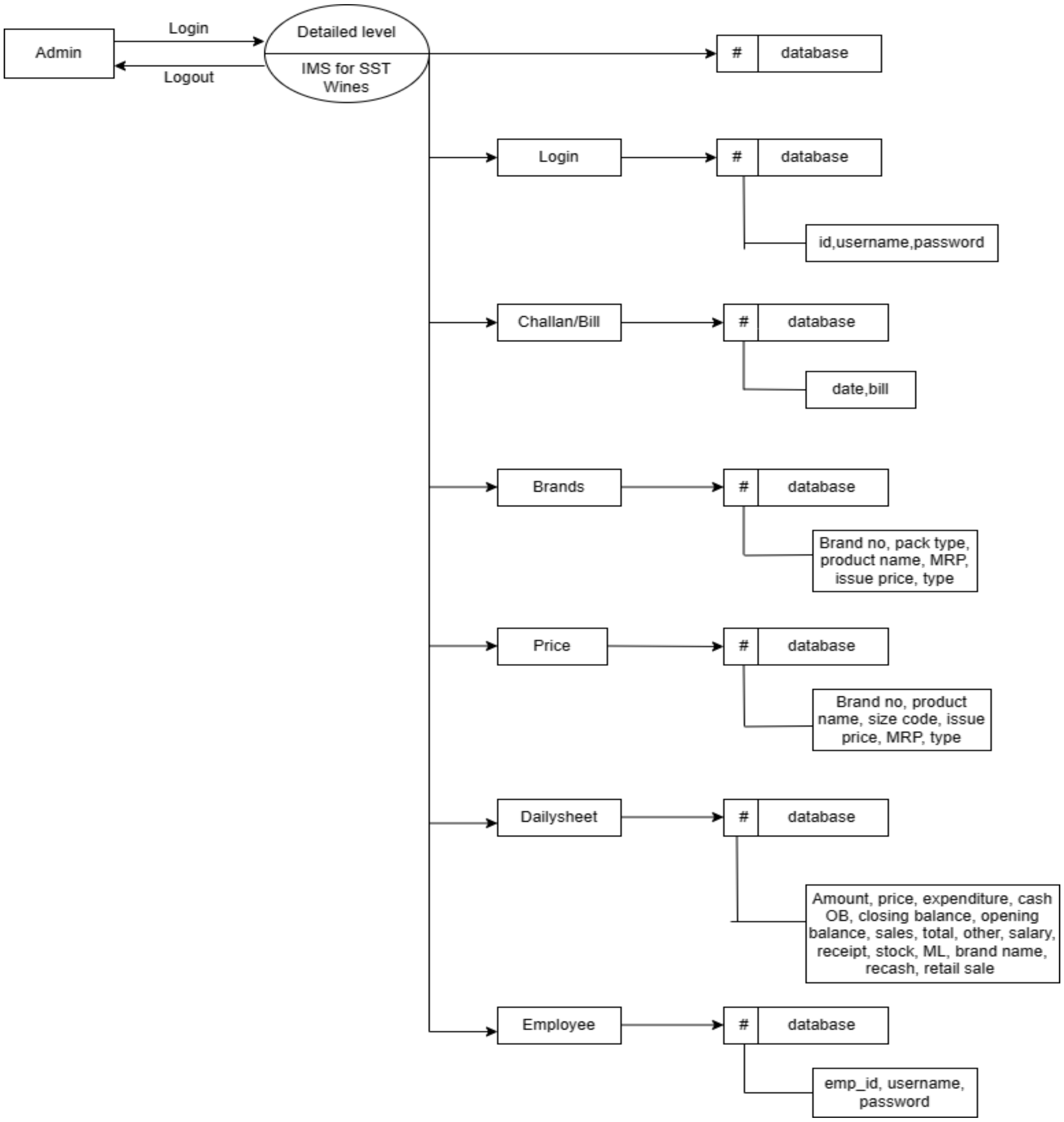

Admin
Login
Logout
Detailed level
IMS for SST Wines
#
database
Login
id,username,password
Challan/Bill
date,bill
Brands
Brand no, pack type, product name, MRP, issue price, type
Price
Brand no, product name, size code, issue price, MRP, type
Dailysheet
Amount, price, expenditure, cash OB, closing balance, opening balance, sales, total, other, salary, receipt, stock, ML, brand name, recash, retail sale
Employee
emp_id, username, password

Figure 4.2.3.1 Detailed level data flow diagram for admin

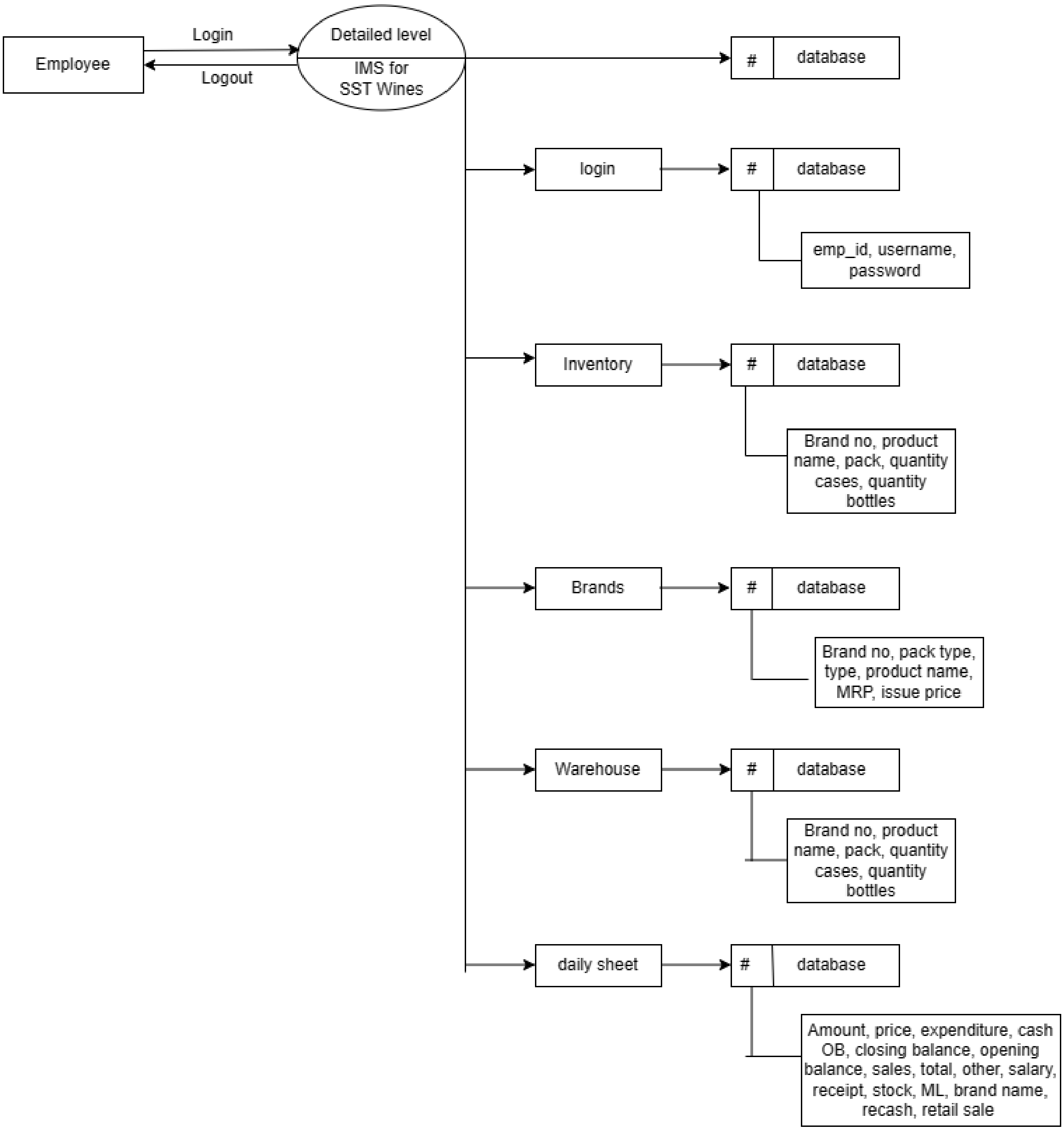

Employee
Login
Logout
Detailed level
IMS for SST Wines
#
database
login
#
database
emp_id, username, password
Inventory
#
database
Brand no, product name, pack, quantity cases, quantity bottles
Brands
#
database
Brand no, pack type, type, product name, MRP, issue price
Warehouse
#
database
Brand no, product name, pack, quantity cases, quantity bottles
daily sheet
#
database
Amount, price, expenditure, cash OB, closing balance, opening balance, sales, total, other, salary, receipt, stock, ML, brand name, recash, retail sale

Figure 4.2.3.2 Detailed level data flow diagram for employee

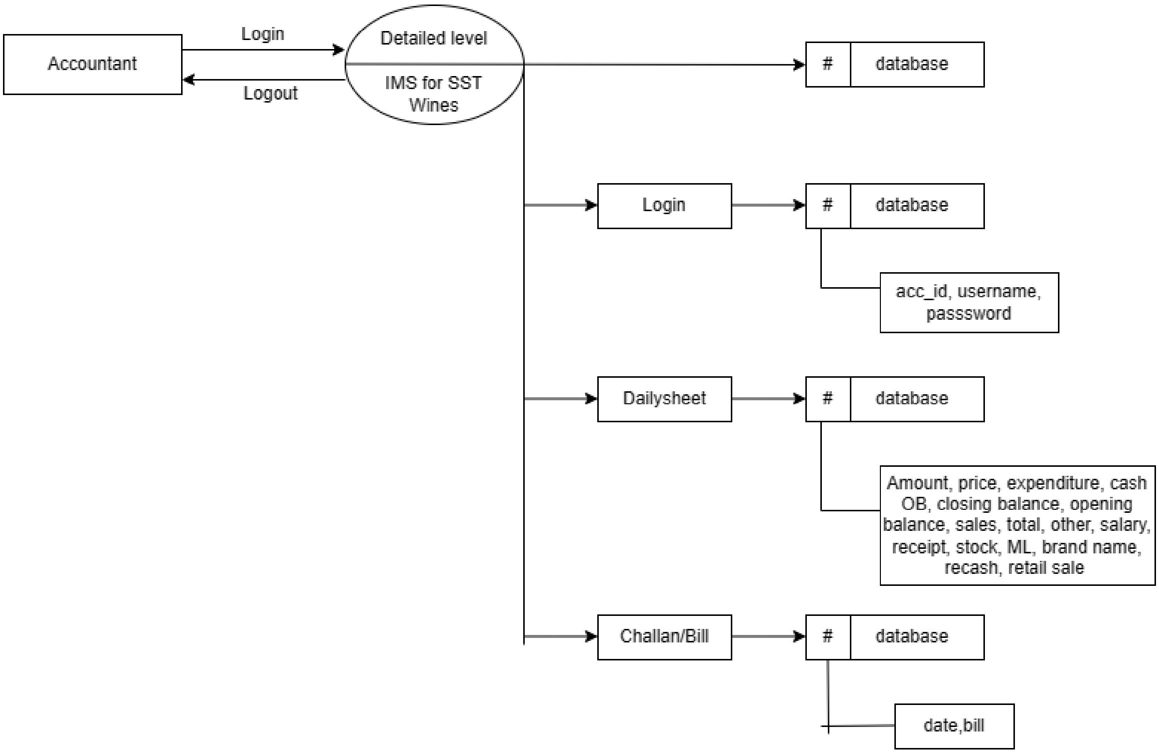

Accountant
Login
Logout
Detailed level
IMS for SST Wines
#
database
Login
acc_id, username, passsword
Dailysheet
Amount, price, expenditure, cash OB, closing balance, opening balance, sales, total, other, salary, receipt, stock, ML, brand name, recash, retail sale
Challan/Bill
date,bill

Figure 4.2.3.3 Detailed level data flow diagram for accountant

4.3 ER DIAGRAM:

In software engineering, an entity-relationship model (ER model) is a data model for describing data or information aspects of a business domain or its process requirements. In an abstract way the main components of ER model are entities and the relationships that can exist among them.

Elements in ER diagram

There are three basic elements in an ER Diagram

- Entity
- Attribute
- Relationship

There are more elements which are based on the main elements [4]. They are weak entities, multi valued attributes, derived attributes, weak relationships and recursive relationships.

Cardinality is one of the notations used in ER diagrams.

Entity: An entity can be a person, place, event or object that is relevant to a given system.

They are represented by a rectangle and named using nouns.

Weak Entity: A weak entity is an entity that depends on the existence of another entity.

It can be defined as an entity that cannot be identified by its own attributes.

Attribute: An attribute is a property, or characteristic of an entity, relationship, or another attribute.

Multi valued Attribute: If an attribute can have more than one value .

Derived Attribute: An attribute derived from another attribute.

Relationship: A relationship describes how entities interact.

Cardinality: Cardinality specifies how many instances of an entity relate to one instance of another entity. Cardinality specifies the maximum number of relationships.

Entities and their attributes

The following figure 4.3 shows ER Diagram

Admin: id(Primary Key),username, password.

Employee: emp_id (Primary key), username, email, password, c_password, mobile, dob.

Accountant: acc_id (Primary key), username, email, password, c_password, mobile, dob.

Brands: Brand_no (Primary key), product name, pack type, issue price, MRP, type.

Bill: date, bill

Inventory: Brand number, brand name, size, quantity bottles, quantity cases

Price: Brand number, brand name, size, issue price, MRP, type.

Warehouse: Brand number, brand name, size, quantity bottles, quantity cases

Daily sheet: date(Primary key), shop name, emp name, brand name, ML, OB, receipt, ob_total, other, stock, CB, sales, price, amount, retail sale, expenditure, balance, cash OB, total, re cash, hand cash.

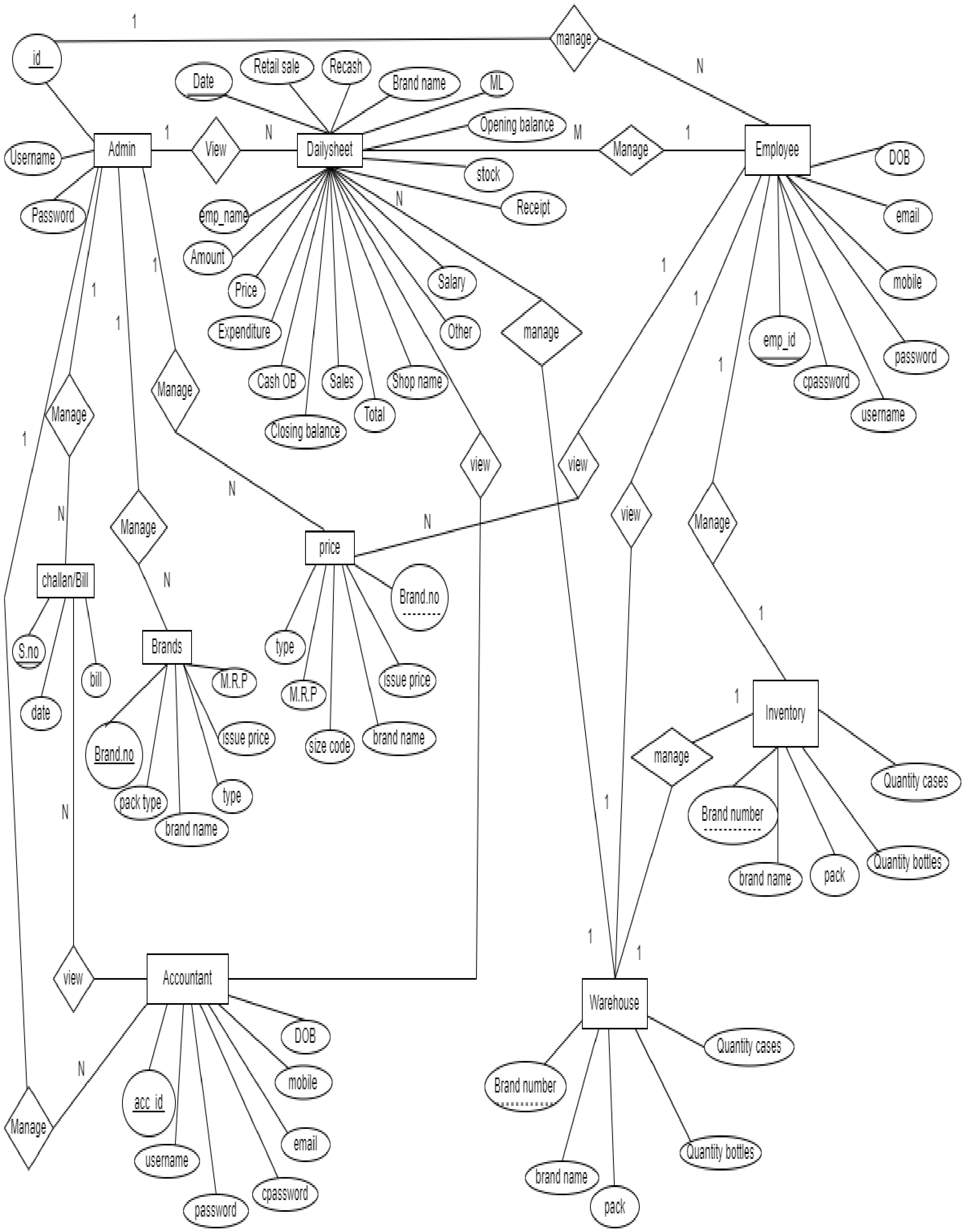


Figure 4.3 Entity Relationship diagram

4.4 UML DIAGRAMS:

The Unified Modeling Language (UML) is a Standard language for specifying,visualizing, constructing and documenting the software system and its components . The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. Structural models represent the framework for the system and this framework is the place where all other components exist. So, the class diagram, component diagram and deployment diagrams are the part of structural modeling.

They all represent the elements and the mechanism to assemble them. But the structure of the model never describes the dynamic behavior of the system. Behavioral model describes the interaction in the system. It represents the interaction among the structural diagrams[5]. Behavioral modeling shows the dynamic nature of the system.Architectural model represents the overall framework of the system. It contains both structural and behavioral elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modeling.

The Unified Modeling Language encompasses a number of models

- Use Case diagram
- Class diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram

4.4.1 Use Case Diagram:

Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems). Use case diagrams are central to modeling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations. The key points are:

- The main purpose is to show the interaction between the use cases and the actor.
- To represent the system requirement from the user's perspective.
- Use cases are the functions that are to be performed in the module.

The following figure 4.4.1.1 shows the use case diagram for Admin. The admin will login using his username and password, after login he can manage employee, accountant details. He can also manage the price updates and adding of new brands.

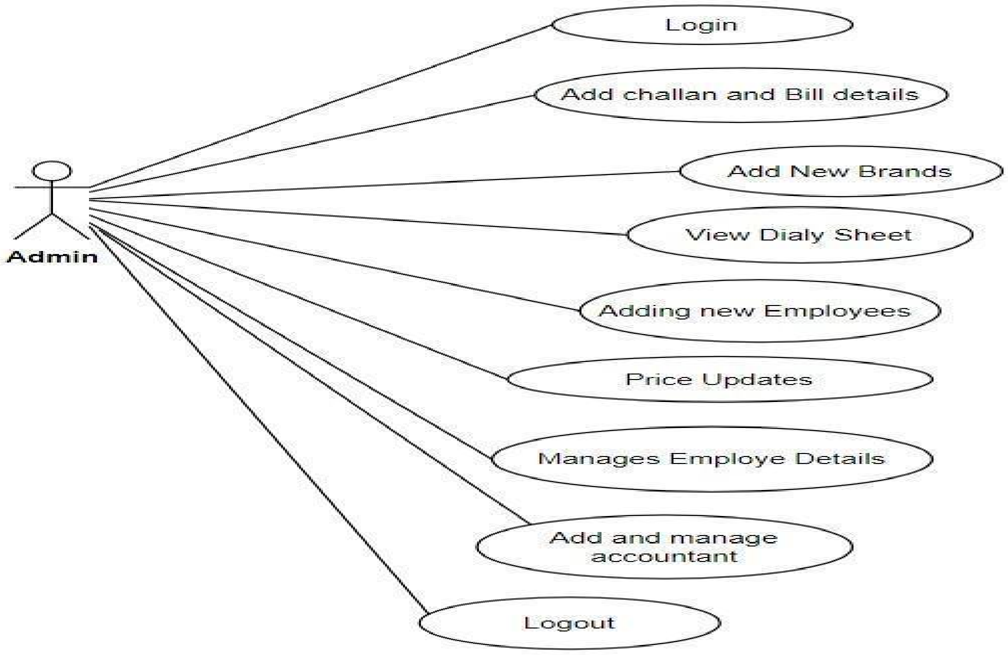


Figure 4.4.1.1 Use case diagram for admin

The following figure 4.4.1.2 shows the use case diagram for employees. The employee will login using his username and password, after login he can manage the stock delivered (on the day), entering the daily transactional sheet. He can view the price list(updated),and new brands to the shop.

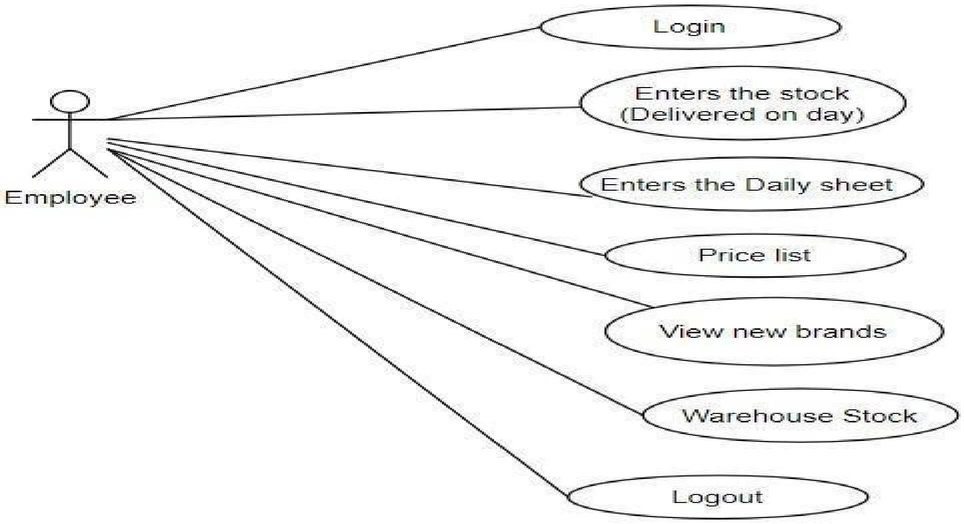


Figure 4.4.1.2 Use case diagram for employee

The following figure 4.4.1.3 shows the use case diagram for accountants . The accountant will login using his username and password, after login he can view the daily sheet and challan details.

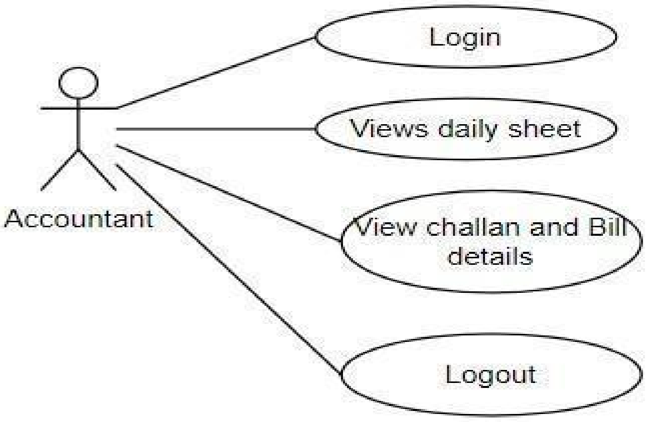


Figure 4.4.1.3 Use case diagram for accountant

4.4.2 Class Diagram:

A “Class Diagram” shows a set of classes, interfaces and collaborations and their relationships.These diagrams are the most common diagrams in modeling object-oriented systems. The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

The below Figure 4.4.2 shows the methods that are used in this class diagram and

The description of each class is listed below:

- **Admin:**This class is used for getting employee, accountant, inventory, daily sheet, warehouse, brands, price list, challan and bill details from the database and also store the employee and accountant details to the database.

- **Employee:** This class is used for entering daily stock details in the daily sheet, and can view the price list updates and new brands, warehouse stock.
- **Accountant:**This class is used for viewing daily sheets and challan details.
- **Inventory:** Thisclass is used for storing details of quantity stock.
- **Daily sheet:**This class is used for storing daily stock details.
- **Warehouse:**This class is used for storing the details of stock.
- **Brands:**This class is used for storing the information regarding brands.
- **Price list:**This class is used for storing the price list.
- **Challan and bill:**This class is used for storing the challan and bill.

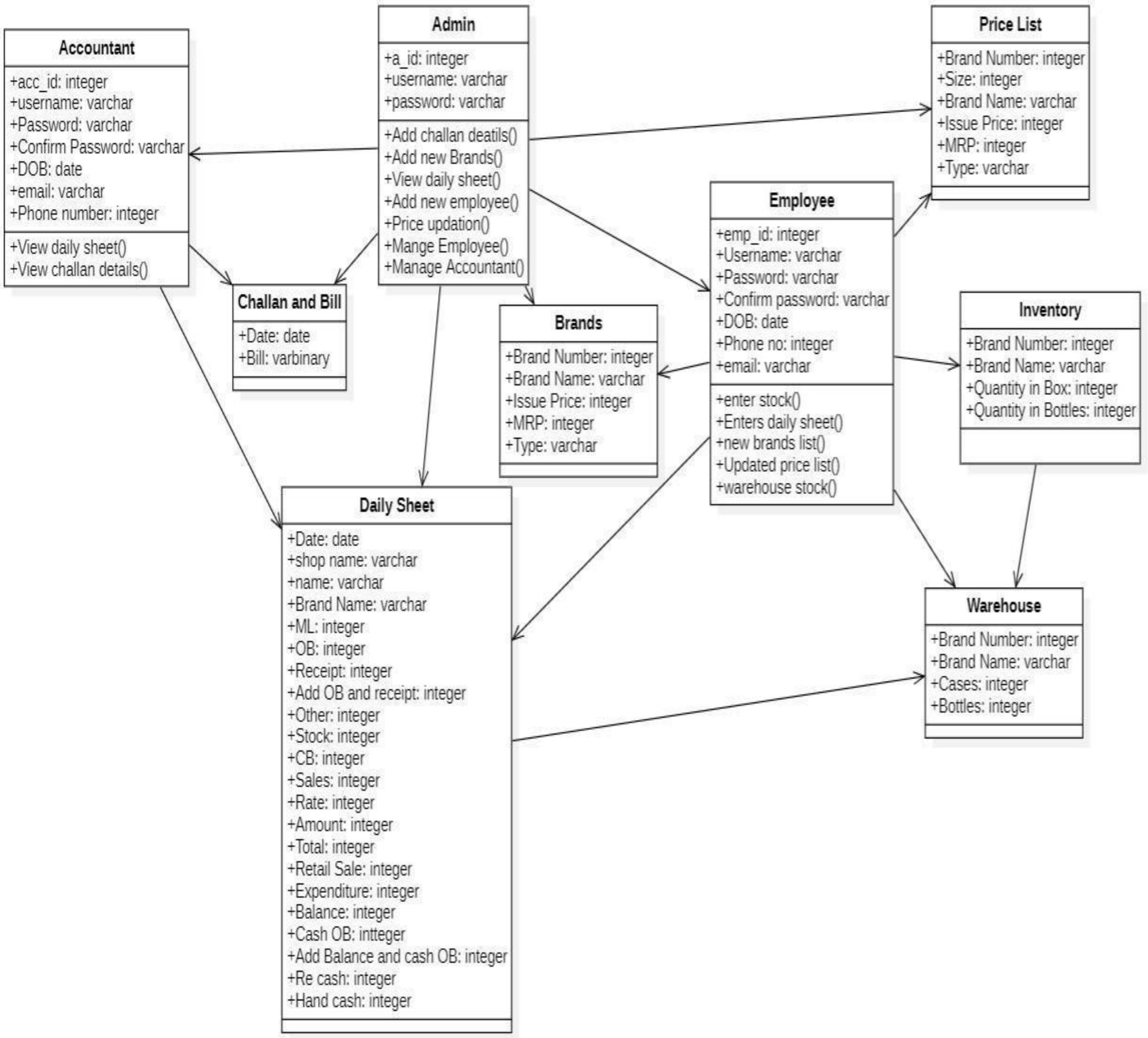


Figure 4.4.2 **Class diagram**

4.4.3 Sequence Diagram:

Sequence diagram is an interaction diagram which focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system.

The key points are:

1.The main purpose is to represent the logical flow of data with respect to a process.

2.A sequence diagram displays the objects and not the classes.

Below figure 4.4.3.1 shows the admin operations. The admin first enters login credentials then the back-end takes actions. If the login credentials are matched with the database then the admin is successfully logged into the site.Here the admin can perform operations like managing employee and accountant details and also prices and brands.

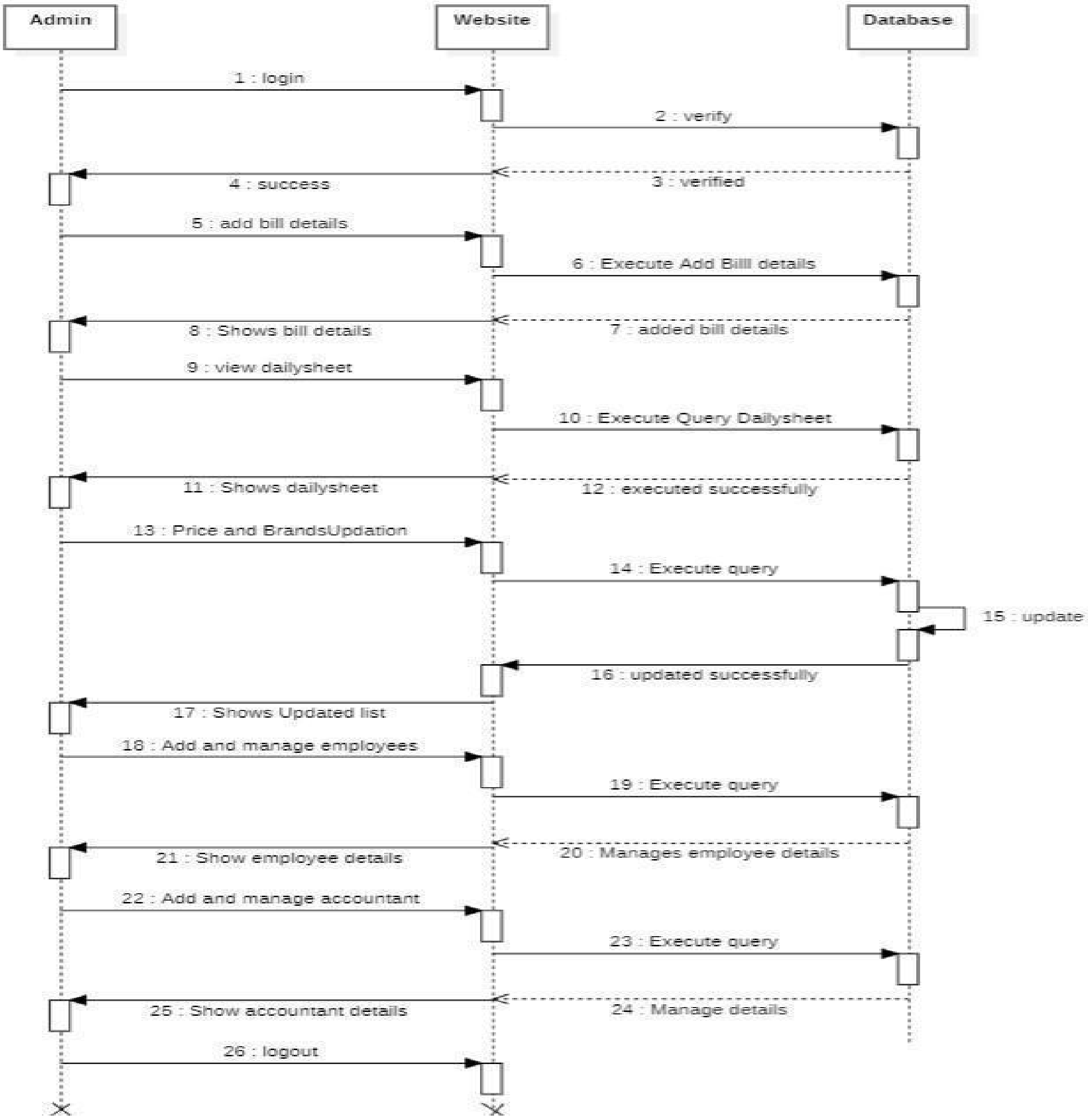


Figure 4.4.3.1Sequence diagram for admin

Figure 4.4.3.2 shows the employee operations. Here he can add the stock delivery on the day,mages the daily sheet and viewing the price list(updated),new brands.

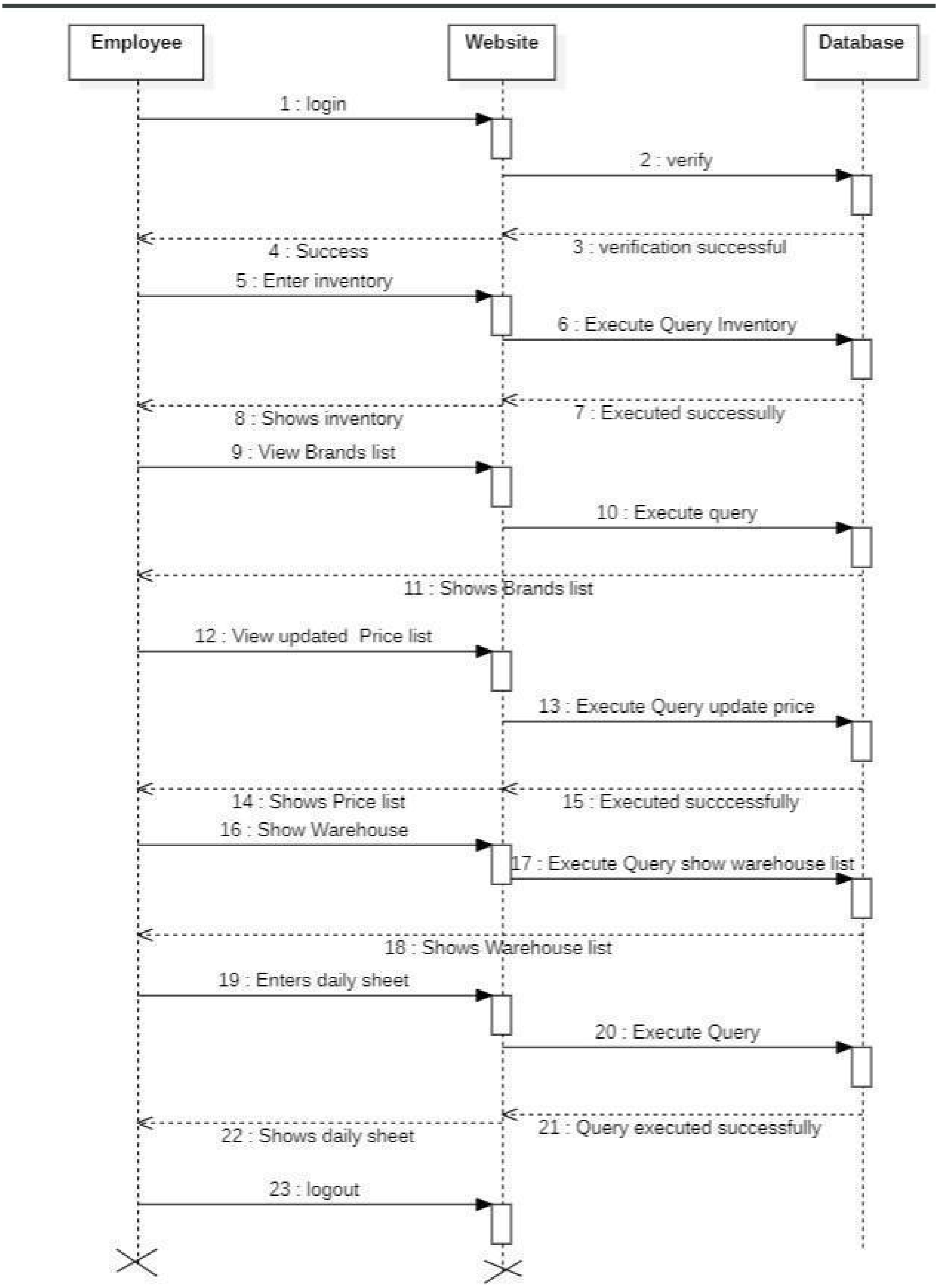


Figure 4.4.3.2Sequence diagram for employee

Figure 4.4.3.3 shows the accountant operations. Here the sequence flows after the accountant enters the correct username and password, he will be directed to the page where can view the daily sheet and challan deatils .

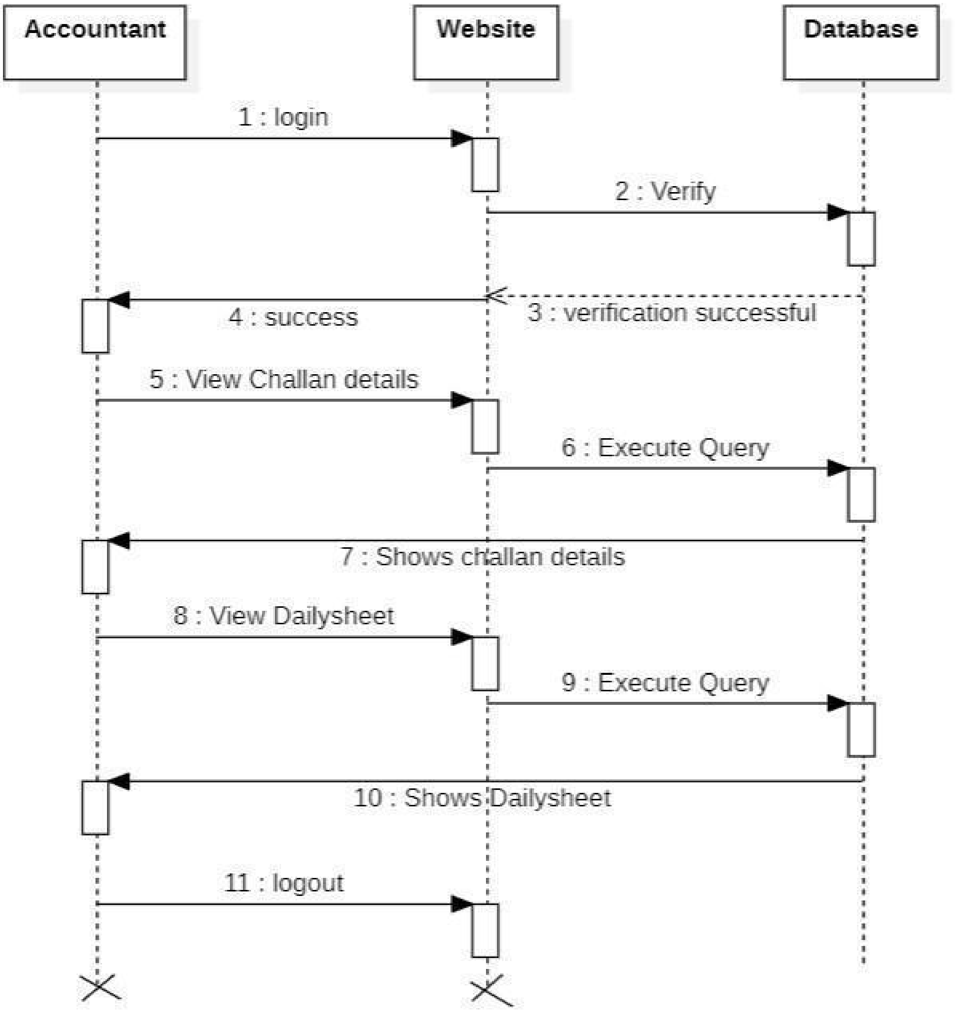


Figure 4.4.3.3Sequence diagram for accountant

4.4.4 Collaboration diagram:

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. It is also called a communication diagram. Collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other. Collaboration diagrams convey the same information as sequence diagrams, but they focus on object roles instead of times that messages are sent. In the sequence diagram, object roles are the vertices and messages are the connecting links. The object-role rectangles are labeled with either class or object names (or both).

Figure 4.4.4.1 shows the collaboration diagram of the admin. Admin, database, website are the main objects. Admin login to the website and information can be stored and retrieved from the database.

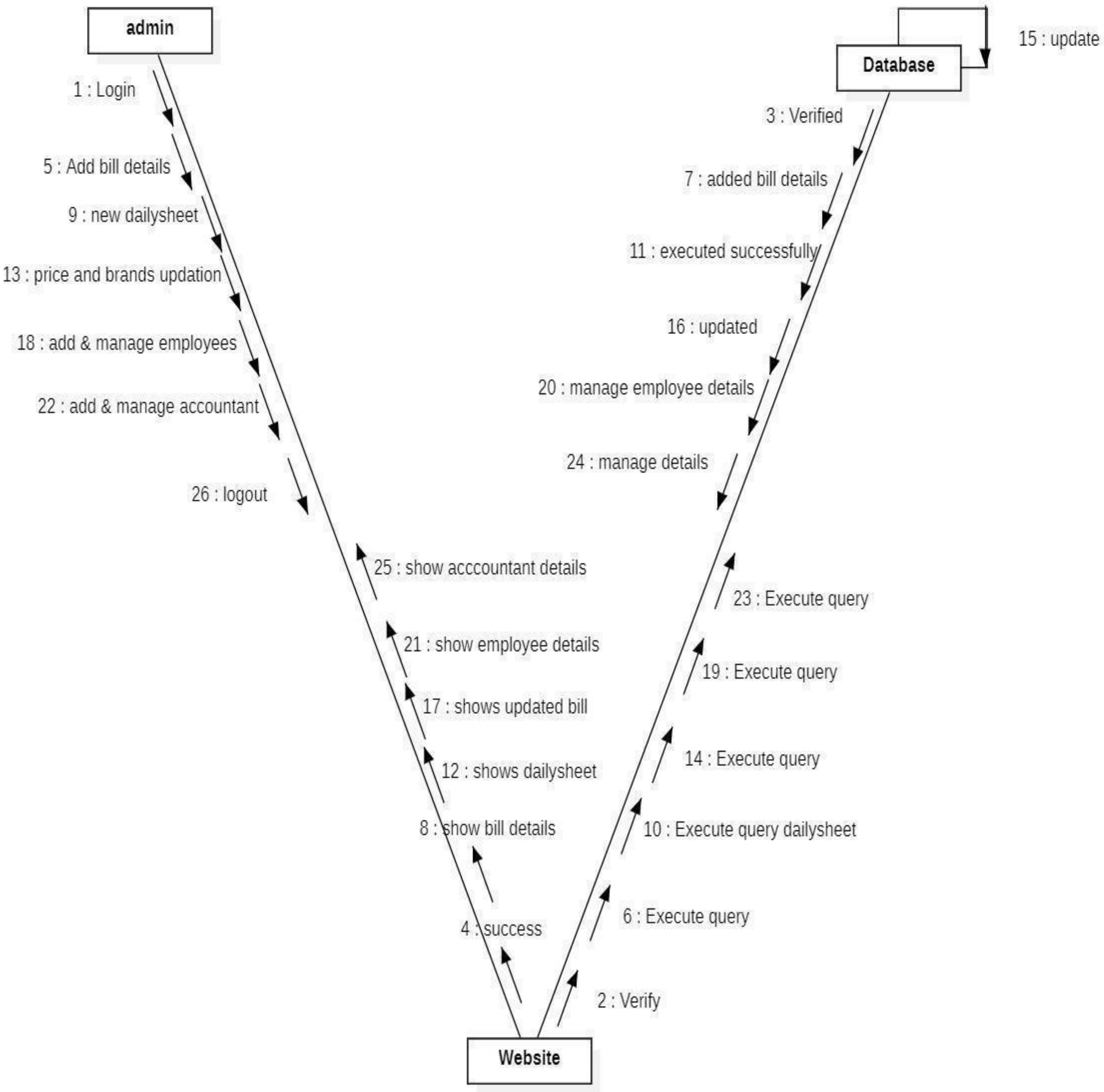


Figure 4.4.4.1Collaboration diagram for admin

Figure 4.4.4.2 shows the collaboration diagram of the employee. Employees will login to the website and perform view and store operations.

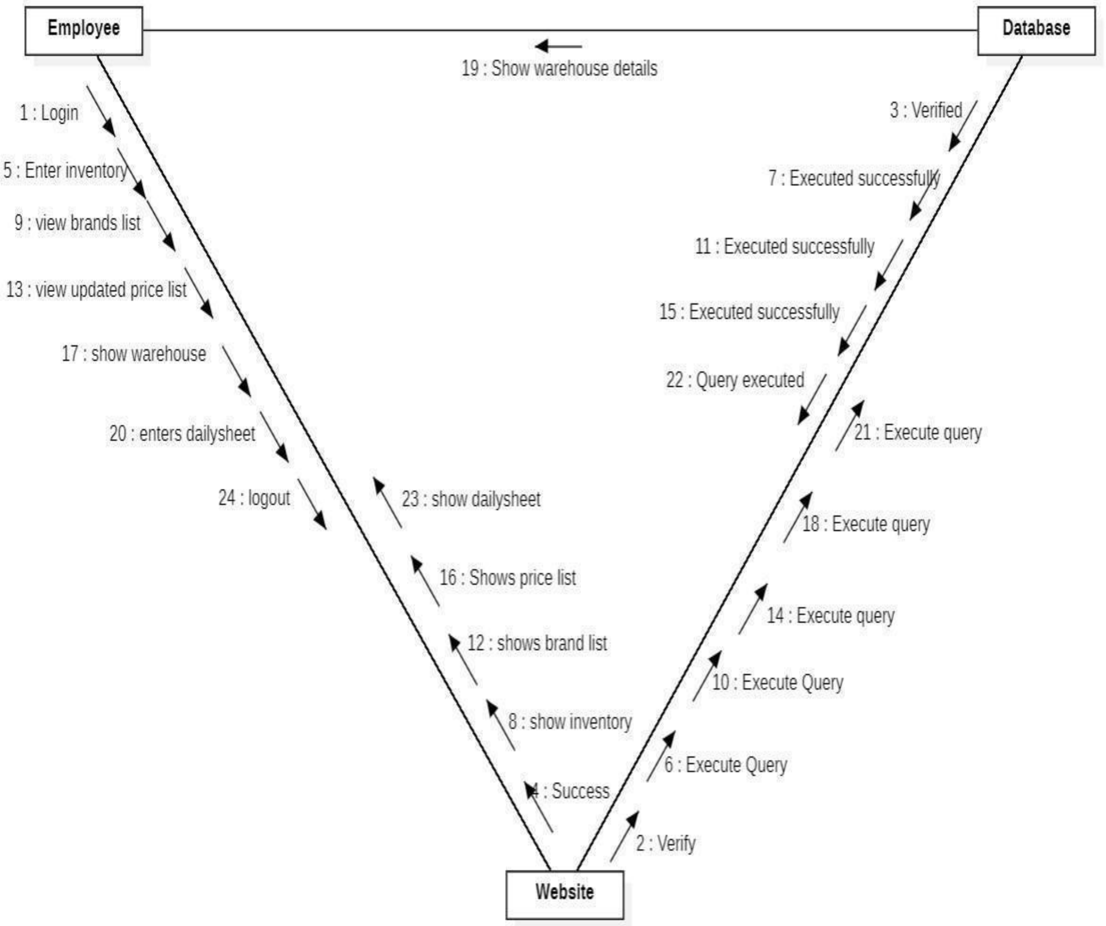


Figure 4.4.4.2**Collaboration diagram for employee**

Figure 4.4.4.3 shows the collaboration diagram of the accountant. Accountant logins with his username and password, and views the daily sheet and challan details.

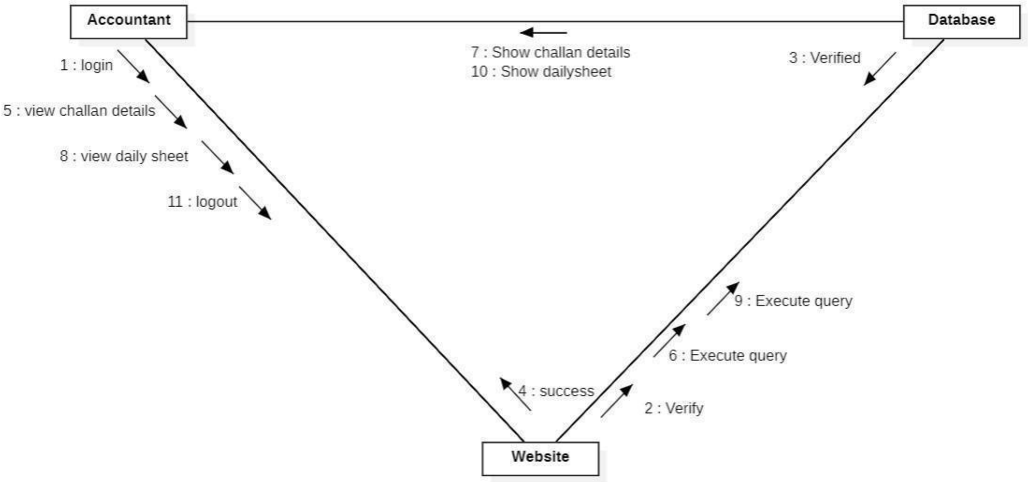


Figure 4.4.4.3**Collaboration diagram for accountant**

4.4.5 Activity Diagram:

An Activity diagram shows the flow from activity to activity within a system; it emphasizes the flow of control among objects.

The admin can perform activity actions as shown in Figure 4.4.5.1 which is the activity diagram for admin.

Admin tries to login with his username and password, after the successful login he can manage employee , accountant details and price updates ,adding new brands. And he can also add new employees.

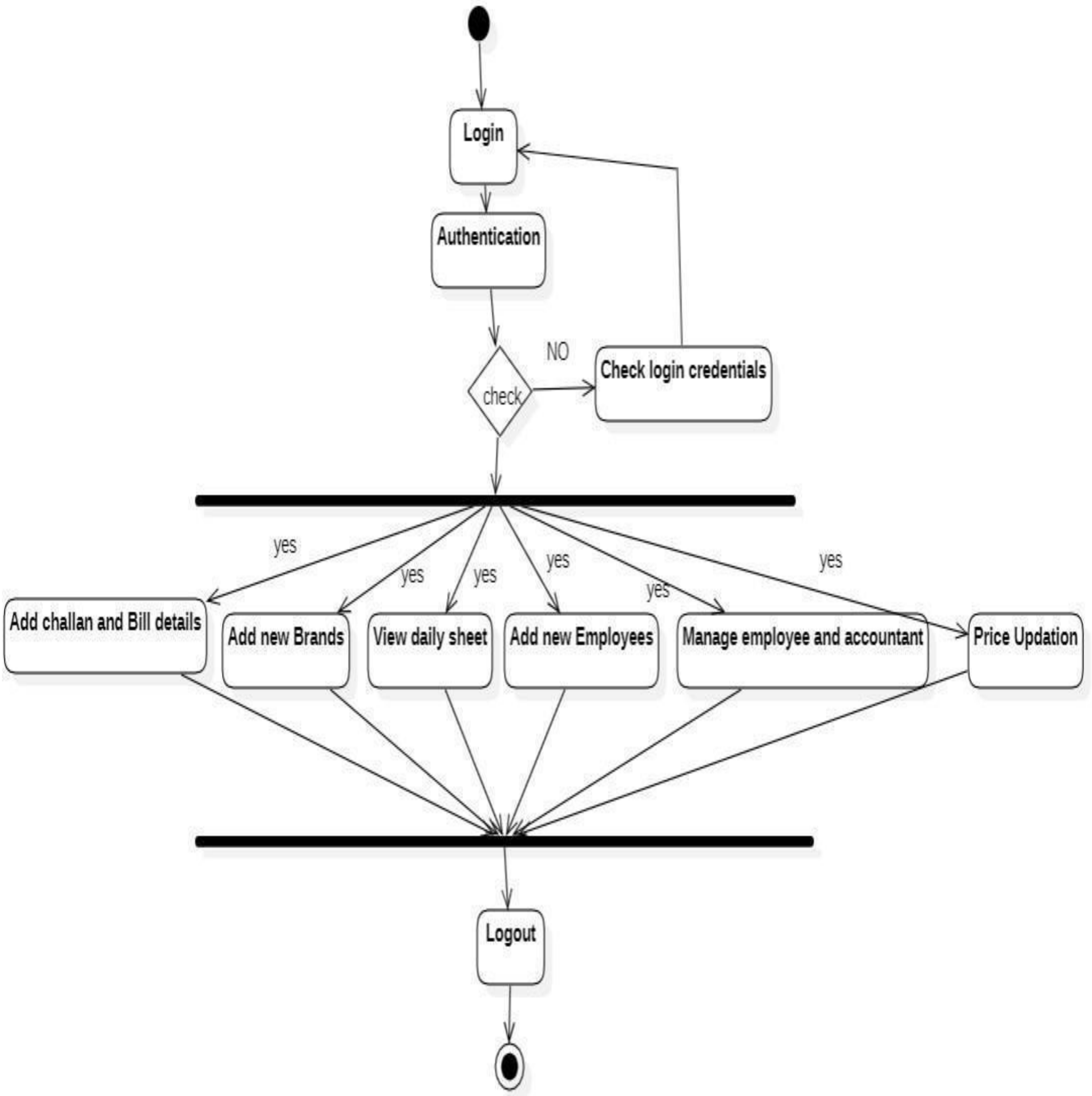


Figure 4.4.5.1 Activity diagram for Admin

The employee can perform activity actions as shown in Figure 4.4.5.2 which is the activity diagram for the employee.

Employee, after entering valid username and password, he enters the stock delivered on the day,daily transactional sheet and he can view the price list and new brands added.

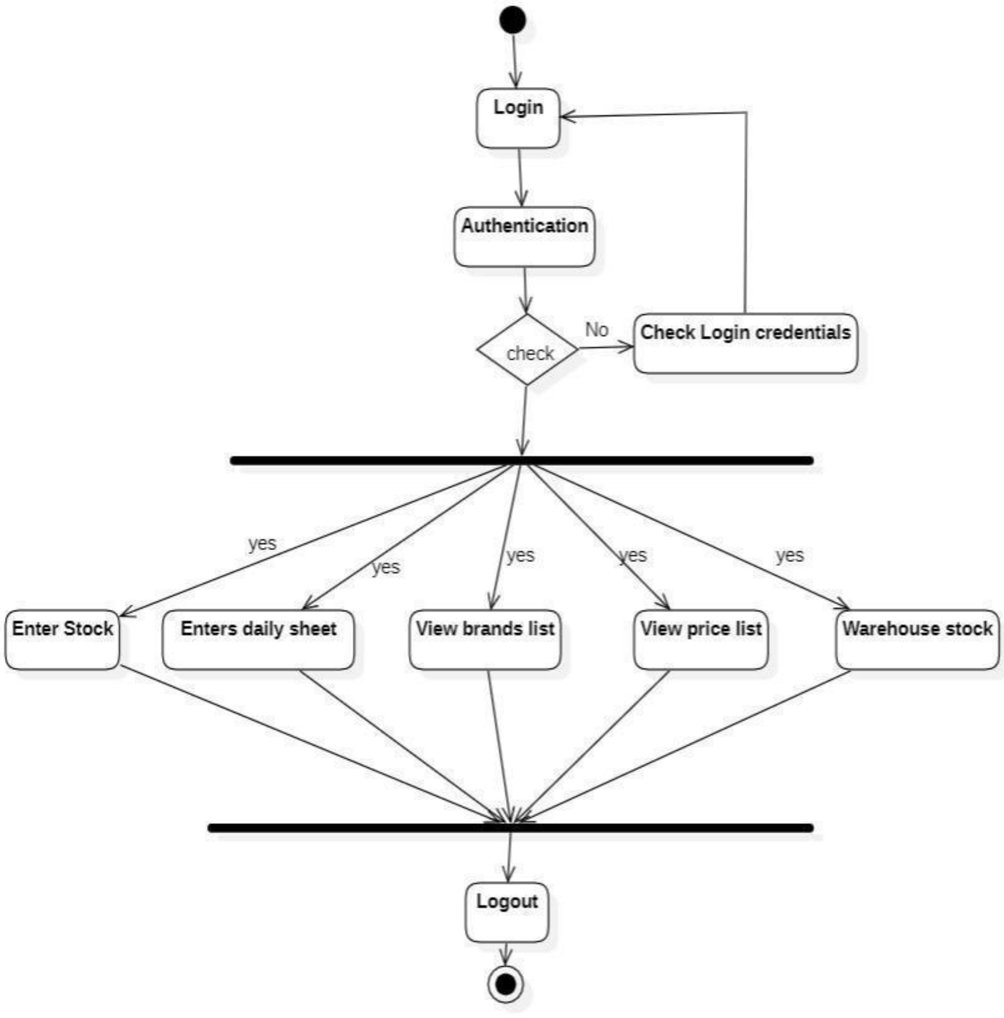


Figure 4.4.5.2 Activity diagram for Employee

The accountant can perform activity actions as shown in Figure 4.4.5.3 which is the activity diagram for the accountant.

Accountant, after entering a valid username and password, he/she can view the daily sheet and challan details.

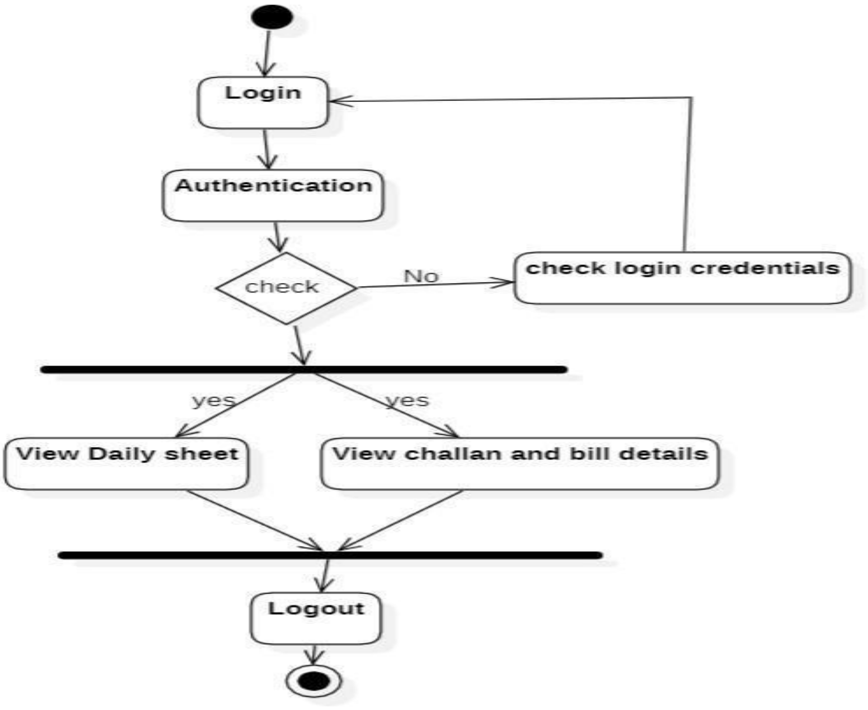
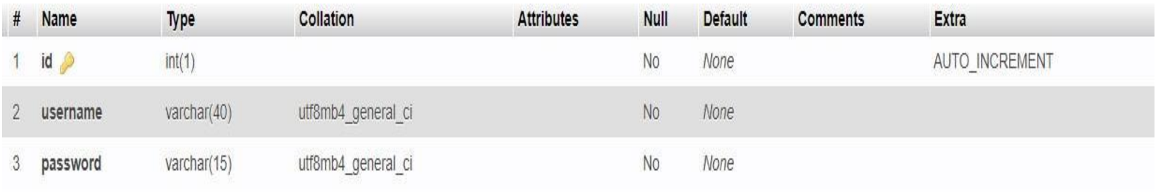


Figure 4.4.5.3 Activity diagram for accountant

4.5 DATA DICTIONARY:

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project. It describes the meanings and purposes of data elements within the context of a project, and provides guidance on interpretation, accepted meanings and representation. A Data Dictionary also provides metadata about data elements. The metadata included in a Data Dictionary can assist in defining the scope and characteristics of data elements, as well the rules for their usage and application.

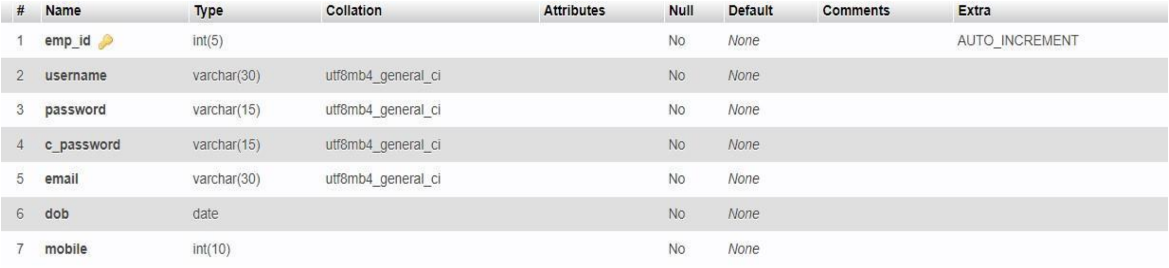
Admin :



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(1)			No	*None*		AUTO_INCREMENT
2	username	varchar(40)	utf8mb4_general_ci		No	*None*		
3	password	varchar(15)	utf8mb4_general_ci		No	*None*		

Table 4.5.1 Database for admin

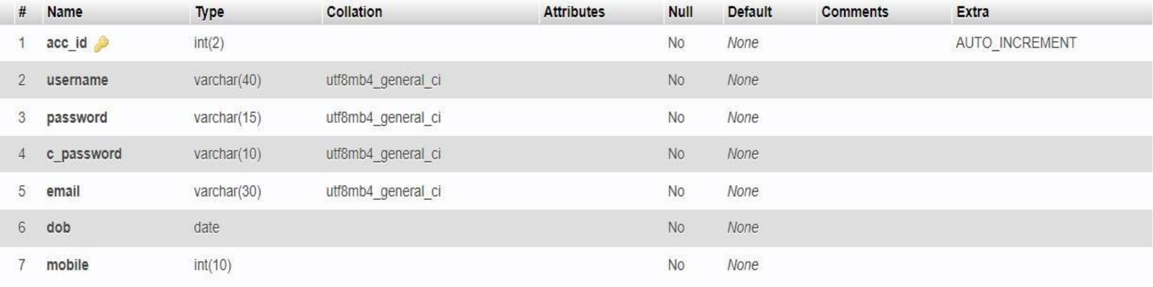
Employee :



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	emp_id	int(5)			No	*None*		AUTO_INCREMENT
2	username	varchar(30)	utf8mb4_general_ci		No	*None*		
3	password	varchar(15)	utf8mb4_general_ci		No	*None*		
4	c_password	varchar(15)	utf8mb4_general_ci		No	*None*		
5	email	varchar(30)	utf8mb4_general_ci		No	*None*		
6	dob	date			No	*None*		
7	mobile	int(10)			No	*None*		

Table 4.5.2 Database for employee

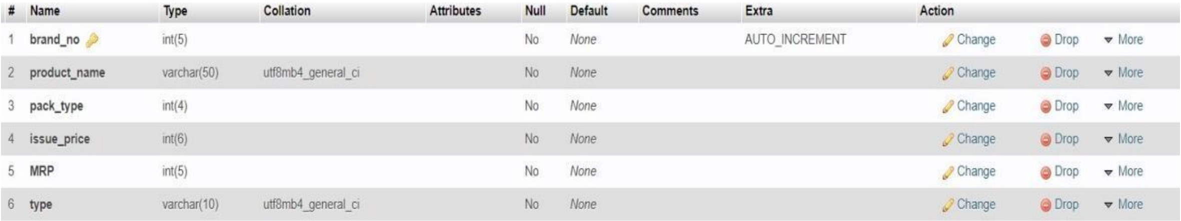
Accountant:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	acc_id	int(2)			No	*None*		AUTO_INCREMENT
2	username	varchar(40)	utf8mb4_general_ci		No	*None*		
3	password	varchar(15)	utf8mb4_general_ci		No	*None*		
4	c_password	varchar(10)	utf8mb4_general_ci		No	*None*		
5	email	varchar(30)	utf8mb4_general_ci		No	*None*		
6	dob	date			No	*None*		
7	mobile	int(10)			No	*None*		

Table 4.5.3 Database for accountant

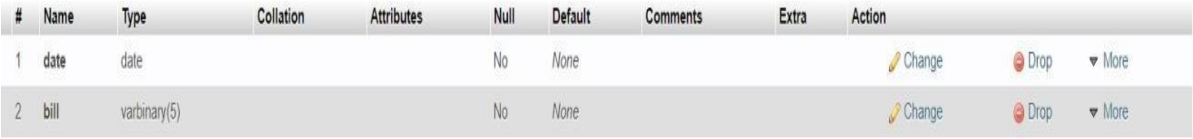
Brands :



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	brand_no	int(5)			No	*None*		AUTO_INCREMENT	Change	Drop	▾ More
2	product_name	varchar(50)	utf8mb4_general_ci		No	*None*			Change	Drop	▾ More
3	pack_type	int(4)			No	*None*			Change	Drop	▾ More
4	issue_price	int(6)			No	*None*			Change	Drop	▾ More
5	MRP	int(5)			No	*None*			Change	Drop	▾ More
6	type	varchar(10)	utf8mb4_general_ci		No	*None*			Change	Drop	▾ More

Table 4.5.4 Database for brand

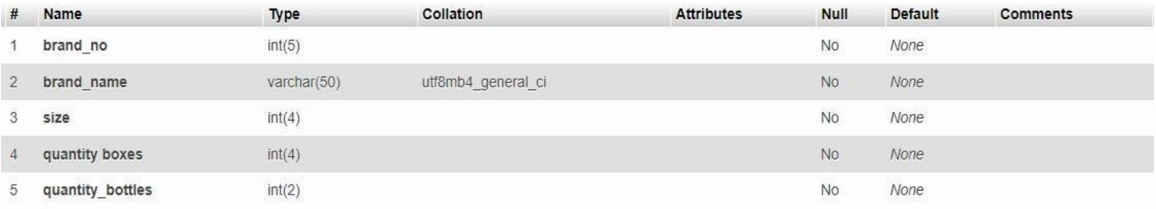
Bill:



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	date	date			No	*None*			Change	Drop	▾ More
2	bill	varbinary(5)			No	*None*			Change	Drop	▾ More

Table 4.5.5 Database for bill

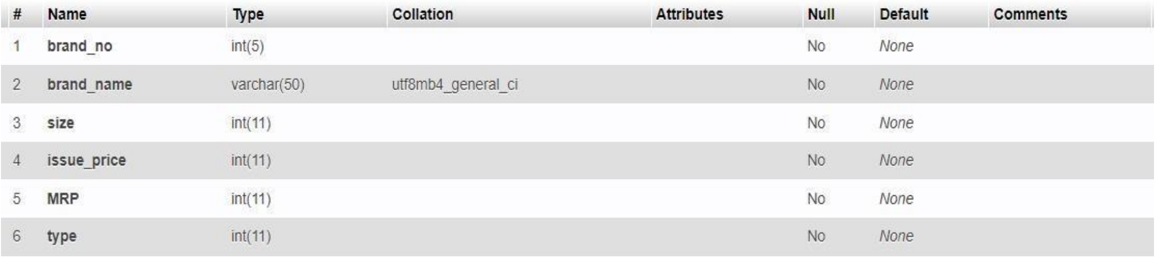
Inventory:



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	brand_no	int(5)			No	*None*	
2	brand_name	varchar(50)	utf8mb4_general_ci		No	*None*	
3	size	int(4)			No	*None*	
4	quantity boxes	int(4)			No	*None*	
5	quantity_bottles	int(2)			No	*None*	

Table 4.5.6 Database for inventory

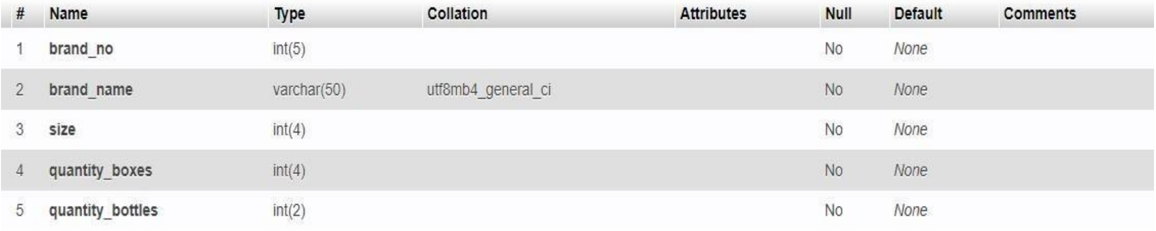
Price:



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	brand_no	int(5)			No	*None*	
2	brand_name	varchar(50)	utf8mb4_general_ci		No	*None*	
3	size	int(11)			No	*None*	
4	issue_price	int(11)			No	*None*	
5	MRP	int(11)			No	*None*	
6	type	int(11)			No	*None*	

Table 4.5.7 Database for price

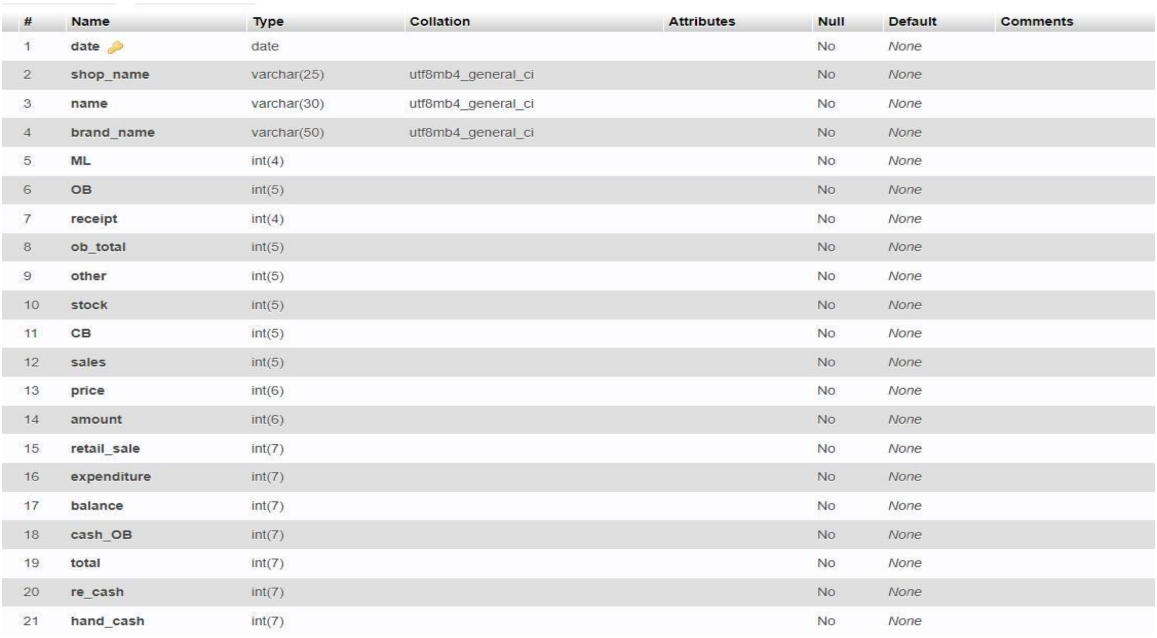
Warehouse:



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	brand_no	int(5)			No	*None*	
2	brand_name	varchar(50)	utf8mb4_general_ci		No	*None*	
3	size	int(4)			No	*None*	
4	quantity_boxes	int(4)			No	*None*	
5	quantity_bottles	int(2)			No	*None*	

Table 4.5.8 Database for Warehouse

Daily sheet:



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	date 🔑	date			No	*None*	
2	shop_name	varchar(25)	utf8mb4_general_ci		No	*None*	
3	name	varchar(30)	utf8mb4_general_ci		No	*None*	
4	brand_name	varchar(50)	utf8mb4_general_ci		No	*None*	
5	ML	int(4)			No	*None*	
6	OB	int(5)			No	*None*	
7	receipt	int(4)			No	*None*	
8	ob_total	int(5)			No	*None*	
9	other	int(5)			No	*None*	
10	stock	int(5)			No	*None*	
11	CB	int(5)			No	*None*	
12	sales	int(5)			No	*None*	
13	price	int(6)			No	*None*	
14	amount	int(6)			No	*None*	
15	retail_sale	int(7)			No	*None*	
16	expenditure	int(7)			No	*None*	
17	balance	int(7)			No	*None*	
18	cash_OB	int(7)			No	*None*	
19	total	int(7)			No	*None*	
20	re_cash	int(7)			No	*None*	
21	hand_cash	int(7)			No	*None*	

Table 4.5.9 Database for Daily sheet

CHAPTER 5

CONCLUSION

To conclude, Inventory management system is a simple desktop application suitable for Sri Sai Thirumala Wines. Firstly, we have understood all the requirements of the project and then we proposed a solution to the existing system. After analysis we have decided to have three modules in the system, they are admin, employee and accountant. In the application respective users can update, insert and delete the items as per the requirement. This application also provides a report on a daily basis to show the daily sales and stocks. At the end of Stage-1 we have designed the system model and its architecture. In Stage-2 the designs will be implemented.