Auto-Spare Parts Management System

A MINI-PROJECT REPORT

Submitted by

RA2011003010910 - ANUJ

RA2011003010925 - BASIM AHAMED

RA2011003010936 - RHYTHM PAHWA

Under the guidance of

Mrs. Viji D

(Assistant Professor, Department of Computing Technologies)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled “Auto-Spare Parts Management System" is the bonfire work of [“Anuj-RA2011003010910”, “Basim Ahamed-RA2011003010925”, “Rhythm Pahwa-RA2011003010936”] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Department of Computing Technology
Kattankulathur
603 203
SRM IST

Signature

Mrs VIJI D.
Assistant Professor
Department Of Computing Technologies

M.Pushpalatha

Signature

Dr Pushpalatha M
Head Of The Department
Dept. Of Computing Technologies

ABSTRACT

The Auto-Spare Part Management is developed for sale/purchase organization. This system manages customer, product, sales order, and dealer and purchase order. It provides a platform to the trader to automate their record keeping.

The purpose of this project is to develop an application program to reduce the manual work for managing inventory, product, customer, dealer and orders.

This project is built to help admin in managing his organization. Admin can store customer details in the database. Next time when customer comes in then there is no need to enter his details again.

It can selected from the list displaying all customer. The traditional approach to be replaced by centralized monitoring database by maintaining records.

TABLE OF CONTENT

1. INTRODUCTION

1.1 Introduction

The Auto-Spare Part Management is developed for sale/purchase organization. This system manages customer, product, sales order, and dealerand purchase order. It provides a platform to the trader to automate their record keeping. The purpose of this project is to develop an application program to reduce the manual work for managing inventory, product, customer, dealer and orders. This project is built to help admin in managing his organization. Admin can store customer details in the database. Next timewhen customer comes in then there is no need to enter his details again. It canselected from the list displaying all customer

Functions of Auto-Spare Part Management:

- Admin can store all customer and dealer record.
- This application keeps tracks of all the products being sold inthe organization.
- This system provides facility of making sales and purchase order.
- All products, customer and dealer record is maintained as master data. User can selected these from display list while making sales order and purchase order.

1.2 **Problem Statement**

The main reason to build this project is to provide the seller a in-look of the shop.

This project wills the guide owner to the products available in the shop and which are required to order from the sellers.

By this, shopkeeper doesn’t have to keep track of the records manually.

The more comfortable users of computerized systems were mostly users with a fair accounting knowledge. In some cases business owners left the entire system to the accountant who was more familiar with the system, a situation some said possess a business risk.

Deductions from the requirements analysis show that ease of use and knowledge prerequisite are the main reasons why most SMEs are not able to adopt computerized systems fully for tracking and managing their sales processes.

This situation makes them stick to old and sometimes wasteful methods of tracking sales.

The effect of this is loss of profits due to poor record keeping.

Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

1.3 SCOPE

The main reason to build this project is to provide the seller a in-look of the shop. This project wills the guide owner to the products available inthe shop and which are required to order from the sellers. By this, shopkeeper doesn't have to keep track of the records manually. The more comfortable users of computerized systems were mostly users with a fair accounting knowledge. In some cases business owners left the entiresystem to the accountant who was more familiar with the system, asituation some said possess a business risk. Deductions from the requirements analysis show that ease of use and knowledge prerequisite are the main reasons why most SMEs are not able to adopt computerized systems fully for tracking and managing their sales processes. This situation makes them stick to old and sometimes wasteful methods of tracking sales. The effect of this is loss of profits due to poor record keeping. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it.

APPLICATIONS

My proposed system will be designed to simplify the management of auto-spare parts by automating key processes such as tracking inventory levels, ordering parts, and managing supplier relationships. With this system, you will be able to easily access real-time inventory data, generate purchase orders and invoices, and manage suppliers, all from a single platform.

Key features of the system will include:

- Inventory Management: The system will enable you to track inventory levels, set reorder points, and generate alerts when parts are running low. This will help you to minimize stock-outs and ensure that you always have the parts you need on hand.
- Order Management: The system will allow you to generate purchase orders and invoices, track delivery times, and receive notifications when orders are received. This will enable you to manage your orders more efficiently and reduce the risk of errors or delays.
- Supplier Management: The system will enable you to manage your supplier relationships by tracking their performance, negotiating contracts, and monitoring delivery times. This will help you to build stronger relationships with your suppliers and ensure that you are getting the best possible deals.

Overall, my proposed auto-spare parts management system will provide your company with the tools it needs to manage its spare parts inventory more efficiently and effectively.

List of General and Unique Services in the database

- Here are some general and unique services that can be included in a database application for autospare management.
- Inventory management: This service would allow users to track the availability of auto parts in their inventory, monitor stock levels, and receive alerts when stock levels fall below a certain threshold.
- Purchase order management: This service would allow users to create and track purchase orders for auto parts, receive notifications when orders are fulfilled, and manage supplier information.
- Sales order management: This service would allow users to manage customer orders for auto parts, create and send invoices, and track the status of orders.
- Warranty and return management: This service would allow users to manage warranties and returns for auto parts, track the status of warranty claims, and manage customer returns.
- Pricing and discount management: This service would allow users to manage pricing for auto parts, set up discounts and promotions, and track sales performance.
- Customer relationship management (CRM): This service would allow users to manage customer information, track customer interactions, and manage customer feedback and complaints.
- Service and repair management: This service would allow users to manage service and repair orders for auto parts, track the status of orders, and manage technician schedules.
- Analytics and reporting: This service would allow users to generate reports on sales performance, inventory levels, customer behavior, and other key metrics, and use data to inform business decisions.
- Barcode scanning: This service would allow users to scan barcodes on auto parts to quickly retrieve information about the part, such as its price, availability, and location in the inventory. Integration with suppliers and partners: This service would allow users to integrate their database application with suppliers and partners, enabling them to receive real-time updates on inventory levels, pricing, and other key information.

SYSTEM REQUIREMENTS

Hardware Requirement:

- Core to Dual or above
- 2 GB RAM
- 25 GB Free Hard disk space
- Network interface card or Modem
- LAN Network

Software Requirements:

- Windows XP/7/8/10
- Linux
- DBMS
- NetBeans IDE 8.0.2
- SQL databases

2. LITERATURE SURVEY

2.1 Existing system

Literature survey on existing systems of auto-spare parts management reveals that there are several systems available in the market with varying features and functionalities. Some of the widely used systems are:

- SAP ERP: SAP ERP is a comprehensive business management system that includes modules for managing inventory, procurement, and logistics. It provides real-time visibility into inventory levels and helps automate the process of ordering and managing auto-spare parts.
- Oracle E-Business Suite: Oracle E-Business Suite is an integrated business management system that includes modules for managing inventory, procurement, and supply chain. It helps organizations to optimize their inventory levels and streamline their supply chain processes.
- Infor EAM: Infor EAM is an enterprise asset management system that helps organizations to manage their assets, including auto-spare parts. It provides real-time visibility into asset performance and enables organizations to optimize their maintenance processes.
- IBM Maximo: IBM Maximo is an asset management system that helps organizations to manage their assets, including auto-spare parts. It provides real-time visibility into asset performance, helps to optimize maintenance processes, and enables organizations to track inventory levels and order parts.
- Wasp Barcode Inventory Control: Wasp Barcode Inventory Control is a cloud-based inventory management system that helps organizations to manage their inventory, including auto-spare parts. It provides real-time visibility into inventory levels, enables organizations to track inventory movements, and helps to automate the process of ordering and managing parts.

Overall, existing auto-spare parts management systems offer a range of features and functionalities that can help organizations to streamline their inventory management processes and optimize their supply chain. However, the selection of a suitable system should be based on the specific needs and requirements of the organization.

2.2 Comparison of Existing vs Proposed system

Existing auto-spare parts management systems and the proposed system have many similarities, as they are designed to streamline inventory management processes and optimize supply chain operations. However, there are also some key differences between the existing and proposed systems.

Here is a comparison of some of the key features of existing vs proposed system of auto-spare parts management:

- Inventory Management: Both existing and proposed systems offer inventory management features such as tracking inventory levels, setting reorder points, and generating alerts when parts are running low. However, the proposed system goes further by providing real-time inventory data, which enables organizations to make more informed decisions about inventory management.
- Order Management: Existing systems and the proposed system both offer order management features such as generating purchase orders and invoices, tracking delivery times, and receiving notifications when orders are received. However, the proposed system provides greater flexibility by allowing for customization of order and invoice formats, which can help to improve efficiency and reduce errors.
- Supplier Management: Existing systems and the proposed system both offer supplier management features such as tracking supplier performance, negotiating contracts, and monitoring delivery times. However, the proposed system goes further by providing detailed analytics on supplier performance, enabling organizations to make data-driven decisions about supplier management.
- Integration: Existing systems and the proposed system both offer integration with other business systems such as accounting and finance. However, the proposed system is designed to be highly customizable and can be tailored to the specific needs and requirements of the organization.

- Accessibility: Existing systems and the proposed system can be accessed via desktop or mobile devices. However, the proposed system is designed to be highly intuitive and easy to use, enabling organizations to quickly adopt the system and start reaping the benefits of streamlined inventory management.

In conclusion, while there are similarities between existing auto-spare parts management systems and the proposed system, the proposed system offers greater flexibility, customization, and data-driven decision-making capabilities. This can help organizations to optimize their inventory management processes, reduce costs, and improve overall supply chain operations.

3.SYSTEM ARCHITECTURE AND DESIGN

3.1 ARCHITECTURE DIAGRAM

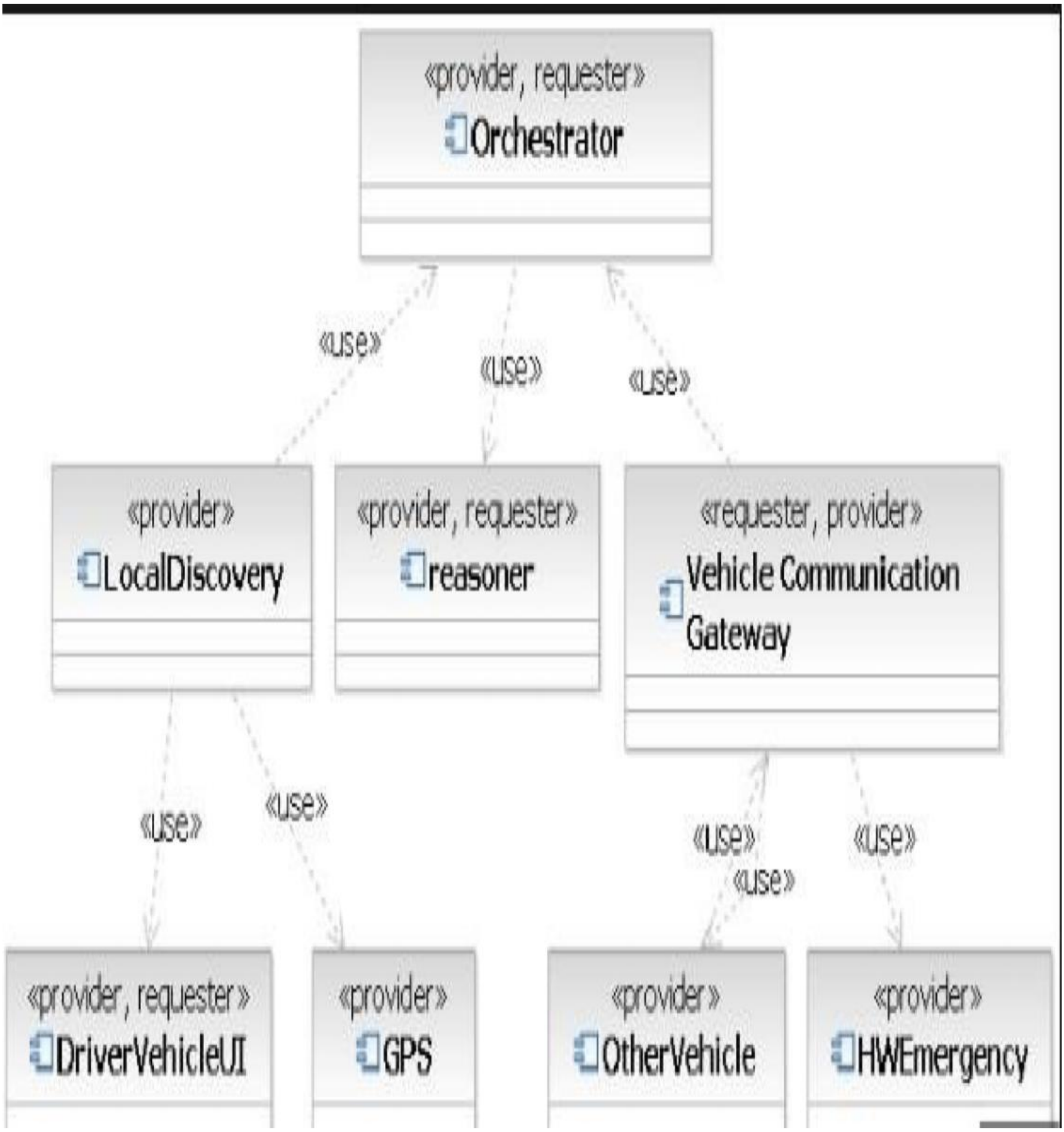


Fig. 3.1: Architecture Diagram of Auto-Spare Part

In fig 3.1 an architectural diagram is a graphical representation of the components and structure of a system or software application. It provides a high-level view of the system's architecture, including its major components, their relationships, and how they interact with one another. An architectural diagram can be used to communicate the overall design of a system to developers, stakeholders, and other interested parties.

In an auto spare part management project, the architectural diagram would likely include the following components:

User interface: This component would include the graphical user interface (GUI) that users interact with to search for and order spare parts.

Database: The database would store information about the available spare parts, including part numbers, descriptions, prices, and inventory levels.

Application server: This component would handle the business logic of the system, including validating user input, processing orders, and updating the database.

Integration with suppliers: The system may need to integrate with suppliers to automatically order spare parts when inventory levels fall below a certain threshold.

Reporting and analytics: The system may include reporting and analytics capabilities to help managers track inventory levels, sales trends, and other key performance indicators.

Overall, the architectural diagram for an auto spare part management project would provide a high- level overview of the system's components and how they interact with one another to enable efficient and effective management of spare parts inventory.

3.1.1 <u>Front End (UI) Design</u>

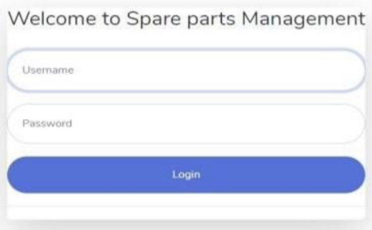


Fig. 3.2: login page of Auto-Spare Part Management. Here users can either create account or login to existing account.

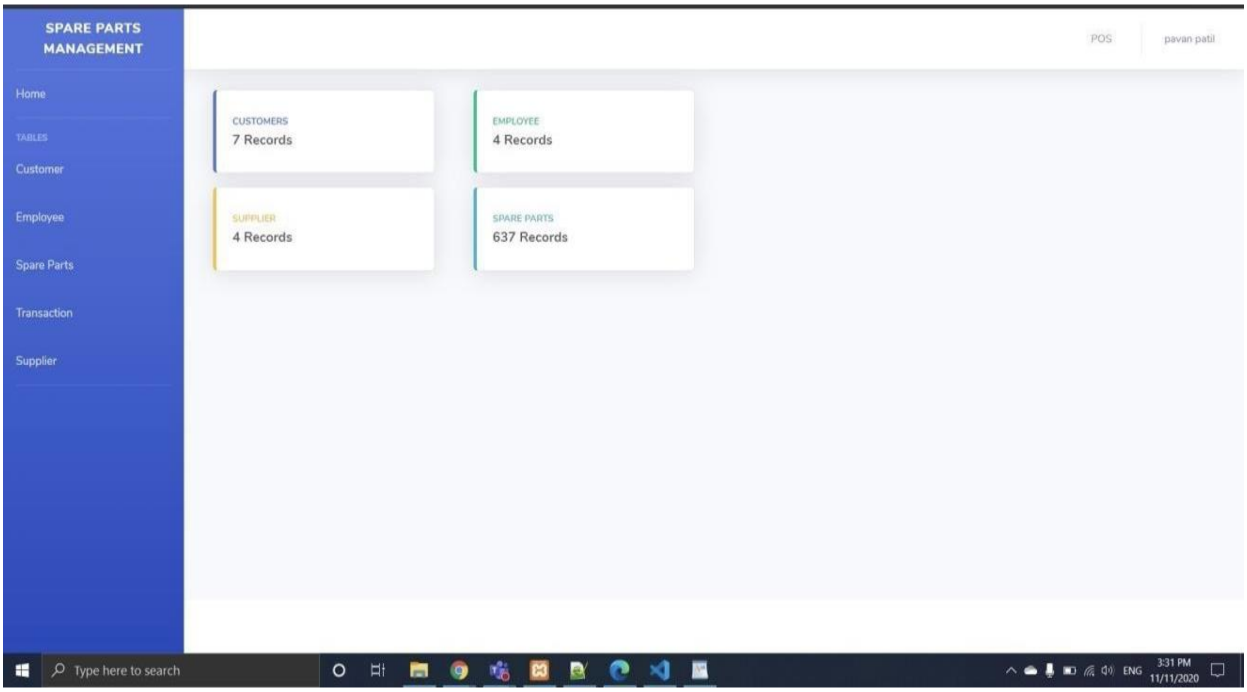
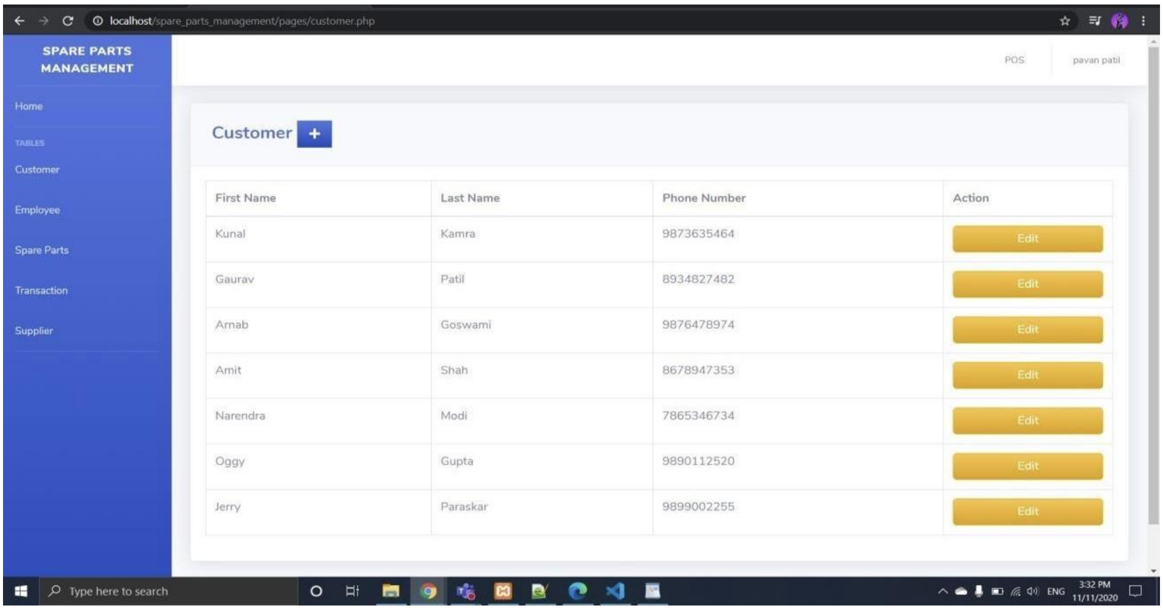


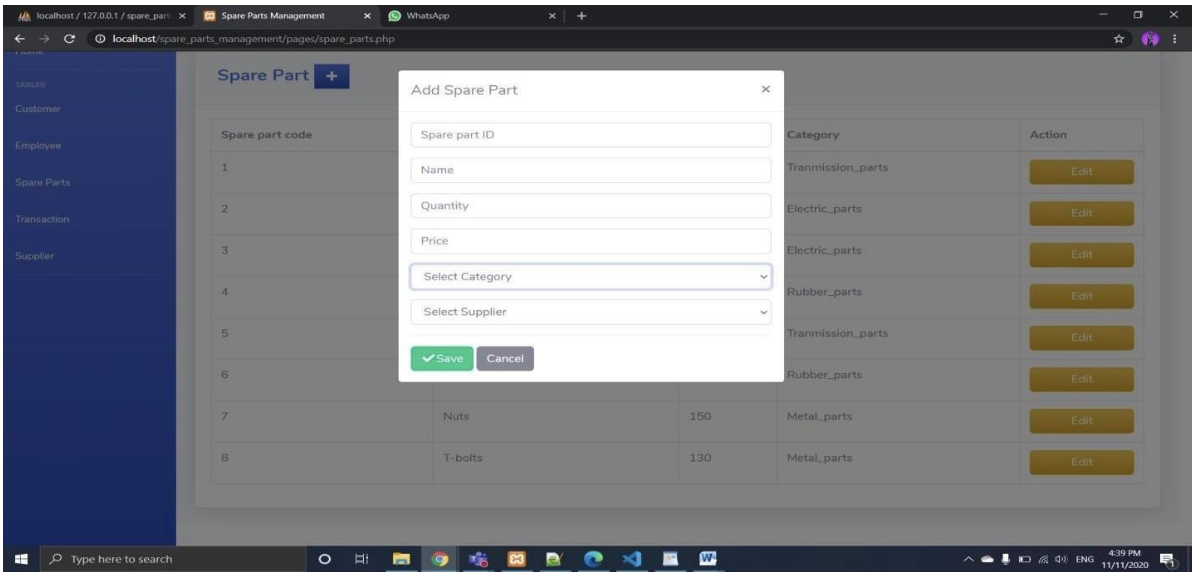
Fig. 3.3: Records page of Auto-Spare Part Management. It displays the available spare parts.



Customer +

First Name	Last Name	Phone Number	Action
Kunal	Kamra	9873635464	Edit
Gaurav	Patil	8934827482	Edit
Arnab	Goswami	9876478974	Edit
Amit	Shah	8678947353	Edit
Narendra	Modi	7865346734	Edit
Oggy	Gupta	9890112520	Edit
Jerry	Paraskar	9899002255	Edit

Fig. 3.4: Customer page of Auto-Spare Part Management. It displays the details of the customers.



Spare Part +

Add Spare Part

Spare part ID

Name

Quantity

Price

Select Category

Select Supplier

✓ Save Cancel

Spare part code			Category	Action
1			Tranmission_parts	Edit
2			Electric_parts	Edit
3			Electric_parts	Edit
4			Rubber_parts	Edit
5			Tranmission_parts	Edit
6			Rubber_parts	Edit
7	Nuts	150	Metal_parts	Edit
8	T-bolts	130	Metal_parts	Edit

Fig. 3.5: New spare part with their details of Auto-Spare Part Management. Here we update the stock of spare parts.

Software used

- Windows 11
- Linux
- DBMS
- NetBeans IDE 8.0.2
- SQL databases

3.1.2 **Back end (Database) design, software used with explanation**

- In addition to MySQL, there are a number of software tools that can be used to design the backend of an autospare management system, including:
- MySQL Workbench:MySQL Workbench is a visual database design tool that allows developers to create, manage, and manipulate databases.
- MySQL Workbench can be used to design database schemas, build queries, and generate reports.MySQL Workbench is a free and open-source tool that is available for Windows, macOS, and Linux.
- Apache:Apache is a widely-used web server that can be used to host web-based autospare management systems.
- Git:Git is a version control system that can be used to manage the source code for an autospare management system.
- PHP:PHP is a popular server-side scripting language that can be used to create dynamic web pages and interact with MySQL databases..
- The backend design of an autospare management system will depend on the specific requirements of the system and the business needs. However, there are some common features that may be included in the backend design of an autospare management system:
- Database design:The database design is a critical aspect of the backend design for an autospare management system.The database schema should be designed to efficiently store and manage data related to autospare inventory, suppliers, customers, orders, and other relevant data.
- APIs and web services:APIs and web services are used to enable communication between the frontend and backend components of the autospare management system.
- Security:Security is a crucial aspect of the backend design of an autospare management system, especially when dealing with sensitive data such as customer information and financial data.
- Performance:The performance of the backend design should be optimized to ensure fast and efficient processing of data.
- The backend design should be designed with maintenance and monitoring in mind, to ensure the system is operating smoothly and any issues can be quickly identified and resolved.

3.2 ER-DIAGRAM

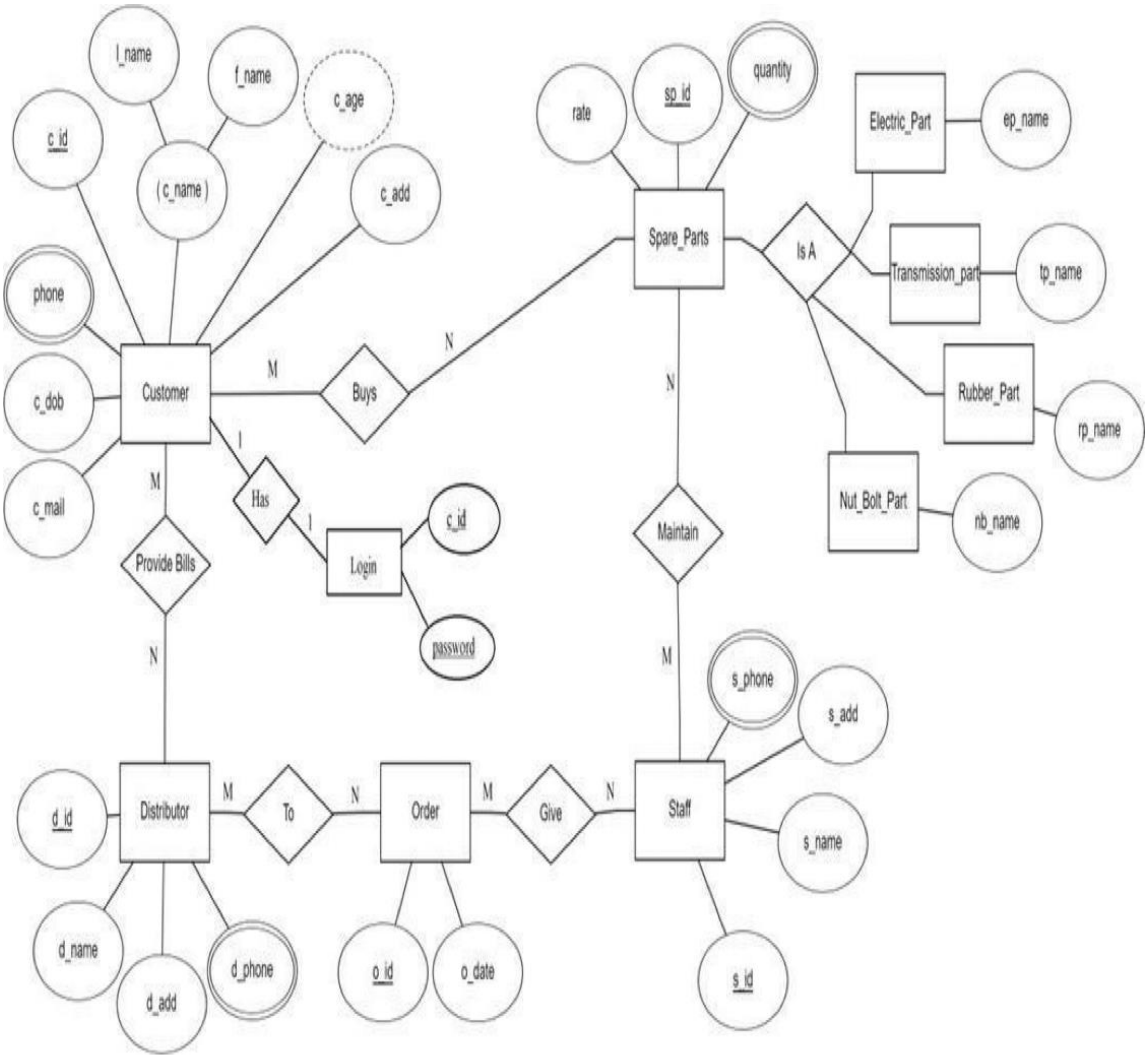


Fig. 3.6: ER Diagram of Auto-Spare Part Management

In fig 3.6 an Entity-Relationship (E-R) diagram is a visual representation of the entities and relationships involved in a system or software application. In an auto spare part management project, the E-R diagram would show the entities (such as parts, orders, customers, and suppliers) and the relationships between them.

In this diagram, there are several entities, including:

- Parts: This entity represents the spare parts that are available for purchase. Each part has a part number, description, and price.
- Orders: This entity represents the orders that customers place for spare parts. Each order has an order number, date, and status.
- Customers: This entity represents the customers who place orders for spare parts. Each customer has a unique ID, name, and contact information.
- Suppliers: This entity represents the suppliers who provide the spare parts. Each supplier has a unique ID, name, and contact information.

There are also several relationships between these entities:

- Parts can be ordered by customers: This relationship is represented by the "Ordered By" relationship between Parts and Customers. This indicates that each part can be ordered by one or more customers, and each customer can order one or more parts.
- Parts can be supplied by suppliers: This relationship is represented by the "Supplied By" relationship between Parts and Suppliers. This indicates that each part can be supplied by one or more suppliers, and each supplier can supply one or more parts.
- Orders can include multiple parts: This relationship is represented by the "Includes" relationship between Orders and Parts. This indicates that each order can include one or more parts, and each part can be included in one or more orders.

Using this E-R diagram, the auto spare part management system can be designed and developed to manage inventory levels, process orders, and maintain relationships with customers and suppliers.

USECASE- DIAGRAM

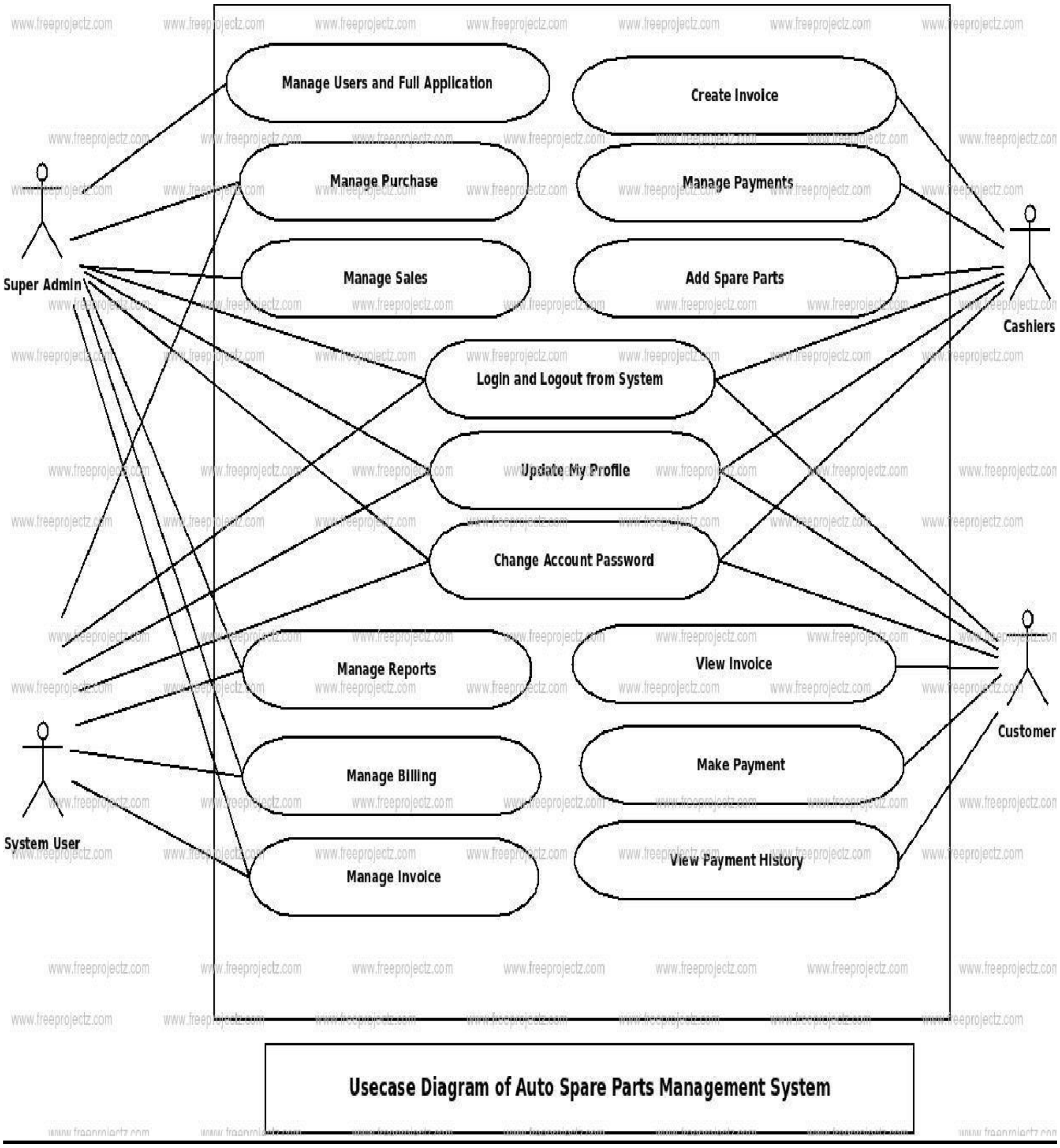


Fig. 3.6: Usecase Diagram of Auto-Spare Part Management

4.<u>List of Modules and Functionalities in each of the modules</u>

- Here is a list of some common modules and functionalities in autospare management.

- Inventory Management Module: Add, edit, and delete auto parts in the inventory Set up minimum and maximum stock levels for each part.

- Receive alerts when stock levels fall below the minimum threshold Track inventory movements and adjustments Generate inventory reports and analytics Set up barcode scanning for easy tracking and identification of parts Sales and Purchase Management Module.

- Manage customer information, including contact details and purchase history Track customer interactions, including inquiries, complaints, and feedback Assign and track follow-up tasks Generate customer reports andanalytics Accounting Module.

4.1 Type of Connectivity used for database access

- The type of connectivity used for database access in autospare management depends on several factors, including the size of the database, the number of users accessing the database, the level of security required, and the availability of network infrastructure. Here are some common types of connectivity used for database access in autospare management.
- Local access: In this type of connectivity, the database is installed on a local machine or server, and users access it directly from that machine. This can be an appropriate solution for small businesses with a single user or a small team.
- LAN access: In a LAN (Local Area Network) setup, the database is installed on a server that is connected to a network, and users access it from their individual machines. This type of connectivity is appropriate for small to medium-sized businesses with multiple users accessing the database from the same physical location.
- WAN access: In a WAN (Wide Area Network) setup, the database is installed on a server that is accessible over the internet, and users can access it from anywhere in the world with an internet connection. This type of connectivity is appropriate for businesses with remote workers or multiple locations.
- Cloud-based access: In a cloud-based setup, the database is hosted on a third- party server, and users access it through a web browser or a dedicated application. This type of connectivity is appropriate for businesses of all sizes, and it offers the flexibility and scalability to meet changing business needs.
- Mobile access: In a mobile setup, users can access the database through a mobile device, such as a smartphone or tablet. This type of connectivity is appropriate for businesses with workers who need to access the database while on the go.
- Regardless of the type of connectivity used, it is important to ensure that the database is secure and that access is restricted to authorized users. This can be achieved through the use of passwords, encryption, and other security measures.

5. CODING AND TESTING

Coding:

1. DDL and DML commands

```
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 11, 2020 at 06:43 AM
-- Server version: 10.4.14-MariaDB
-- PHP Version: 7.4.9


SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO"; START
TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT
*/;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;


--
-- Database: `spare_parts`
--
-
```

```
`LAST_NAME` varchar(50) DEFAULT NULL,
`GENDER` varchar(50) DEFAULT NULL,
`EMAIL` varchar(100) DEFAULT NULL,
`PHONE_NUMBER` varchar(11) DEFAULT NULL,
`JOB_ID` int(11) DEFAULT NULL,
`LOCATION_ID` int(11) DEFAULT NULL,
`HIRED_DATE` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `employee`
--


INSERT INTO `employee` (`EMPLOYEE_ID`, `FIRST_NAME`,
`LAST_NAME`, `GENDER`, `EMAIL`, `PHONE_NUMBER`, `JOB_ID`,
`LOCATION_ID`, `HIRED_DATE`) VALUES
(1, 'pavan', 'patil', 'Male', 'pavan@gmail.com', '9876567865', 1, 113, NULL),
(5, 'karan', 'nakum', 'Male', 'karan@gmail.com', '89763546334', 1, 163, NULL),
(6, 'hrushikesh', 'rajankar', 'Male', 'rushi@gmail.com', '9874657834', 2, 164, NULL),
(7, 'gopal', 'paraskar', 'Male', 'gopal@gmail.com', '7894653125', 2, 165, NULL);


--
-- Table structure for table `job`
--
CREATE TABLE `job` (
`JOB_ID` int(11) NOT NULL,
`JOB_TITLE` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `job`
--


INSERT INTO `job` (`JOB_ID`, `JOB_TITLE`) VALUES
(1, 'Manager'),
(2, 'Cashier');




--
-- Table structure for table `location`
--


CREATE TABLE `location` (
`LOCATION_ID` int(11) NOT NULL,
`PROVINCE` varchar(100) DEFAULT NULL,
`CITY` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `location`
--
```

```
INSERT INTO `location` (`LOCATION_ID`, `PROVINCE`, `CITY`) VALUES
        (113,       'pune',           'kondhwa'),
        (159,       'pune',           'kondhwa'),
        (160,       'pune',            'kothrud'),
        (161,       'nagpur',  'kuhi'),
        (162,       'amravati', 'morshi'),
        (163,       'pune', 'katraj'),
        (164,       'nagpur', 'narkhed'),
--
-- Table structure for table `manager`
--


CREATE TABLE `manager` (
`FIRST_NAME` varchar(50) DEFAULT NULL,
`LAST_NAME` varchar(50) DEFAULT NULL,
`LOCATION_ID` int(11) NOT NULL,
`EMAIL` varchar(50) DEFAULT NULL,
`PHONE_NUMBER` varchar(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;






--
-- Table structure for table `product`
--
```

```
CREATE TABLE `product` (
`PRODUCT_ID` int(11) NOT NULL,
`PRODUCT_CODE` varchar(20) NOT NULL,
`NAME` varchar(50) DEFAULT NULL,
`DESCRIPTION` varchar(250) NOT NULL,
`QTY_STOCK` int(50) DEFAULT NULL,
`ON_HAND` int(250) NOT NULL,
`PRICE` int(50) DEFAULT NULL,
`CATEGORY_ID` int(11) DEFAULT NULL,
`SUPPLIER_ID` int(11) DEFAULT NULL,
`DATE_STOCK_IN` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `product`
--
INSERT INTO `product` (`PRODUCT_ID`, `PRODUCT_CODE`, `NAME`,
`DESCRIPTION`, `QTY_STOCK`, `ON_HAND`, `PRICE`,
`CATEGORY_ID`, `SUPPLIER_ID`, `DATE_STOCK_IN`) VALUES

   (28,    '1'   'wires',        ",    1,    1,    300,   4,    19,           "),
           ,
   (29,    '1'   'wires',        ",    1,    1,    300,   4,    19,           "),
           ,
   (30,    '1'   'wires',        ",    1,    1,    300,   4,    19,           "),
           ,
   (31,    '1'   'wires',        ",    1,    1,    300,   4,    19,           "),
           ,
   (32,    '1'   'wires',        ",    1,    1,    300,   4,    19,           "),
           ,
```

(33,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(34,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(35,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(36,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(37,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(38,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(39,	'1' ,	'wires',	'',	1,	1,	300,	4,	19,	''),
(40,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(41,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(42,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(43,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(44,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(45,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(46,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(47,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(48,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(49,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),
(50,	'2' ,	'headlights',	'',	1,	1,	2500,	1,	17,	''),

```
-- Table structure for table `supplier`
--

CREATE TABLE `supplier` (
`SUPPLIER_ID` int(11) NOT NULL,
`COMPANY_NAME` varchar(50) DEFAULT NULL,
`LOCATION_ID` int(11) NOT NULL,
`PHONE_NUMBER` varchar(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `supplier`
--

IN
```

```
--
-- Table structure for table `users`
--

CREATE TABLE `users` (
`ID` int(11) NOT NULL,
`EMPLOYEE_ID` int(11) DEFAULT NULL,
`USERNAME` varchar(50) DEFAULT NULL,
`PASSWORD` varchar(50) DEFAULT NULL,
`TYPE_ID` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`ID`, `EMPLOYEE_ID`, `USERNAME`,
`PASSWORD`, `TYPE_ID`) VALUES

  (1,     1,     'admin1',     '6C7CA345F63F835CB353FF15BD6C5E052EC08E7A',

  1),

  (2,     5,     'admin2',     '315F166C5ACA63A157F7D41007675CB44A948B33',
  1),

  (3,     6,     'admin3',     '33AAB3C7F01620CADE108F488CFD285C0E62C1EC',
  2),
  (4,     7,       'admin4',      'EA053D11A8AAD1CCF8C18F9241BAEB9EC47E5D64
                                                        '

                                                        ,

  2);


-- Indexes for dumped tables

--



--

-- Indexes for table `category`

--

ALTER TABLE `category`

ADD PRIMARY KEY (`CATEGORY_ID`);



--

-- Indexes for table `customer`

--

ALTER TABLE `customer`

ADD PRIMARY KEY (`CUST_ID`);



--

-- Indexes for table `employee`

--

ALTER TABLE `employee`

ADD PRIMARY KEY (`EMPLOYEE_ID`),

ADD UNIQUE KEY `EMPLOYEE_ID` (`EMPLOYEE_ID`), ADD UNIQUE KEY
`PHONE_NUMBER` (`PHONE_NUMBER`), ADD KEY `LOCATION_ID`
(`LOCATION_ID`),

ADD KEY `JOB_ID` (`JOB_ID`);
```

```
--

-- Indexes for table `job`

--

ALTER TABLE `job`

ADD PRIMARY KEY (`JOB_ID`);



--

-- Indexes for table `location`

--

ALTER TABLE `location`

ADD PRIMARY KEY (`LOCATION_ID`);



--

-- Indexes for table `manager`



ALTER TABLE `manager`

ADD UNIQUE KEY `PHONE_NUMBER` (`PHONE_NUMBER`), ADD KEY `LOCATION_ID` (`LOCATION_ID`);



--

-- Indexes for table `product`

--

ALTER TABLE `product`

ADD PRIMARY KEY (`PRODUCT_ID`),

ADD KEY `CATEGORY_ID` (`CATEGORY_ID`), ADD KEY `SUPPLIER_ID` (`SUPPLIER_ID`);
```

```
--

-- Indexes for table `supplier`

--

ALTER TABLE `supplier`

ADD PRIMARY KEY (`SUPPLIER_ID`),

ADD KEY `LOCATION_ID` (`LOCATION_ID`);

--

-- Indexes for table `transaction`

--

ALTER TABLE `transaction`

ADD PRIMARY KEY (`TRANS_ID`),

ADD KEY `TRANS_DETAIL_ID` (`TRANS_D_ID`), ADD KEY `CUST_ID`
(`CUST_ID`);



--

-- Indexes for table `transaction_details`

--

ALTER TABLE `transaction_details` ADD PRIMARY KEY (`ID`),

ADD KEY `TRANS_D_ID` (`TRANS_D_ID`) USING BTREE;



--

-- Indexes for table `type`

--

ALTER TABLE `type`

--

ALTER TABLE `employee`

   MODIFY `EMPLOYEE_ID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=8;
```

```
--

-- AUTO_INCREMENT for table `location`

--

ALTER TABLE `location`

   MODIFY `LOCATION_ID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=166;



--

-- AUTO_INCREMENT for table `product`

--

ALTER TABLE `product`

   MODIFY `PRODUCT_ID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=665;



--

-- AUTO_INCREMENT for table `supplier`

--

ALTER TABLE `supplier`

MODIFY `SUPPLIER_ID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=21;



--

-- AUTO_INCREMENT for table `transaction`

--

ALTER TABLE `transaction`

MODIFY `TRANS_ID` int(50) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=15;
```

```
--
-- AUTO_INCREMENT for table `transaction_details`
--
ALTER TABLE `transaction_details`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=24;


--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
MODIFY `ID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;


--
-- Constraints for dumped table


--
--
-- Constraints for table `employee`
--
ALTER TABLE `employee`
   ADD CONSTRAINT `employee_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`),
   ADD CONSTRAINT `employee_ibfk_2` FOREIGN KEY (`JOB_ID`) REFERENCES `job` (`JOB_ID`);


--
-- Constraints for table `manager`
--
ALTER TABLE `manager`
```

```sql
  ADD CONSTRAINT `manager_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`);



--
-- Constraints for table `product`
--
ALTER TABLE `product`
  ADD CONSTRAINT `product_ibfk_1` FOREIGN KEY (`CATEGORY_ID`) REFERENCES `category`(`CATEGORY_ID`),
  ADD CONSTRAINT `product_ibfk_2` FOREIGN KEY (`SUPPLIER_ID`) REFERENCES `supplier` (`SUPPLIER_ID`);



--
-- Constraints for table `supplier`
--
ALTER TABLE `supplier`
  ADD CONSTRAINT `supplier_ibfk_1` FOREIGN KEY (`LOCATION_ID`) REFERENCES `location` (`LOCATION_ID`);



--
-- Constraints for table `transaction`
--
ALTER TABLE `transaction`
  ADD CONSTRAINT `transaction_ibfk_3` FOREIGN KEY (`CUST_ID`) REFERENCES `customer` (`CUST_ID`),



  ADD CONSTRAINT `transaction_ibfk_4` FOREIGN KEY (`TRANS_D_ID`) REFERENCES `transaction_details` (`TRANS_D_ID`);
```

```
--
-- Constraints for table `users`
--
ALTER TABLE `users`
ADD CONSTRAINT `users_ibfk_3` FOREIGN KEY (`TYPE_ID`) REFERENCES
`type` (`TYPE_ID`),
ADD CONSTRAINT `users_ibfk_4` FOREIGN KEY (`EMPLOYEE_ID`) REFERENCES
`employee` (`EMPLOYEE_ID`); COMMIT;


/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
```

Connectivity

```
<?php
                $db = mysqli_connect('localhost', 'root', '') or
                die ('Unable to connect. Check connection.');
                mysqli_select_db($db, 'Spare_parts' ) or

die(mysqli_error($db));
```

6. RESULTS AND DISCUSSIONS

Auto-spare parts software tool is used for creating, maintaining, and manipulating data in a database. They are used by organizations to store, manage, and retrieve large amounts of data efficiently. Here are some general results and discussions related to DBMS:

Results:

- Improved data management: Auto-spare parts Management system allows for the efficient and organized storage of data in a centralized location, making it easier to manage and access.
- Increased data security: It provides built-in security measures to ensure that only authorized users can access and modify data.
- Faster data retrieval: It allows for quick and easy retrieval of data, which can be critical in time-sensitive situations.
- Scalability: It can handle large amounts of data and can be easily scaled to accommodate future growth.

Discussions:

- Cost: Implementing a Auto-spare parts Management can be expensive, and there may be ongoing maintenance and licensing fees.
- Performance: Poorly designed databases or inefficient queries can result in slow performance and decreased productivity.
- Data integrity: Maintaining data integrity can be challenging, as errors can lead to incorrect data or inaccurate reports.
- Backup and recovery: It should have a robust backup and recovery plan in place to prevent data loss due to hardware failure, natural disasters, or other unforeseen circumstances.

Overall, it is an essential tool for organizations that need to store and manage large amounts of data efficiently and securely. Proper implementation and maintenance of a DBMS can result in improved productivity, data accuracy, and better decision-making.

7. Conclusion and Future Scope

At last we want to conclude that this application will decrease the overhead of the owner of the shop. It will be easily able to monitor the available stock present inthe shop. It will also helpful for the owner that whose money is pending or how much he has to pay to the supplier. The main motivation behind this application is to reduce tension and easily monitor the current condition of the shop. Now there isno need of books and paper in the shop and manually keep track of each record.

As you see, IT industry is booming the industry from last few decades. It will keep on booming as you go on and on. As you see our project is limited to one user\owner. But, we can further expand this to multi-user to multi-customer and there is way we can do that. We all know that there and various online shopping application and website available for their basic need. But, there is no application or website available for the Automobile Spare Parts. These were we can enhance this project further. We can make an application like Amazon were you get every single piece of the spare part on your application and need to go anywhere to find it. By providing the best product at best prices everyone going to come and visit the site. We can show the picture of the product with the offers on the particular occasion.

We can also add the feature that how amount of work has been done by the employee at your shop and how leaves taken by him/her. We can also add online payment methods with the help of Google Pay, PhonePe etc.

Reference

[1] http://docs.oracle.com/cd/B12037_01/server.101/b10739/create.htm

[2] http://docs.oracle.com/javafx/index.html

[3] http://stackoverflow.com/questions/15482623/javafx-1-3-listview

[4] https://www.tutorialspoint.com/php/index.htm

[5] https://www.w3schools.com/php/DEFAULT.asp

[6] Nayak, D.R.; Mahapatra, A.; Mishra, P. A survey on rainfall prediction using artificial neural network. Int. J.Comput. Appl. **2013**, 72, 16.

[7] Muka, Z.; Maraj, E.; Kuka, S. Rainfall prediction using fuzzy logic. Int. J. Innov. Sci. Eng. Technol. **2017**, 4, 1–5.

[8] Mislan, H.; Hardwinarto, S.; Sumaryono, M.A. Rainfall monthly prediction based on artificial neural network:A case study in Tenggarong Station, East Kalimantan, Indonesia. Procedia Comput. Sci. **2015**, 59, 142–151.