



**COMPANY NAME: WINGS**

# **Database Design Document For Inventory Management System**

**DATE:01/11/2022**

Prepared by  
Group Number: 8

Gnana Prakash P	B200840CS	puligundla_b200840cs@nitc.ac.in
Deepak CH	B200802CS	chekurthideepak_b200802cs@nitc.ac.in
Ramnivas P	B200838CS	polineni_b200838cs@nitc.ac.in
Pramodh P	B200852CS	podety_b200852cs@nitc.ac.in
Sai Krishna T	B200816CS	tirukkovalluri_b200816cs@nitc.ac.in

# Database Design Document

## Inventory Management System

### Version<1.0>

---

## Table of Contents

---

<b>1 Purpose .....</b>	<b>4</b>
1.1 Document Objectives .....	4
1.2 Intended Audience .....	4
1.3 Acronyms and Abbreviations .....	5
1.4 Key Personnel .....	5
 <b>2 Assumptions, Constraints and Dependencies .....</b>	 <b>5</b>
2.1 Assumptions .....	5
2.2 Constraints .....	6
 <b>3 System Overview .....</b>	 <b>7</b>
3.1 Database Management System Configuration .....	7
3.2 Database Software Utilities .....	7
3.3 Support Software .....	7
 <b>4 Architecture .....</b>	 <b>8</b>
4.1 Hardware Architecture .....	8
4.2 Software Architecture.....	10
4.3 Datastores .....	11
 <b>5 Database-Wide Design Decisions .....</b>	 <b>11</b>
5.1 Interfaces .....	11
5.2 Key Factors Influencing Design .....	12
5.3 Behavior .....	13
5.4 DBMS Platform .....	13
5.5 Security and Availability .....	14
5.6 Distribution.....	14
5.7 Backup and Restore Operations .....	14
5.8 Maintenance .....	15
5.9 Performance and Availability Decisions .....	15

# Database Design Document

## Inventory Management System

### Version<1.0>

---

<b>6 Database Administrative Functions .....</b>	<b>15</b>
6.1 Database Identification .....	15
6.2 Schema Information .....	16
6.3 Schema Description .....	17
6.4 Physical Design .....	18
6.5 Entity Mapping .....	22
6.6 Relational Mapping.....	23
6.7 Mapping Rules .....	24
6.8 Operational Implications .....	25
6.9 Backup and Recovery .....	25
<b>7 Detailed Database Design .....</b>	<b>26</b>
7.1 Data Software Objects and Resultant Data Structures .....	26
<b>8 Appendix .....</b>	<b>31</b>

## **1 Purpose**

This Functional Database Design document provides detailed information of the IDM data model implemented to support the functional requirements for WINGS target database management system with consideration to the system's performance requirements.

The document describes, how the database that will support the [Application] Data Model with details of the logical and physical definitions. The document provides the functional and non-functional usage of the tables, considerations and requirements.

Further, the document would briefly describe the integration aspects of the Database with the Web Application . The Web Application would provide the users with easy access to IDM data.

### **1.1 Document Objectives**

The Database Design Document has the following objectives:

- To describe the design of a database, that is, a collection of related data stored in one or more computerized files that can be accessed by users or computer developers via a DBMS.
- To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

### **1.2 Intended Audience**

This document is intended for the following audiences:

- Technical reviewers, who must evaluate the quality of this document.
- Developers including:
  - i. Architects, whose overall architecture design must meet the requirements specified in this document.
  - ii. Designers, whose design must meet the requirements specified in this document.
  - iii. Developers, whose software must implement the requirements specified in this document.
  - iv. Quality Assurance personnel, whose test cases must validate the requirements specified in this document

# Database Design Document

## Inventory Management System

### Version<1.0>

---

## 1.3 Acronyms and Abbreviations

Acronym / Abbreviation	Meaning
RDBMS	Relational Database Management System
DBA	Database Administrator
1NF	First Normal Form
2NF	Second Normal Form
3NF	Third Normal Form
BCNF	Boyce Codd Normal Form
OS	Operating System

## 1.4 Key Personnel

Role	Name	Email	Phone No.
DBA	Gnana Prakash P	puligundla_b200840cs@nitc.ac.in	9573942689
DBA	Deepak CH	chekurthideepak_b200802cs@nitc.ac.in	9618006792
DBA	Ramnivas P	polineni_b200838cs@nitc.ac.in	7673909675
DBA	Pramodh P	podety_b200852cs@nitc.ac.in	9182017440
DBA	Sai Krishna T	tirukkovalluri_b200816cs@nitc.ac.in	9441296739

## 2 Assumptions, Constraints and Dependencies

### 2.1 Constraints

The current constraints on the project are related to the provision of database design.

- Many relational database systems impose limits on the lengths of data fields. If you enter more information into a field than it can accommodate, the information will be lost.
- Orders can only be placed when a product is in active status which is a constraint on updating a database. So order cannot be placed .

- Many relational database systems impose limits on the data types of data fields. If you enter wrong details than the actual , information will not be entered into the database.

## **2.2 Dependencies**

### **i. Logical dependencies**

Also known as causal dependencies. These dependencies are an inherent part of the project and cannot be avoided. Tasks characterized as logical dependency usually use the output of the preceding tasks as input so you can't run them in parallel.

### **ii. Resource dependencies**

This dependency originates from a project constraint as it deals with the availability of shared resources. If two tasks require the same resource for completion, then they'll be dependent on the completion of the other.

### **iii. Preferential dependencies**

These dependencies generally depend on the team members, other stakeholders, and industrial practices. Preferential dependencies arise when tasks are scheduled to follow developed standard practices.

### **iv. External dependencies**

No matter how much you plan, there are things bound to be out of your control. Some tasks are dependent on outside factors and project managers can't do anything to influence their [project](#) progress. To deal with these dependencies, it's recommended to have a backup plan.

Delays from the suppliers or other unforeseen circumstances may take place which can affect your progress. A good project manager always makes some contingency plans so everything keeps running smoothly even in the face of adversity.

## **3 System Overview**

---

This database was developed for web applications used by managers of the store to keep record of the stocks. An inventory management system is the combination of technology (hardware and software) and processes and procedures that oversee the monitoring and maintenance of stocked products, whether those products are company assets, raw materials and supplies, or finished products ready to be sent to vendors or end consumers.

### **3.1 Database Software Utilities**

The database management system used is my SQL as the name suggests SQL is used for creating and handling database and all required functions and before performing any related functions or queries Xampp server should be started.

### **3.2 Support Software**

The software directly related to the database we are using for storing our database is my SQL which basically uses SQL as a query language to perform all the database related queries whereas MySQL is written in C and C++. It's SQL parser is written in yacc, but it uses a home-brewed lexical analyzer.

MySQL is the database that stores all your information in an easily queried format. Apache acts as our local web server. It processes requests and serves web assets and content via HTTP. PHP is the programming language that works with apache to help create dynamic web content.

## **4 Architecture**

---

### **4.1 Hardware Architecture**

In hardware architecture we will mostly see how the data is stored in databases and how they are connected. It mostly comprises internal level. So, it deals with how the information is stored.

# Database Design Document

## Inventory Management System

### Version<1.0>

---

#### **1) Login Interface :**

Initially the websites shows a login interface where the workers and admins should login. The worker login details would be provided by the admin. At this interface by entering the login details authentication takes place and logs in when the details are correct.

#### **2) Stock In :**

Stock in provides the admin to add an company, add stock and even add clients. The stock in comes in handy with the suppliers. Where the admin can add even new items based on the requirements.

#### **3) Sell Stocks :**

Orders can be placed by the warehouses where the workers see through it and the admin needs to confirm the order. Once the admin confirms it with the help of the workers then the stock can be sent to the respective warehouses.

#### **4) Generate Reports :**

The reports would be generated on the basis of selling of stocks and buying from the suppliers which are viewed by the admin for making any further orders. These reports also help in forecasting the revenue and helps in taking decisions for the further growth of the company.

#### **5) Admin :**

Admin can add workers or remove them. They even confirm payments for the placed orders, and even place orders for raw materials. Will have the access to all the data present in the system.

#### **6) Workers :**

Workers login with the details provided by the admin and modify the details of items and ship them to warehouses. They take care of all the requirements and make requests for the admin.



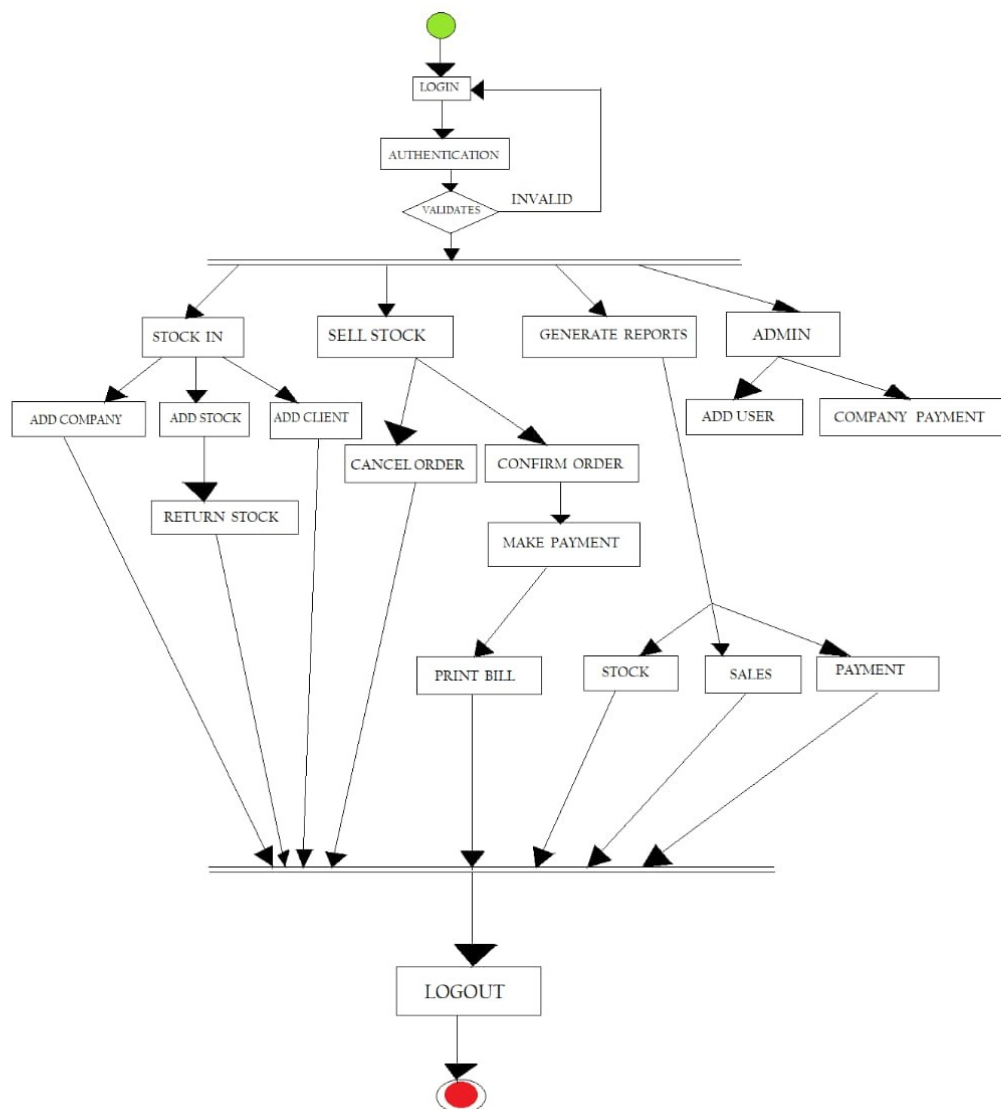
# Database Design Document

## Inventory Management System

### Version<1.0>

---

FLOW CHART:



## Database Design Document

### Inventory Management System

#### Version<1.0>

---

## 4.2 Software Architecture

Software architecture mostly deals with external level. Software architecture deals with interfaces. In Inventory Management System, we have Two interfaces and now let us see how they work.

### **INTERFACES:**

1. Admin Interface.
2. Login Interface.

#### **1.Admin Interface:**

In the login interface, the admin can enter his email and password and it will redirect him to the admin page where he can manage the employees and confirms issuing the payments.

#### **2.Login Interface:**

In the login interface the manager can enter their email and password and they can login to the portal and after login they have many options like Selling a product, Ordering a product, updating the status of product and issuing the transcripts . In this Login interface we have many options like home, if we click on home, we will be redirected to Home page.

### **Admin Interface**

- **Dashboard :**

Dashboard contains all the details of the items, number of products shipped, raw materials used, number of items present, number of workers working and other required details of the workers.

- **Employee :**

Admin can add an employee or remove employee in case of adding employee the admin generates the login details and sends to the employees.

- **Items :**

This shows the details of the items, like item id, price of the item and selling prices.

## Database Design Document

### Inventory Management System

#### Version<1.0>

---

- **Stocks :**  
The quantity of item present, and shows the requirement of the raw materials based on the current present stock.
- **Transcations :**  
Transcations contains both seller's and employees order transaction details. Seller's transactions show the amount spent on the raw materials bought and the employees' orders show the transactions done with the warehouses.
- **Supplier :**  
Contains the details of the suppliers who supplied the raw materials till now.
- **Reports :**  
Contains the graphs and pie charts on the amount of revenue generated and the number of supplied products. This shows a brief overview on the revenue spent and the revenue generated.
- **Worker Interface**
- **Stocks :**  
Worker can see the amount of stock present and request for the required raw materials, and even ship the finished products to the warehouses.
- **Orders :**  
Worker takes orders from the warehouses and modifies the amount of stocks present.

### 4.3 Datastores

Data is stored in the Inventory management database and in the tables mentioned above. For file management, we are storing the content inhtdocs in the xampp folder. Php files are used for backend and html and CSS files are used for front end.

## **5. Database-Wide Design Decisions**

### **5.1 Interfaces**

There are 2 types of interfaces, they are,

1. Admin Interface
2. Worker Interface

Initially we would have a login page where workers login to the site through worker's window and the admin logs in through admin window.

The admin creates the login details for the workers and even remove workers. Admin interface also has dashboard, employees, items, stocks, transactions, suppliers and reports. Where the worker only deals with stocks and orders.

### **5.2 Key Factors Influencing Design**

Key factors influencing the design are the functional and nonfunctional requirements.

- **Functional Requirements**

- F1.** Admin logs in to the website.
- F2.** Admin creates account for workers and shares details of it.
- F3.** Workers can login,logout of the site.
- F4.** Workers can change the availability of a product as available and not available.
- F5.** Workers have to update brands as total brands registered vs active brands.
- F6.** Workers have to order and verify the stock.
- F7.** If the stock received is expired Workers have to return the product.
- F8.** Workers can confirm and cancel the orders from customers.
- F9.** Workers have to generate reports of sales,stock and payments.
- F10.** Admin approves companies payment for stock purchase.
- F11.** Workers have to keep track of supplier and payments from supplier.
- F12.** Admin analyse and keep track of investment vs turnover.

# Database Design Document

## Inventory Management System

### Version<1.0>

---

- **Nonfunctional Requirements**

1. **Flexibility**

The operation may be flexible and reports can be presented in many ways.

2. **Maintainability**

After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

3. **Portability**

The software can be deployed at any machine.

4. **Reusability**

The data and record that are saved in the database can be reused if needed.

5. **Robustness**

If there is any error in any window or module then it does not affect the remaining part of the software.

6. **Timelines**

The time limit is very important. It will save much time and provide fast accessing.

7. **Security**

Data is backed up regularly, the database is protected and users cannot edit the database. For security reasons, only administrators are allowed to edit the database, and administrators can only access the database.

8. **Performance**

Performance should be accurate and response time should be kept to a minimum so that there are no issues with processing the request. Data should be backed up regularly as a log file to prevent data loss if the data is completely lost due to server crash or file corruption. To be more effective, the DBMS needs to be able to process large amounts of data and take action in less time.

## **5.3 Behavior**

### **1. Login**

To enter into the website either the worker or the admin must login into the website, any ways the admin has his details and the worker's login details are shared by the admin when the approval is accepted.

### **2. Admin**

Admin can see the dashboard, add or remove employees and see their details, can see the details of the items like the selling price, quantity present, prices of the item. Admin must finalize the approval for the raw materials order and the shipping of products to the warehouses. Reports are even produced by monetering the products supply and the revenue generated. Admin can even contact the supplier if required.

### **3. Worker**

Worker just login to the site with the details provided by the admin and monitor the items, by shipping the finished products after the acceptance of the admin and requests the order of raw materials for the further requirements. All the work done would be seen by the admin.

## **5.4 DBMS Platform**

This DBMS software can be implemented with ease on Linux based operating systems, windows and mac OS. An open-source cross platform web-server should be installed to create a database (example- xampp) Which can have baseline requirements to meet to install.

Baseline requirements for the system:

- Windows 2008 server or later
- Mac os X 10.6
- CentOS,Ubuntu,Fedora,Gentoo,Arch,SUSE
- Hard disk space : 40GB, RAM :256MB
- Processor: Pentium(R)Dual-core CPU

## **5.5 Security and Availability**

This software will,

- Authenticate each user based on the type of user which are admin and manager.
- When a user performs an action which is not authorized by the user, the system will display an error message if it's found to be unauthorized.

## **5.6 Distribution**

The master database is the SQL Server primary configuration database. It contains information about all databases on the server, including physical database files and their location. The master database file also contains SQL Server configuration settings and login account information.

Components in the master database

- Registrations and Remote Logins
- Local Databases and Database Files
- Login Accounts
- Server Configuration Settings
- Processes and Locks

A current backup will be kept updating it from time to time as it is critical to any server recovery. Integrity standards are high and the privacy is kept for the user, no unauthorized information will not be disclosed. No business rules are included.

## **5.7 Backup and Restore Operations**

The user information will be kept private for safety and security issues and will not be disclosed to any other third-party organizations so that user privacy is intact and information is safe.

The data is backed up on a regular basis so the data will not be lost if database crashes or any other harm which leads to loss of data. Also, as a safety measure, the data is stored on a private storage so it can't be accessed from outside. For detailed information on actions in backup please refer to the 6.16 module below.

## **5.8 Maintenance**

Maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error message generation.

## **5.9 Performance and Availability Decisions**

The developed DBMS software needs to be able to output information effectively when needed and save the data without waiting time to avoid problems. Several factors that affect performance are the need for adequate system resources and the basic requirements for the software to run successfully.

Software availability is sufficient for users and some features are limited to users due to the risk of data loss and data protection issues. The database can only be edited by an administrator to change the data as needed.

# **6 Database Administrative Functions**

## **6.1 Database Identification**

Element	Element Name	Description
db_name	Inventory	Inventory is the name of the database when it is originally created.
db_path	C:\xampp\mysql\data\ inventory	The full path to where the database is stored on the system.
db_location	C:\xampp\mysql\data\ library	Location where the database is stored



## **6.2 Schema Information**

In Entity Relation Schema,

Entities:

1. Admin
2. Item
3. Order
4. Supplier
5. Sales
6. Stock
7. Worker

Relations:

1. Supplies
2. Adds
3. Sales\_List
4. Order\_List
5. Present in
6. Updates
7. Sells
8. Places
9. Updates

Attributes:

1. Admin
  - User\_name
  - Admin\_name
  - Password

2.Item

- Item\_ID
- Item\_name
- Brand\_name
- Category
- Purchase\_Price
- Selling\_Price

3. Order

- Order\_id
- Quantity

4. Supplier

- Supplier\_ID
- Email
- Supplier Name

5. Sales

- Sales\_ID
- Quantity

6. Stock

- Stock\_Id
- Quantity

7. Worker

- Worker\_Name
- User\_name
- Password
- E-mail

## **6.3 Schema Description**

Here in the items relation item\_id is the primary key and supplier\_id is foreign key.

In the supplier relation supplier\_id is the primary key.

In the orderlist relation item\_id and order\_id are acting as primary keys at the same time.

In the order relation order\_id is primary key and stock\_id is the foreign key.

In the stock relation stock\_id is the primary key and sales\_id is the foreign key.

In the sales relation sales\_id is the primary key and worker username is foreign key.

In sales list both item\_id and sales\_id are foreign keys acting as primary keys at the same time.

In the worker relation user\_name is the primary key.

# Database Design Document

## Inventory Management System

### Version<1.0>

---

#### Cardinality:

- i. Item-Admin ---- N:1
- ii. Item-Order ----- M:N
- iii. Item-Stock ----- N:1
- iv. Item-Sales----- M:N
- v. Item-Supplier-- N:1
- vi. Admin-Order-- 1:N
- vii. Order-Stock--- N:1
- viii. Stock-Sales--- N:1
- ix. Sales-Worker- N:1

## 6.4 Physical Design

### Normalization of database

- 1) It is a technique used to remove redundancy in the database and to maintain consistency and integrity.
- 2) It is used to remove certain anomalies in the database in order to maintain consistency.

The anomalies are:

1. Insertion Anomalies
2. Deletion Anomalies
3. Update Anomalies

**All the tables of our database are:**

#### **1.ITEM:**

Item_name	<u>Item_ID</u>	purchase_ID	Selling_price	Brand_name	Category	Supplier_ID	Stock_ID

#### **2.SUPPLIER:**

<u>Supplier_ID</u>	Supplier_Name	E-mail

# Database Design Document

## Inventory Management System

### Version<1.0>

---

#### 3.Order\_List:

<u>Item_ID</u>	<u>Order_ID</u>

#### 4.Order:

<u>Order_ID</u>	Quantity	<u>Stock_ID</u>

#### 5.Stock

<u>Stock_ID</u>	Quantity	<u>Sales_ID</u>

#### 6.Sales

<u>Sales_ID</u>	Quantity	User_Name

#### 7.Sales\_List

<u>Item_ID</u>	<u>Sales_ID</u>

#### 8.Worker

<u>User_name</u>	Worker_Name	Password	Email

**Database Design Document**  
**Inventory Management System**  
**Version<1.0>**

---

**Checking 1st normal form:**

As we can see all our tables are already in normal form as in each table there are no redundancies so the table is in 1NF.

**Checking 2nd normal form:**

As we can see in every Table there is no composite key so there is no transitive.

Dependencies and in other tables data completely depend on the two key attributes (i.e. Every attribute depends on Primary Key) so no partial dependencies. So table is in 2NF.

**Checking 3rd normal form:**

And in tables *Item, Supplier, Order, Stock, Sales and Admin* there is no transitive dependency exists for the attributes that are non-prime. So tables are in 3NF.

**Checking Boyce Codd normal form:**

Since all the super keys in each of the tables determine tuples uniquely all tables are in BCNF.

•All the tables above are in normal form already and the tables are normalized up to the Boyce Codd normal form."

# Database Design Document

## Inventory Management System

### Version<1.0>

---

## 6.5 Entity Mapping

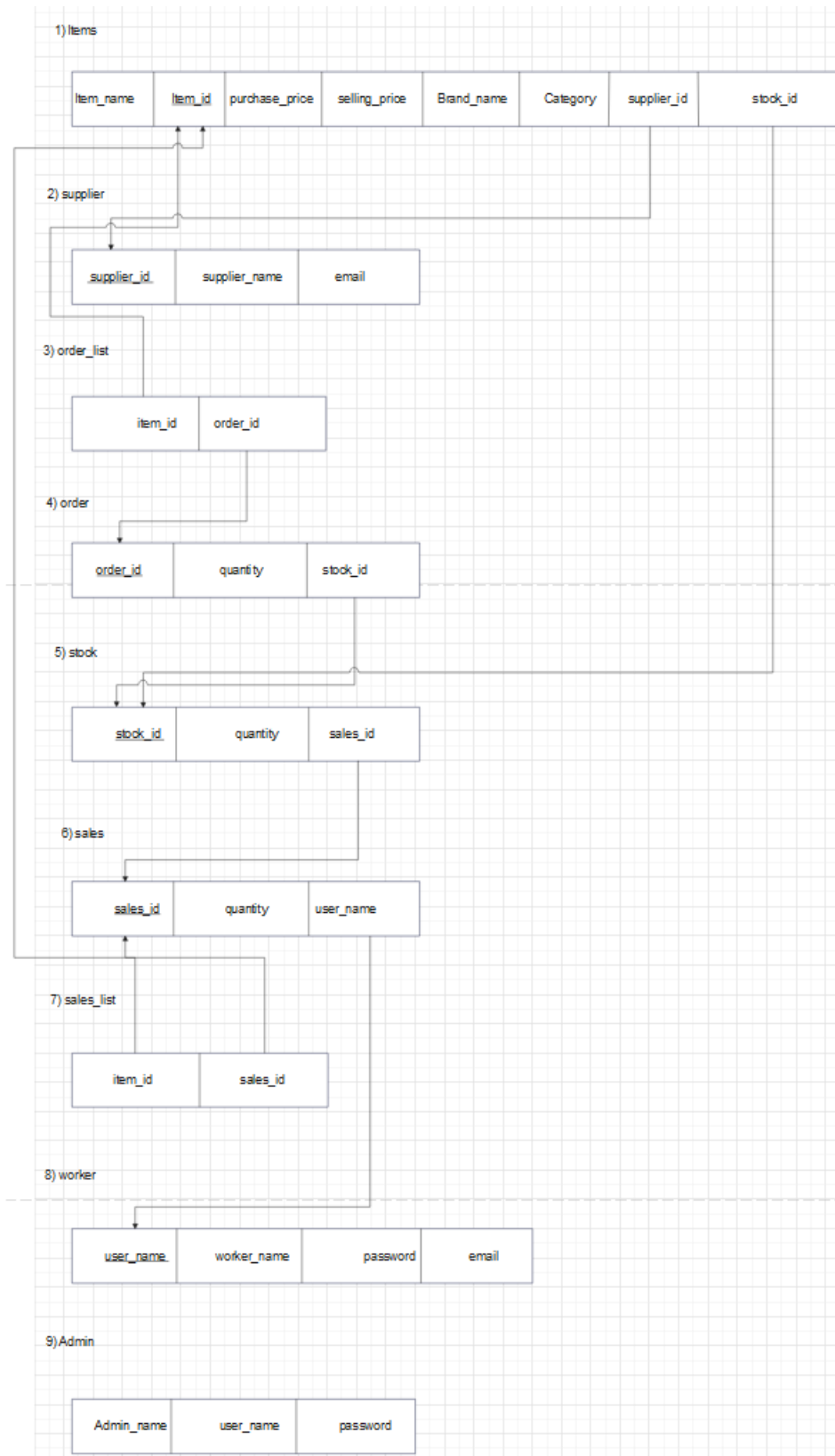


# Database Design Document

## Inventory Management System

### Version<1.0>

## 6.6 Relational Mapping



## **6.7 Mapping Rules**

To complete the mapping from an ERD to relations we must consider the entity types, relationship types, and attributes that are specified for the model.

### **1. Mapping of Stronger-Entity types:**

Strong, or regular, entity types are mapped to their own relation. The PK is chosen from the set of keys available.

### **2. Mapping of Weaker-Entity types:**

Weak entity types are mapped to their own relation, but the primary key of the relation is formed as follows. If there are any identifying relationships, then the PK of the weak entity is the combination of the PKs of entities related through identifying relationships and the discriminator of the weak entity type; otherwise the PK of the relation is the PK of the weak entity.

### **3. Mapping of Binary 1:1 Relationship Types:**

In general, with a one-to-one relationship, a designer has a choice regarding where to implement the relationship. One may choose to place a foreign key in one of the two relations, or in both. Consider placing the foreign key such that nulls are minimized. If there are attributes on the relationship those can be placed in either relation.

### **4. Mapping of Binary 1:N Relationship Types:**

With a one-to-many relationship the designer must place a foreign key in the relation corresponding to the 'many' side of the relationship. Any other attributes defined for the relationship are also included on the 'many' side.

### **5. Mapping of Binary M:N Relationship Types:**

A many-to-many relationship must be implemented with a separate relation for the relationship. This new relation will have a composite primary key comprising the primary keys of the participating entity types and any discriminator attribute, plus other attributes of the relationship if any.

### **6. Mapping of Multivalued Attributes:**

Each multi-valued attribute is implemented using a new relation. This relation will include the primary key of the original entity type. The primary key of the new relation will be the primary key of the original entity type plus the multi-valued attribute. Note that in this new relation, the attribute is no longer multi-valued.



## **6.8 Operational Implications**

When the page is refreshed, it will redirect to the same web page.

### **Update:**

1. As there is no separate link provided for updating workers's data there is no option for updating from the worker side.
2. An admin can update the details of items the changes will be reflected in the database.
3. An admin adds new items and suppliers, updating the items and suppliers.
4. After updating by admin page returns to page with no entries of item.

### **Data transfer:**

1. admin login himself by entering appropriate details.
2. After logging in when he clicks add user in accounts option he will be able to add worker account details and worker will get an email of username and password of him .
3. workers will be enter appropriate details in the login page.
4. After logging in, the worker will be redirected to the student page.
5. Where he can place order for warehouses, where he can select the item and quantity of the item and confirms payment and will be redirected and stock will be updated in database.

## **6.9 Backup and Recovery**

For backup in case server fails or pc is damaged a backup shall be created by following steps:

1. Go to the root directory where the xampp folder is located.
2. Go to the mysql folder in the xampp folder.
3. Go to bin folder
4. Go to sql.
5. Copy the path
6. Open the terminal and go to the directory where xampp is located.
7. Paste the path.
8. Type the command "mysqldump -u root -p portal>backup.sql"

9. Execute the command
10. A backup will be created.
11. Save the backup file.
12. For recovery start xampp server open mysql\php my admin
13. Go to export select the backup .sql file and click export

## **7 Detailed Database Design**

### **7.1 Data Software Objects and Resultant Data Structures**

In detailed database design, the data type of the attributes, their lengths and primary keys and foreign keys are described.

#### **For Item Page:**

we need to store

item\_name,item\_id,purchase\_price,Category,Brand\_name,Selling\_price,Supplier\_ID

item\_id:

- Primary Key
- Data Type: varchar
- Length: 6

item\_name:

- Data Type: varchar
- Length: 25

Purchase\_Price:

- Data Type: int

Category:

- Data Type: varchar
- Length: 15

# Database Design Document

## Inventory Management System

### Version<1.0>

---

Brand\_name:

- Data Type: varchar
- Length: 12

Selling\_Price:

- Data Type: int

Supplier\_ID-FK

Item\_ID-PK

Stock\_ID-FK

#### For Supplier Page:

we need to store

Supplier\_ID,Supplier\_Name,Email

Supplier\_ID:

- Primary Key
- Data Type: varchar
- Length: 6

Supplier\_Name:

- Data Type: varchar
- Length: 25

Email:

- Data Type: varchar
- Length: 25

Supplier\_ID-PK

# Database Design Document

## Inventory Management System

### Version<1.0>

---

#### For Order Page:

Order\_ID:

- Primary Key
- Data Type: varchar
- Length: 6

Quantity:

- Data Type: int

Stock\_ID-FK

Order\_ID-PK

#### For Order\_List Page:

Order\_ID-FK,PK

Item\_ID-FK,PK

#### For Stock Page:

Stock\_ID:

- Primary Key
- Data Type: varchar
- Length : 6

Quantity:

- Data Type: int

Stock\_ID-PK

Sales\_ID-FK

# Database Design Document

## Inventory Management System

### Version<1.0>

---

#### For Sales Page:

Sales\_ID:

- Primary Key
- Data Type: varchar
- Length : 6

Quantity:

- Data Type: int

User\_Name -FK

Sales\_ID - PK

#### For Sales\_List Page:

Sales\_ID-FK,PK

Item\_ID-FK,PK

#### For Worker Page:

User Name:

- Data Type: varchar
- Length : 15

Worker\_Name:

- Data Type: varchar
- Length : 15

Password:

- Data Type: varchar
- Length : 15

# Database Design Document

## Inventory Management System

### Version<1.0>

---

Email:

- Data Type: varchar
- Length: 25

User\_Name -PK

**For Admin Page:**

User Name:

- Data Type: varchar
- Length : 15

Admin\_Name:

- Data Type: varchar
- Length : 15

Password:

- Data Type: varchar
- Length : 15

User\_Name -PK

## **8 Appendix**

---

**work done by:**

Gnana Prakash P	B200840CS	puligundla_b200840cs@nitc.ac.in
Deepak CH	B200802CS	chekurthideepak_b200802cs@nitc.ac.in
Ramnivas P	B200838CS	polineni_b200838cs@nitc.ac.in
Pramodh P	B200852CS	podety_b200852cs@nitc.ac.in
Sai Krishna T	B200816CS	tirukkovalluri_b200816cs@nitc.ac.in