**Cleveland State University**

**Monte Ahuja College of Business**

**Department of Information Systems**

**Asha Department Store Business Database Management Project**

**Logo, company name

Description automatically generated**

**IST 634 (Enterprise Databases)**

**Submit to - Dr. Cheng, Wei-Hung**

**By,**

**Hasritha Puthalapattu (2842923)**

**12/05/2022**

Table of Contents

[1. Database Project Proposal 3](#_Toc117990485)

Topic Research & [Executive Summary 3](#_Toc117990486)

[Statement of Work 3](#_Toc117990487)

[Project Management Approaches, Schedule, and Milestones](#_Toc117990487) 4

[Methodology and Implementation Plan](#_Toc117990487) 5

[Expected Results: The Deliverables](#_Toc117990487) 5

[Personnel](#_Toc117990487) 6

[Supporting Facilities](#_Toc117990487) 6

2. Proposal Progressive Report V1……………………………………………………………..…7

Project Analysis………………………………………………………………………………...7

Mission Statement………………………………………………………………………………7

Mission Objective………………………………………………………………………………8

[3. Proposal Progressive Report V2](#_Toc117990488) 8

[List of Characteritics](#_Toc117990487) 9

[Preliminary Field List (Final), Value, Calculate List](#_Toc117990487) 9

[Final Table List and Table Structures](#_Toc117990487) 10

[4. Proposal Progressive Report V3](#_Toc117990489) 12

[Field Specification Sheet](#_Toc117990490) 12

[Table Matrix](#_Toc117990487) 26

[5. Proposal Progressive Report V4](#_Toc117990489) 26

[Relationship Characteristics](#_Toc117990490) 26

[Business Rule Specifications](#_Toc117990490) 28

[ER Diagram](#_Toc117990490) 37

[View Table with View Specicification](#_Toc117990490) 37

[Progressive Implementation Details](#_Toc117990487) 39

**Asha Department Store Business Database Management Project Proposal**

**Title:** Asha Department Store Business

* Project title: Asha Department Store Business
* Project Designer’s Information: Hasritha Puthalapattu (2842923), h.puthalapattu@vikes.csuohio.edu, 216-457-1330
* Submitted to: Dr. Cheng for IST 634 Enterprise Databases Fall 2022 Project
* Proposal date: September 12, 2022

**Topic Research and Executive Summary**

The Asha department store business is about proper management of different sub stores under one roof. These features include controlling data on different items, employees, managers, customers, invoices, etc. An effective technique to manage the information for the department store is provided by this system which enables the customer to buy and pay for the purchased things.

**Statement of Work**

* Project Description:

This project is based on how products are billed and sold at a store. The initial task involves entering the items, their stock level, and names of the things that are going on sale into the system. The user is required to enter the item name and quantity as soon as the consumer purchases the products and approaches the billing counter.

The software is used to manage the sales activity carried out in a department store, to keep track of stock information, and to keep track of the sales made on a given day or month or year. The users will spend less time calculating and the sales activity will be completed in a matter of seconds as opposed to the manual system that requires the user to write it down, which is a long process. This will result in less paperwork and allow the user to devote more time to overseeing the store. The project will be simple to use and friendly to the users.

The system will show every item whose name begins with the letter the user has chosen. He has a choice among those that are exhibited. The final step is to create a unique bill for each consumer. The database will record this and view any periodic records whenever it wants. The department store places an order with a designated vendor and purchases any unavailable goods. The total amount gathered during the sales activity will be subtracted before the amount is paid.

* Challenges:

1. Logical design of Relational Database
2. Learning Microsoft Access application for design database
3. Learning C# programming to create an application that helps to connect to Microsoft Access Database.
4. Interviewing the customer and the project team will help you understand and assess the client's objectives and desired results (expectations). This will help you focus your project request

* Scope:

Asha department store system software is developed by using a Database Management System that will help to automate department stores. The store has the database management system (DBMS) and various interfaces such as Vendor Order, Stock entry, Display Customer information interface.

Project Benefits:

The project's benefits include, saving time, security, backing up, generating reports, and restoring data.

1.Timesaving: Automation of operations will help in saving time significantly for track of store sales daily.

2.Security: provides security to information, mainly securing customer details and authenticating users to prevent access from unauthorized users.

3.Generating Reports: The system will generate reports that provide a list of available products to write down each time an item is removed from the Supermarket.

4. It helps protect the database from theft, physical destruction, and computer worms and malware to make it more secure than file-based systems.

**Project Management Approach, Schedule, and Milestones**

This project is done by a single person as the given time for the project by the client is for 5 weeks it is necessary to divide the project accordingly while during the first week of the project all the requirements and resources to be gathered and then implement a design for the project. During the second week of the project, we should consult the developer and present our idea and budget required and the tools used to be mentioned and the actual implementation to be started during the remaining weeks all the development part is done later testing and the final project is ready for the end user [1].

* Stage 1 Analysis: Identify and Understand the Business Model and create a rough skeleton of the database. (Analysis topic business)
* Stage 2 Logical Design: Understanding Microsoft Access and create a database layout.
* Stage 3 Physical Design: Understanding, and articulating database as per drawn user requirements and guidelines using the Microsoft Access database
* Stage 4 Implementation: Design the end-user UI (User-Interface) with C# as the programming language with visual studio as a supporting platform for the code development in compliance with the Microsoft access database.
* Stage 5 Modification and Maintenance: Testing data- accuracy, request response such that if not as desired, make relevant modifications.

The schedule is mentioned below:

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Week | Date | Tasks assigned |
| 1 | Week 1&2 | 08/29/22-09/12/22 | Stage 1 Analysis: Knowing purpose of database, taking inputs, checking resources, and consulting a business analyst. |
| 2 | Week 3&4 | 09/13/22-09/26/22 | Stage 2 Logical Design: Understanding MS Access. |
| 3 | Week 5&6 | 09/27/22-10/10/22 | Stage 3 Physical Design: Designing of database |
| 4 | Week 7&8 | 10/11/22-10/24/22 | Stage 4 Implementation: Application designing using C# |
| 5 | Week 9&10 | 10/25/22-11/07/22 | Stage 5 Modification and maintenance: Validating and send for production. |

**Methodology**

The Asha Department Store Business being developed must possess the ability to serve the necessities of both the workers and customers. This proposed system also possesses higher reliability and it must be quicker, easier, and more intuitive in nature. The desired qualities that the newly designed department store management system must contain are as follows:

Enhance documentation operation, Automating the generation of any reports or documentations. Quick in terms of reaction time. Lessening of the cost incurred for the processing. The department store management system is based on a desktop computer running Windows OS-based application. The major functionalities impacted by our management system are given below.

1. Tracking the record of the total products that got sold.
2. Estimate the bill.
3. Modify the Graphical UI of the system.
4. Keeping track of products along with its price data and other data.

Thus, our research project is dependent on the product stocking and transaction of sales that take place in the department store businesses.

**Implementation Plan**

Outlining the requirements and expected outcomes would be created before implementation. Next step, which entails developing an implementation strategy, such as coding, and then establishing a logical conceptual and physical design based on ER-Diagrams and use-cases. Further, proceed with the database's real time implementation. Making use of MS Access, ensuring that all the specifications are met, and integrating it with C# using visual studio as a platform for the user interface.

I will be learning the prerequisite for the database (such as models, queries) which is DBMS and dive into deep understanding of relational database management system, applying using Microsoft access and learning C# for connecting the database with the frontend design using visual studio IDE. These helps the project to keep track of all the records in the database and manages the products availability.

There will be a few changes in the plan during its development phase later as we go deeper into the project, but the outcome should be according to the initial idea.

**Milestones**

1) Start Project Requirements gathering

2) Developing Database Design

3) Prepare for the application for Testing

4) Project Delivery

**Database**

Microsoft Access is a database and information management tool that lets you save data for future reference, reporting, and analysis. When working with large amounts of data in excel or another spreadsheet program, access can assist you in overcoming these limitations.

**Application**

Microsoft Visual Studio is a programming environment where one can fabricate an application by outwardly making the user interface first, followed by writing code. Visual Essential fundamental projects could take reams of program code to write in C#.

**Expected Results: The Deliverables**

By developing a departmental store application, they can access it efficiently. The store owners can view the details of the employees, the number of staff working daily, and the profit or loss details with ease. The Store manager can use the database system to view the no of people working in the store and can view the accurate details of the orders. Sales can be tracked on the system by monitoring the database. The system also provides for processing the return of purchased items. the primary goal of this system is to develop a self-contained sale to keep track of records, the purchase, and the return of products.

Database:

Microsoft Access: Microsoft Access is a database and information management tool that lets you save data for future reference, reporting, and analysis. When working with large amounts of data in Excel or another spreadsheet program, access can assist you in overcoming these limitations.

Application:

Visual Studio: Microsoft Visual Fundamental was a programming climate where one could fabricate an application by outwardly making the UI first and afterward adding code. Conversely, even the littlest Visual Essential entire projects could take reams of program code to write in C # or C++.

**Personnel**

Project Manager: Hasritha Puthalapattu

Email: h.puthalapattu@vikes.csuohio.edu

Education: Pursuing master’s in management information systems

I have people skills and strong analytical skills, which I use to highlight my advertising abilities. I know a lot about database management systems. These initiatives enable me to develop my abilities and stand out in the forthcoming projects.

**Supporting Facilities**

Database: Designed in Microsoft Access. Application Program Language: C#

**References**

[1] Shahrear Amir Rajib. “Departmental Store Management System”. Problem Analysis and Methodology,15, 2017.

http://dspace.ewubd.edu:8080/bitstream/handle/123456789/2349/Shahrear\_Amir\_Rajib.pdf?sequence=1&isAllowed=y.

**Project Progressive Report**

**Reporter:** Hasritha Puthalapattu

**Project Analysis (09/13/2022):**

The Asha departmental store database for the business system aims to provide a solution to existing challenges. To create a database that can track sales data, customer products purchase, and store purchase from the suppliers, the department manager would need to input the necessary information into the database daily. Additionally, the department manager should also include access past data to make predictions about future performance. By creating reports that show different aspects of the store's performance over time, as a result the manager can track changes in key areas and make necessary adjustments. Finally, this is a detailed research project that would require input from several different departments within the store.

**Mission Statement (09/15/2022):**

The mission of the Asha departmental store is to ensure data integrity, consistency, and accuracy and to promote sales effectiveness through ongoing, systematic planning and evaluation efforts to support manager and staff in achieving the sales track and profits accurately. monitor all the sales from time to time, to improve the efficiency of the store services and profits and keep track of sales in data base. By storing the required information of each that help us to keep note of products, sales, employees, store information etc. Database is used for the data to store. The database will include clear information about the daily sales and past sales on each month. This stored information will be accessed only by authorized users and will help the store to easily detect any losses that might occur to the business to bring down.

## **Support Dialogues:**

Developer: What is the project's main goal?

Manager: The Asha department store database management project is aimed to assist in maintaining the products history and keeping the customer records, and ensuring their information is secure. The accessibility of the database will also help facilitate the increasing number of customers.

Developer: How do you track your operation records for the business on the daily purchases and sales made?

Manager: Usually, we use the receipt book record to track down our daily operations.

Developer: What are the challenges that you face by using receipt book records to track down your daily operations?

Manager: The receipt book record is time consuming. There are a lot of errors in manual calculations and the procedure must be repeated two to three twice a day. It is very hard to find a misplaced receipt book.

Developer: What are the reasons for implementing a DBMS?

Manager: To improve the quality service to the customers by keeping track of their interests.  Challenges with record keeping and demand a better plan in our business to be safe. Therefore, having a database can help us facilitate running the business and help in serving a large group of customers.

Developer: How soon do you require the database installed at the store?

Manager: If possible, the database must be ready in a month to help with the challenges of wasting energy, time, and resources.

## **Mission Objective (09/16/2021):**

The mission objectives of Asha Department Store Database to help customers own their time by delivering the most convenient and seamless store experience and offering great values in apparel, home, beauty, accessories and more. We're guided by our purpose to create a brighter future with bold representation and easy to handle all the store products. Maintain the department store report. Keep track of the sales and inventory at the store. Keep track the predictions about the future sales trends. Keep the store’s records safe from loss or damage.

## **Support Dialogues:**

Developer: What information would you like to be able to track with this database?

Manager: The information that can be tracked using this department store database includes sales data, inventory levels, customer spending habits, products list, and store purchases from the suppliers.

Developer: Can you think of any other information you would like to list of things to be able to track with this database?

Manager: I think, it would be helpful to have access to past data so that we can make predictions about the future.

Developer: That sounds like a great idea! I'll add that capability to the list of things we want to include in this database.

Manager: Great, thank you!

Developer: Is there anything else you would like to share about how you envision using this database?

Manager: We are thinking about the ability to create reports that show different aspects of the store's performance over time.

Developer: We will add that capability to the list of things we want to include in this database.

Manager: Thank you!

**The List of Subjects (09/20/2022):**

* Employee
* Customer
* Product
* Sale
* Store
* Supplier
* Parking lot

## **Support Dialogues:**

Developer: What methods did you use to track your daily operations in the store?

Manager: We used to keep track of our daily operations through the old method of storing the receipt books. After closing the store in the evening, the staff must take some time to ensure all the receipt books are updated and match with all the sales and purchases conducted that day. Also, a conclusion must be made of whether the store gained profits or suffered losses that day.

Developer: Does the database be protected from outsiders accessing it?

Manager: Yes, the unauthorized people cannot reach the data from our databases because it's secured, and every login is monitored.

## **The List of Characteristics (09/27/2022)**

* **Employee** -Employee Name, Employee\_id, Contact \_number, Receptionist
* **Customer**- Customer\_id, customer name, address
* **Product**- Product\_name, Product\_id, Product\_type, Quantity Sold
* **Supplier**-Supplier\_id, name, Product\_id, Quantity delivered
* **Sales**-Sales\_id, sales date, Sales amount, Employee\_id
* **Users**- User name, password, email address
* **Store** – area code, address, location, parking lot

## **Support Dialogue**

Developer:How do you ensure that all your daily records are correct?

Manager:There is no way to ensure that all records are correct in the department store. However, all employees can help to ensure that records are accurate and up to date by regularly checking and updating them. Additionally, the store can develop and implement policies and procedures to help ensure that records are accurate and up to date.

Developer: How do you keep track of sales?

Manager: Sales will be noted on the system on time and checked in the database on daily basis

**Preliminary Field List (10/01/2022)**

* Product Sales
* Supplier Name
* Supplier Address
* Employee Address
* Product ID
* Product Name
* Sales Quantity
* Sales Total Amount
* Sales Date
* Customer Shipping Address
* Sales total amount
* Product Delivery Status
* Totals amount
* Supplier Product Name
* Shipping address
* Supplier Location
* Deliver Status
* Delivery Date
* Quantity In
* Quantity Out
* Payment Method
* Customer Location

**Final Preliminary List**

* Product Sales
* Supplier Name
* Supplier Address
* Employee Address
* Product Id
* Product Name
* Sales Total Amount
* Sales Date
* Customer Shipping Address
* Sales total amount
* Product Delivery Status
* Supplier Product Name
* Shipping Address
* Location
* Delivery Date
* Quantity In
* Quantity Out
* Payment Method

**Calculate List**

* Sales Quantity
* Total Amount

**Value list**

* Location
* Deliver Status

**Preliminary Table List (10/04/2022)**

* Employee: Stores all the information about the employees working in Asha department store.
* Customer: Stores all the customers history for the products purchased
* Product: This table will keep all the information for the items being sold at the store
* Supplier: The supplier table will hold information for those people supplying products and goods to the store
* Sales: Sales table will keep information about all the transactions made.
* Users: Users are the staff who use the system (admin, cashier etc.)
* Parking: Users can park their cars

**Final Tables list (10/06/2022)**

|  |  |  |
| --- | --- | --- |
| **Tables** | **Description** | **Type** |
| Employee | This is used by all the employees working in the grocery store. Information like Employee id, Names, Salary, Phone number, Address, and State are all recorded here. | Data |
| Customer | This contains all the customers who have purchased the grocery. It helps the management make a sound decisions.eg Customer id, name, contact, address… | Data |
| Products | This table saves all the products available in the store and those in the store. Data like Product id, name, quantity sold, Quantity remaining, and Supplier Id is saved here. | Data |
| Supplier | This shows all the data for suppliers delivering items to the store. It will include information such as Supplier id. Name, Contact, Product Id, Product Name, Quantity delivered, Amount Paid, Pending Balance etc. | Data |
| Sales | This contains information about sales that have taken place in the store. Sales Id, Sale Date, Sale Amount, Customer Name etc. will be recorded | Data |
| Users | The table will be a list of the system users. It will record User id, username, user password | Data |
| Parking | This contains information about parking lot for the store | Data |
| Supplier\_Suplies\_Products | This is a linking table that links suppliers with the products. It contains data such as SuppId, ProdId, and Product name. | Linking |

**Table Structure (10/07/2022)**

|  |  |
| --- | --- |
|  |  |

* PK: Primary Key
* FK: Foreign Key
* CK: Candidate key

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Employee** | **Customer** | **Products** | | **Sales** | **User** | **Supplier** | **Employee Contact** | **Supplier\_Supplies\_Products** | **StateTable** | **Parking Lot** |
| Employee\_Id (pk) | Cust\_Id (pk) | Prod\_Id(pk) | | Sales\_Id(pk) | User\_Id (pk) | Sup\_Id(pk) | Emp\_Id (  fk) | Supp\_Id | State\_Abbr | Parking lot number |
| Employee\_Names | Cust Names | Prod\_Name | | Sales\_Date | Username | Sup\_Name | Emp\_Name | Prod\_Id  (fk) | StateName | Parking address |
| Phone Number(ck) | Prod\_id(fk) | Qty\_sold | | Sale\_Amount | Password | Prod\_Id | Contact |  | Postal\_Code (pk) | Parking location lattitude |
| Adress | Adress | Sup\_Id(fk) | |  | Emp\_Email (fk) | Prod\_Name |  |  |  |  |
| State | Contact |  | | Cust\_Id( ck) |  | Qty\_Delivered |  |  |  |  |
| Salary | State |  | | Prod\_Id |  | Amount\_paid |  |  |  |  |
| Employee Email | Shipping\_Address |  | |  |  | Pending\_Balance |  |  |  |  |
|  | | |  | | | |  |  |  |  |

**Field Specification Sheet:10/19/2022**

**Customer Table**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **Cust­\_Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **Customer** | Source Specification: |
| Alias (es): |
| Shared By: **All Customers** | |
| Description: Cust­\_Id Provides a unique id that identifies the Customers from others | |
| **Physical Elements:** | |
| Data Type: STRING | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Phone Number | Specification Type:  Unique  Generic  Replica |
| Parent Table: Customer | Source Specification: |
| Alias (es): |
| Shared By: Customer | |
| Description: Prod\_Id help in identifying the products that the customer has purchased. | |
| **Physical Elements:** | |
| Data Type: INTEGER | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 10 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: 0-9 | |

**Employee Table**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **Emp Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **Employee** | Source Specification: |
| Alias (es): |
| Shared By: **All Employees and Sales** | |
| Description: Emp Id provides a unique id that identifies the employees from others. It also helps identify which employee made a certain sale. | |
| **Physical Elements:** | |
| Data Type: **String** | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Cust\_Id | Specification Type:  Unique  Generic  Replica |
| Parent Table: Employee | Source Specification: |
| Alias (es): |
| Shared By: Employee, Customer | |
| Description: Cust\_Id helps identify customers that the employee has served. It helps gauge the productivity of an employee. | |
| **Physical Elements:** | |
| Data Type: String | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 10 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

**Products Table**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **Pro\_Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **Product** | Source Specification: |
| Alias (es): |
| Shared By: **All Products, Sales, Suppliers** | |
| Description: Prod\_Id Provides a unique id that identifies the products from other products. | |
| **Physical Elements:** | |
| Data Type: **String** | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Sup\_Id | Specification Type:  Unique  Generic  Replica |
| Parent Table: Product | Source Specification: |
| Alias (es): |
| Shared By: Product, Supplier, | |
| Description: Sup\_Id help in identifying the supplier that is responsible for delivering certain products and items. | |
| **Physical Elements:** | |
| Data Type: String | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 10 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

**Sales Table:**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **Sales\_Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **Sales** | Source Specification: |
| Alias (es): |
| Shared By: **Sales** | |
| Description: Sales\_Id Provides a unique id that identifies the sales from others | |
| **Physical Elements:** | |
| Data Type: **String** | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Emp\_Id | Specification Type:  Unique  Generic  Replica |
| Parent Table: Sales | Source Specification: |
| Alias (es): |
| Shared By: Sales, Employee | |
| Description: Emp\_Id help in identifying the employer who oversaw a certain sale. | |
| **Physical Elements:** | |
| Data Type: String | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 10 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Cust\_Name | Specification Type:  Unique  Generic  Replica |
| Parent Table: Sales | Source Specification: |
| Alias (es): |
| Shared By: Sales, Customer | |
| Description: Cust\_Name helps identify the customer who made the purchase. | |
| **Physical Elements:** | |
| Data Type: VarChar | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 30 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

**User Table:**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **User\_Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **User** | Source Specification: |
| Alias (es): |
| Shared By: **Users** | |
| Description: User\_Id provides a unique id that identifies all the users of the system. | |
| **Physical Elements:** | |
| Data Type: **String** | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Emp\_Email | Specification Type:  Unique  Generic  Replica |
| Parent Table: User | Source Specification: |
| Alias (es): |
| Shared By: Employee, User | |
| Description: Emp\_Email help in identifying a certain employee in the system. | |
| **Physical Elements:** | |
| Data Type: VarChar | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 10 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

**Supplier Table**

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: **Sup\_Id** | Specification Type:  Unique  Generic  Replica |
| Parent Table: **Supplier** | Source Specification: |
| Alias (es): |
| Shared By: **Suppliers, Products** | |
| Description: Sup\_Id provides a unique id that identifies the supplier from others in the database. It also helps know which supplier supplies certain products. | |
| **Physical Elements:** | |
| Data Type: **String** | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: **10** |
| Decimal Places: **None** |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Prod\_Name | Specification Type:  Unique  Generic  Replica |
| Parent Table: Supplier | Source Specification: |
| Alias (es): |
| Shared By: Supplier, Product | |
| Description: Prod\_Name describes the product that the supplier is delivering. | |
| **Physical Elements:** | |
| Data Type: VarChar | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 30 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

|  |  |
| --- | --- |
| **FIELD SPECIFICATIONS** | |
| **General Elements:** | |
| Field Name: Sup\_Name | Specification Type:  Unique  Generic  Replica |
| Parent Table: Supplier | Source Specification: |
| Alias (es): |
| Shared By: Supplier | |
| Description: Sup\_Name describes the name or the company of the products supplier. | |
| **Physical Elements:** | |
| Data Type: String | Character Support:  Letters (A-Z)  Keyboard (&\*?)  Numbers (0-9)  Special(€©β) |
| Length: 30 |
| Decimal Places: None |
| **Logical Elements** | |
| Key Type:  Non  Primary  Foreign  Candidate | Edit Rule:  Enter Now, Edits Allowed  Enter Now Edits, Not Allowed  Enter Later, Edits Allowed  Enter Later, Edits Not Allowed  Not Determined at This Time |
| Key Structure:  Simple  Composite |
| Uniqueness:  Unique  Non-Unique |
| Null Support:  No Nulls  Nulls Allowed |
| Value Entered By: User  System |
| Required Value:  Yes  No |
| Range Of Values: | |

**Table Matrix (11/10/2022)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Employee | Product | Sales | User | Supplier | Customer |
| Employee | Null | Null | 1: N | 1:1 |  | 1: N |
| Product | Null | Null | 1: N | Null | 1:1 | 1: N |
| Sales | 1:1 | 1: N | Null | Null | Null | 1:1 |
| Store | 1:1 | Null | Null | Null | Null | Null |
| Supplier | Null | 1: N | Null | Null | Null | Null |
| Customer | Null | 1: N | Null | Null | Null | Null |

**Relationship Characteristics for each table - Deletion Rules, Type of Participation, and Degree of Participation**

**1) Customer Table**

**Deletion Rules**

* When a Cust­\_Id becomes inactive, you cannot delete customer record. Here you use a DENY rule.
* A RESTRICT rule is used when a customer record is connected to a product and cannot be deleted.
* All products associated with the customer must also be deleted from the customer table (Cascade rule)
* All products linked with the product’s customer must also be erased (Nullify rule)

**Type of Participation**

Mandatory – Before inserting any record from product, there must exist at least one customer record.

**Degree of Participation**

Cust\_Id has a degree of participation of (1, N) because the customer should make a minimum of 1 order before being recorded and no maximum number of products.

**2) Supplier table**

**Deletion Rules**

* Supplier record cannot be removed since it must be declared unavailable (DENY RULE)
* If their products records related to the supplier, then one cannot delete Supplier (RESTRICT RULE)
* One must first delete the product records associated with the product from the product table and then delete the supplier from the supplier table. (RESTRICT RULE)
* All supplier records associated with the product must also be deleted from the supplier table. (CASCADE RULE.)
* The quantity remaining for the product associated with the supplier must also be deleted. (NULLIFY)

**Type of Participation**

Mandatory – A supplier must exist before adding any data to the product record

**Degree of Participation**

Supplier has a degree of participation of (1, N) because the supplier is allowed to deliver one or more products.

**3) Employee Table**

**Deletion Rules**

* One cannot remove Employee Record (DENY.)
* One cannot delete Employee records if there is related sales. (RESTRICT.)
* One must delete the sales associated with the employee from the sales table and then delete the employee from the employee table. (RESTRICT.)

**Type of Participation**

Optional **–** There must be no employees before making any sales

**Degree of Participation**

Emp\_Id has a degree of participation of (0, N) because employee makes unlimited sales

**4) Sales table**

**Deletion Rules**

* One cannot delete Sales records if there are related Customer records. (Restrict rule.)
* One cannot delete the sales record; you must designate the customer as closed. (Deny rule.)

**Type of Participation**

Optional – There must be at least one record of the menu before entering any records to other tables.

**5) Product table**

**Deletion Rules**

* One cannot delete the product record if there are related sales records. (Restrict rule.)
* One cannot remove the product record; you must designate the sales as close. (Deny rule.)
* You must first delete the sales associated with the product from the sales table and then product from the product table. (Restrict rule.)

**Type of Participation**

Mandatory– There must be at least one record of product before entering any records to other tables.

**6) User Table**

**Deletion Rules**

* One cannot remove the user record; you must designate the employee as closed. (Deny rule.)
* One cannot delete user records if there are related employee records. (Restrict rule.)
* You must first delete the employee associated with the user from the employee table and then delete user from the user table. (Use the Restrict rule.)

**Type of Participation**

Mandatory– There must be at least one record of a user item before entering any records to other tables.

**BUSINESS RULE SPECIFICATION (11/13/2022):**

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Every sale must be associated with the respective employee who completed it. | | | | Constraint: Emp\_Id should be used to identify the employee who oversees Sale | | | | Type:  Database Oriented  Application Oriented  **Structures affected** | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |  |  | | --- | | Field Name: Emp\_Id | | Table Name: Sale, employee |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | Required Value for Emp Id was set to “Yes”  Null support was set to “No Null” | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: No Invalid Cust\_Id is allowed. | | | | Constraint: Entities of Emp\_Id entered in the sales table should be restricted to the values of Emp\_Id in the employee table. | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific  **Structures affected** | Test On:  Insert  Update  Delete |  |  | | --- | | Field Name: Cust\_Id | | Table Name: Sales, Employee |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | The type of participation (Employee is Optional, Sales is Mandatory). The Range of Values “Any value within Cust\_Id field of the sales table. A Deny deletion rule was defined for the relationship between the two tables. | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Every user should have an employee detail that they use to access the system. | | | | Constraint: Emp\_Email should be used to identify employees. | | | | Type:  Database Oriented  Application Oriented  **Structures affected** | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |  |  | | --- | | Field Name: Emp\_Email | | Table Name: Employee, Users |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | Null support was set to “No Null”  Required Value for Emp\_email was set to “Yes” | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Emp\_email should be valid | | | | Constraint: All values for Emp\_email entered in the user table should exist in the records of Emp\_email in user table. | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |   **Structures affected**   |  | | --- | | Field Name: Emp\_email | | Table Name: Employee, User |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | The type of participation (employee is Optional, user is Mandatory). The Range of Values “Any value within Emp\_email field of the emp table. A Deny deletion rule was defined for the relationship between the two tables.  . | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Every Sales should have a product in it. | | | | Constraint: Prod\_Id identifies the products in the sales | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific  **Structures affected** | Test On:  Insert  Update  Delete |  |  | | --- | | Field Name: Prod\_Id | | Table Name: Product, Sales |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | Null support was set to “No Null”  Required Value for Prod\_Id was set to “Yes” | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Invalid Prod\_Id is not allowed. | | | | Constraint: All values for Prod\_Id entered in the sales table should exist in the records of Prod\_Id in product table | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |   **Structures affected**   |  | | --- | | Field Name: EmpId | | Table Name: Employee, Kitchen Operation |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | The type of participation (product is Optional, sales is Mandatory). The Range of Values “Any value within Prod\_Id field of the employee table. A Deny deletion rule was defined for the relationship between the two tables. | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Every product should have a supplier | | | | Constraint: SuppId should be used to identify the supplier who delivers it. | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |   **Structures affected**   |  | | --- | | Field Name: SuppId | | Table Name: Supplier, Product |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | Null support was set to “No Null”  Required Value for SuppId was set to “Yes” | |

|  |
| --- |
| **Business Rule Specifications** |
| **Rule Information**   |  |  |  | | --- | --- | --- | | Statement: Invalid SuppId is not allowed. | | | | Constraint: All values for SuppId entered in the product table should exist in the records of SuppId in supplier table | | | | Type:  Database Oriented  Application Oriented | Category:  Field Specific  Relationship Specific | Test On:  Insert  Update  Delete |   **Structures affected**   |  | | --- | | Field Name: EmpId | | Table Name: Employee, Kitchen Operation |   **Field elements affected**   |  |  |  | | --- | --- | --- | | **Physical elements** | | | | Data Type | Length | Character Support | | **Logical elements** | | | | Key Type  Key Structure  Uniqueness | Null Support  Values Entered By  Required Value | Range of Values  Edit Rules |   **Relationship Characteristics allowed**   |  |  |  | | --- | --- | --- | | Deletion Rule | Type of Participation | Degree of Participation |   **Action Taken**   |  | | --- | | The type of participation (Product is Optional, Supplier is Mandatory). The Range of Values “Any value within SuppId field of the employee table. A Deny deletion rule was defined for the relationship between the two tables. | |

**ER Diagram:**

A picture containing text, map, indoor

Description automatically generated

**View Specification**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Name: Customer Purchased List | Type:  Data  Aggregate  Validation | | Description: The Manager can see the history of purchases made by customers.  **General Information** | |   **Basic Table**   |  | | --- | | Customer, Product |   **Calculated Field Expression**   |  |  | | --- | --- | | Field Name | Expression | | Total Amount | Ex: Products Price+ Total Tax |   **Filter**   |  |  | | --- | --- | | Field Name | Condition | | Amount | All Taxes must be included | |

**View table with diagram for the database**

A picture containing text, white

Description automatically generated

**Implementation progressive details:**

Microsoft Access Implementation: This is a database management system (DBMS) that the company has used to store and manage data for the database management system. It is part of the Microsoft 365 application and is considered a powerful tool in productivity regarding business operations and users. It is made to store and manage large amounts of data in simple forms, making it easy to use in other applications. This has helped the department store to keep large quantities that can be used in further analysis applications. It has also made it easy to keep the store’s data safe. Its compatibility with other data analysis applications has made it easy and cheap to analyze data and store it directly in Microsoft Access without further interference with the data. This has also helped to avoid errors caused by too much data manipulation and analysis.

Features of this kind of database that have made it easy for the implementation progress of this project include.

* Queries – this is another feature Microsoft Access possesses. It is possible to find any information within the database since it supports complex queries. This feature has helped in the case of department stores, where we have many users in the database.
* Forms – Access has this feature that allows database developers to create forms that allow users to enter their information into the database without using other Microsoft applications like spreadsheets. It has made it easier for customers to feed in their details with fewer errors than when using spreadsheets.
* Reports – this is the final feature that Access has, enabling users to quickly create their custom reports. It has helped manage our database because users can quickly sort, group, and label their data for printing purposes.
* Relational database—this database has this feature that allows tables to be interrelated; for example, it has been easy because the department store can now maintain separate tables for Inventory, orders, customers, and data information.
* Tabulation – Microsoft Access stores data in tables; therefore, it is easy for the business to build a database that includes all the data for the whole business.

Visual studio application implementation: This application has been helpful so far in the implementation of the database in different ways.

* Highly esteemed – this application is the most popular choice in society. Therefore, it has saved business management from the high cost of control because it's easy to get help and information in times of trouble.
* Compatibility - Visual Studio application is highly compatible with other servers like SQL, among others; therefore, running their activities in the Department store database management system using this application has helped in the easy way of managing our customers and other users of the database.

**Preliminary field list:**

During this portion of the database design process, I have defined a preliminary table list which is used to identify and establish the tables for the new database. I have used three procedures to develop the list. The first one involves using preliminary field list, the second involves using the list of subjects which I have gathered during the interviewing process, and the third involves using mission objectives which I had defined in the previous report of the database design process. Then I built the structure of each table using fields from the preliminary field list.

**Table structure:**

During this portion of the database design process, I have created a final table list using the preliminary field list. The final table list mainly consists of the description of each preliminary field list and its type. From the table field list, I have considered all the fields and made a table structure which describes the main fields of the new database through which we can analyze our data and create a new database using Microsoft Access and Visual Studio.

Field specification sheet:

* Several field specification tables have been created to represent the database entities and attributes.
* Some of the tables created the employee table, customer table, supplier table, sales table
* The employee table contains attributes such as employee ID, first name, last name, address, and email.
* The customer table contains attributes such as Customer ID, name, address, and phone number.
* The supplier table contains attributes such as supplier ID, supplier name, product ID, quantity delivered and pending balance.
* The Sales table consists of the attributes such as sales ID, date, sales amount, and product ID
* The parking table consists of parking lot number, address of the location

**System Implementation:**

System implementation is a very crucial phase in this database project. The development of the system has not yet started but the design of the user interface is complete. The system must be able to handle the influx of data that will be stored within the database, and it must be able to provide users with access to their information. The system should also be designed in a way that makes it easy for users to update or modify their information.

**System Architecture:**

The system architecture for this database project will consist of three main components: the back-end server, the front-end server, and the database itself. The back-end server will contain all the functionality that connects to the database, and it will also be responsible for managing user access to the data. The front-end server will provide users with access to their information, and it will also be responsible for displaying the data in a user-friendly manner. Finally, the database itself will contain all the information that is stored within the system.

The following are some key aspects that must be considered when designing this system:

1) User Access: The system must be designed in such a way that users have access to their information in a timely manner. This means that the back-end server must be able to connect to the database quickly and efficiently, and it also means that the front-end server must be able to display data quickly and easily.

2) Data Storage: The system must be designed in a way that allows for efficient storage of data. This means that the back-end server must be able to store large amounts of data without becoming overloaded, and it also means that the front-end server must be able to display data quickly and easily.

3) Data Security: The system must be designed in a way that ensures data security. This means that the back-end server must be able to protect against unauthorized access, and it also means that the front-end server must be able to prevent users from tampering with data.

4) User Interface: The system must be designed in a way that is user-friendly and easy to use. This means that the front-end server must be easy to navigate and use, and it also means that the data displayed on the front-end server must be easy to understand.

5) Performance: The system must be designed in a way that ensures high performance. This means that the back-end server must be able to connect quickly and efficiently to the database, and it also means that the front-end server should not slow down as data is being displayed.

6) Scalability: The system must be designed in a way that allows for scalability. This means that the system can be expanded or upgraded easily, and it also means that the system can handle large amounts of data.

**Database Development**

The database implementation process of Asha department store business management system begins by determining the structure of the data. The structure includes identifying the entities, attributes, and relationships within the data. This involves creating a C# programming interface that allows users on the Asha department store business management system to access and manipulate data stored in a database. The first step in implementing a database management system for Asha department store business management system would be to create a database using Microsoft Access. This database would contain all the necessary tables and fields needed to store information on products, customers, orders, and employees. Once the database is created, it would need to be populated with data. This data could be entered manually or through a database query. Once the data is loaded, it would need to be organized into tables and fields in a manner that makes sense for the Asha department store business management system. Finally, the database should be configured so that it can be accessed by users on the Asha department store business management system. This could be done through a user interface or by using an Access database connection.

 Since the database is created, populated, and configured, the next step is developing a C# programming interface to it. This interface would allow users on the Asha department store business management system to access data from the database and manipulate it in various ways. The interface could also be used to create and manage orders and products. The project would need to be tested before it can be released to users. The C# will function as the programming language for the project because it is easy to use. Once the C# programming interface is developed, it would need to be tested using a lab environment. This testing would ensure that the interface works as intended and that any bugs are fixed. Once the testing is complete, the interface could be released to users of the system.

The database is vital for Asha department store business because it allows users to access data about products and orders. Without a database, the interface would not work. Since the company wants to transform their management system to be automated, the database is essential for the project. The MS Access will function as a bridge between the user interface and the database. The MS Access will allow users to access data stored in the database, while ensuring that data is correctly formatted and correctly entered by the user. This will make it easier for users to use the interface and ensure that all their data is correct. Testing is essential for any project, especially one like this which involves interacting with a database. The interface would not work if there were any bugs in the programming or in the database. Testing will ensure that all the programming is correct, and that the data stored in the database is accurate. Once testing is completed, the interface can be released to users.

There are a few challenges that were experienced when implementing a database for the Asha department store business management system. One challenge is that the system may not be able to handle many transactions. This can be a problem if the department store is constantly busy. Another challenge is that system may not be able to keep up with changes that are made to the department store. If changes are made frequently, then the database may need to be updated often. To overcome the challenge of the system handling many transactions, the database was split into multiple smaller databases. This allowed the system to handle many transactions without bogging down.  Additionally, the system is designed to automatically keep track of changes that are made to the department store. This will ensure that the database is always up to date.

**Application Progress**

The application process for the Asha department store business management system is still in progress. We are currently working on the finalizing the database design and developing the user interface. We are also working on testing and debugging the system. Since Asha department store has many users, the application is designed to be user-friendly and efficient. I am confident that the system will be successful once completed. To ensure the application is user friendly, I have designed the system to be modular. This means that different sections of the application can be modified or updated without affecting other sections. This will make it easier for us to modify and update the system as needed. I am also working on developing an online help system so that users can learn how to use the application quickly and easily. Once the system is completed, I will provide detailed information about how to use it.  The application training will be a very important aspect of the system because once users have access to the system, they will be able to manage Asha department store efficiently and effectively. Training will also help the staff to use the system quickly and easily and for them to manage the system efficiently and effectively.  With the help of training, I will be able to gather information on how the system is being used and how we can improve it. The staff will give feedback based on the training and how well they interacted with the system, and this will be essential in further development of the system.

After training is completed, the system administrator will help to manage the system by providing updates and modifications as needed. The administrator will also provide training to other staff who may need further assistance on how to use the system effectively. testing will be conducted. Once the training is successfully achieved, testing will be conducted to ensure that the system is working as designed.  If there are any issues with the system, the administrator will be able to fix them. If the problems are fixed, testing will be conducted again. This process will continue until the system is fully functioning. Testing is important in this stage because it will help to ensure that the system is accurate and meets the needs of the users. It also facilitates identifying any issues that may need to be addressed during the development phase.

The application security will be ensured using firewalls, intrusion detection systems (IDS), and access control lists (ACLs). The system will be monitored 24/7 to ensure that unauthorized users are not able to gain access to the data.  Since the department store will be serving lots of staff and customers, it is important to ensure that the system is secure. The company manager will be responsible for implementing the system and will be responsible for ensuring that all users are able to use the system effectively. The system administrator will be responsible for ensuring the system is updated regularly and any concerns or feedback are directed to the manager for further review.

Diagram

Description automatically generated

A screenshot of a computer

Description automatically generated