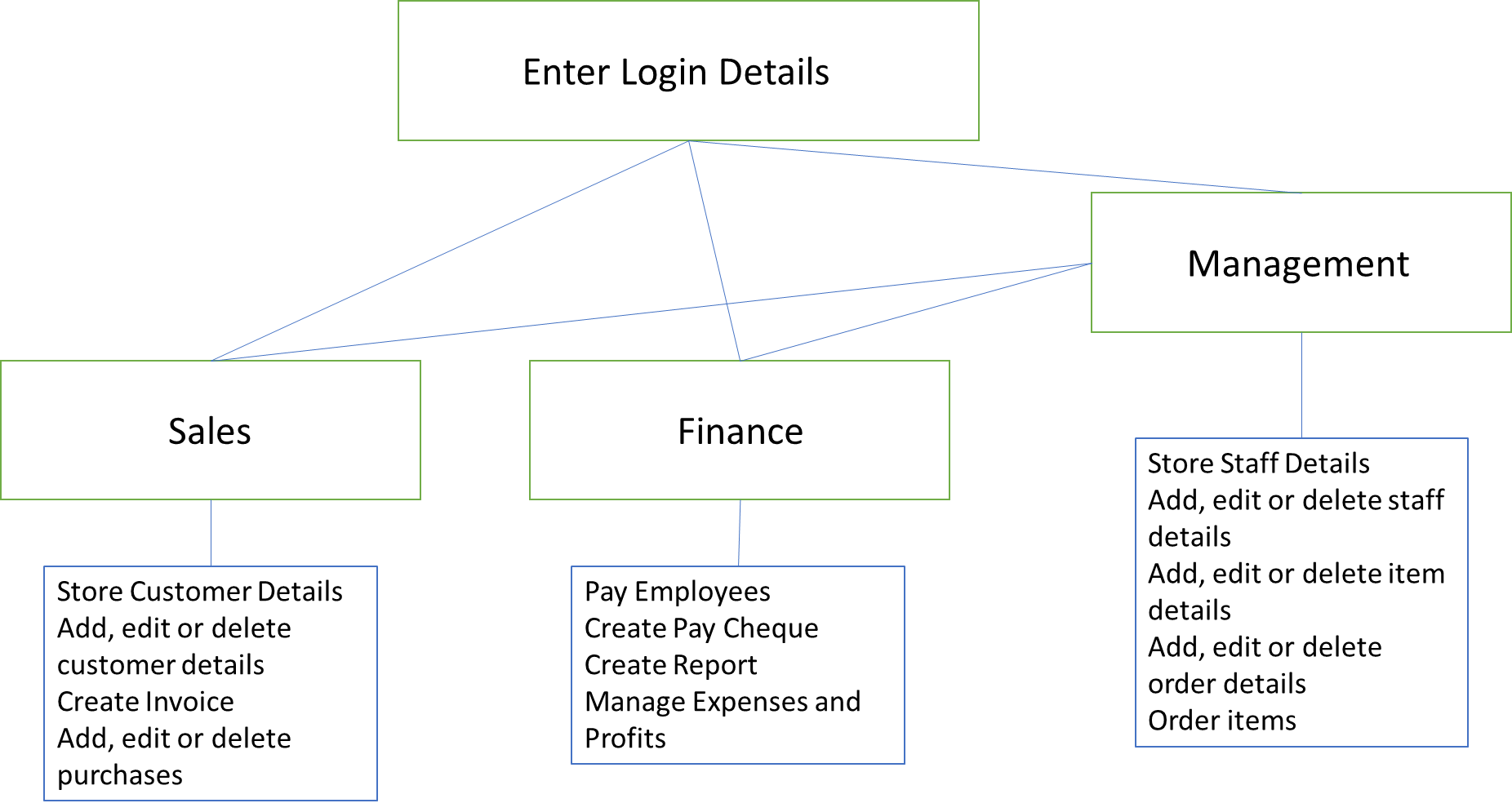
**Design**

Sitemap of the System



Data Dictionaries

**Customer Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Length** | **Validation** | **Default Value** | **Primary/Foreign Key** | **Sample Data** |
| Customer ID | Uniquely identifies the customer | Integer | 4 | Presence Check | None | Primary Key | 1000 |
| First Name | First name of the customer | String | 15 | None | None | None | “John” |
| Surname | Surname of the customer | String | 15 | None | None | None | “Smith” |
| Phone Number | Phone number of the customer allows them to be called | String | 11 | Length Check  \*11 characters | None | None | “07420696969” |
| Email | Email allows customer to be contacted | String | 50 | None | None | None | “Johnsmith@gmail.com” |
| Purchase ID | Identifies the purchase the customer has made | String | 5 | Presence Check | None | Foreign Key | “A1000” |

I have used a presence check to ensure that the primary and foreign key field does not remain empty. I have used a length check so that the phone number will be the correct length as any other phone number.

For this database, the method of access will be direct access because the customers’ data does not need to be stored in any particular order and the business will need to be able to search for the customer that they need to manage.

**Purchases Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Length** | **Validation** | **Default Value** | **Primary/Foreign Key** | **Sample Data** |
| Purchase ID | Uniquely identifies a customer purchase | String | 5 | Presence Check | None | Primary Key | “A1000” |
| Item ID | Identifies the item bought | Integer | 3 | Lookup  \*Must be from the items available  Presence Check | None | Foreign Key | 100 |
| Staff ID | Identifies the staff in charge of managing this purchase. | Integer | 2 | Presence Check | None | Foreign Key | 10 |
| Number of days | Shows the number of days that the item has been rented for | Integer | 2 | None | 7 | None | 12 |
| Purchase Date | The date the item has been purchased | Date | 10 | Format Check  \*Date Format | None | None | 01/01/2020 |
| Return Date | The date the item has to be returned | Date | 10 | Format Check  \*Date Format | None | None | 07/01/2020 |

I have used a lookup so that the items purchased will be from the items that are owned by the business and not any random item that has been typed in. I have used a format check so that the dates will be in date format.

The method of access for this database will be direct as purchases for rentable items will be made daily and people will return their items at different times so we will need to be able to access the purchase that relates to the customer.

**Staff Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Length** | **Validation** | **Default Value** | **Primary/Foreign Key** | **Sample Data** |
| Staff ID | Uniquely identifies the employee | Integer | 2 | Presence Check | None | Primary Key | 10 |
| First Name | First name of the employee | String | 15 | None | None | None | “John” |
| Surname | Surname of the employee | String | 15 | None | None | None | “Smith” |
| Phone Number | Phone number of the employee allows them to be called | String | 11 | Length Check\*11 Characters | None | None | “07420696969” |
| Email | Email allows employee to be contacted | String | 50 | None | None | None | “HopperTruth@gmail.com” |
| Bank Account Number | Allows the employee to receive a cheque | Integer | 8 | Length Check  \*8 Characters | None | None | 67234596 |
| Sort Code | Allows the employee to receive a cheque | String | 8 | Format Check  \*Sort code format | None | None | “20-69-80" |
| Hourly Rate | Shows the hourly wage of the employee | Float | 5 | None | None | None | 11.50 |
| Hours Worked | Shows the hours an employee has worked | Integer | 2 | None | None | None | 22 |
| Username | Username to login to the new system | String | 15 | None | None | None | “ho\_truth” |
| Password | Password to login into the system | String | 15 | Length Check  \*Must be greater than 8 characters | None | None | “abfg697p” |
| Access | Access rights of the employee | String | 15 | None | None | None | “Management”  “Sales”  “Finance” |

I have used a length check on the bank account number so that the number is the right length of an account number. I have used a format check on the sort code so that the sort code will have the formatting of a sort code. I have used a length check on the password so that the password can be more than eight characters long to ensure better security.

Direct access will be used for this database because staff details will be used to check the pay rate, the login details and their levels of access. Since these need to be checked whenever staff uses the system, the system will need to search for a specific record.

**Items Table**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Length** | **Validation** | **Default Value** | **Primary/Foreign Key** | **Sample Data** |
| Item ID | The ID of the item | Integer | 3 | Presence Check | None | Primary Key | 100 |
| Item Name | The name of the item | String | 30 | None | None | None | Game\_1 |
| Rentable | Says whether it can be rented or not | Boolean | 5 | Format Check  \*Must be true or false | None | None | True/False |
| Price | The price of the item | Float | 6 | None | None | None | 22.50 |
| Rental Price | The price to rent the item | Float | 6 | None | None | None | 7.50 |
| Remaining Stock | Number of items remaining | Integer | 3 | None | None | None | 34 |

I have used a format check on the rentable field because whether an item can be rented or not should be represented by true or false.

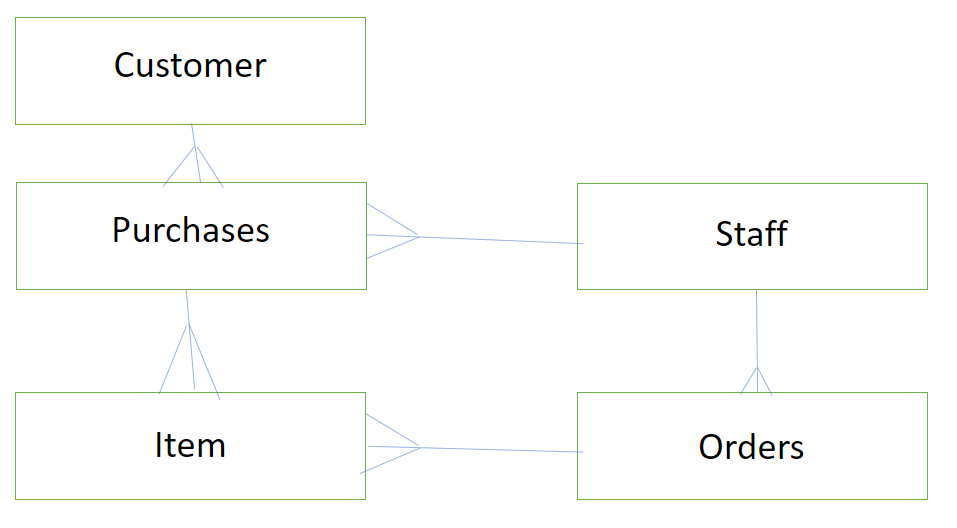
Direct access will be used for this database as some items will sell at different rates than others so the system will have to search for a specific record in this database when a purchase is being made or when the system has to notify management about low stock.

**Orders Table**

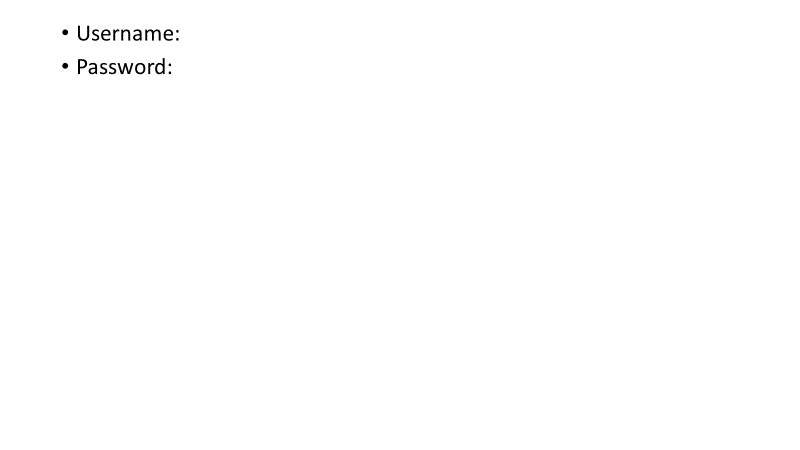
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Field Name** | **Description** | **Data Type** | **Length** | **Validation** | **Default Value** | **Primary/Foreign Key** | **Sample Data** |
| Order ID | Id of the order | String | 4 | Presence Check | None | Primary Key | “B100” |
| Staff ID | The ID of the staff in charge of the making the order. | Integer | 2 | Presence Check | None | Foreign Key | 10 |
| Item ID | Item ID from items table | Integer | 3 | Presence Check | None | Foreign Key | 100 |
| Minimum Number for delivery | The number of stock required for a delivery | Integer | 2 | None | 0 | None | 8 |
| Order Cost | Cost of the order | Float | 6 | None | None | None | £55.00 |

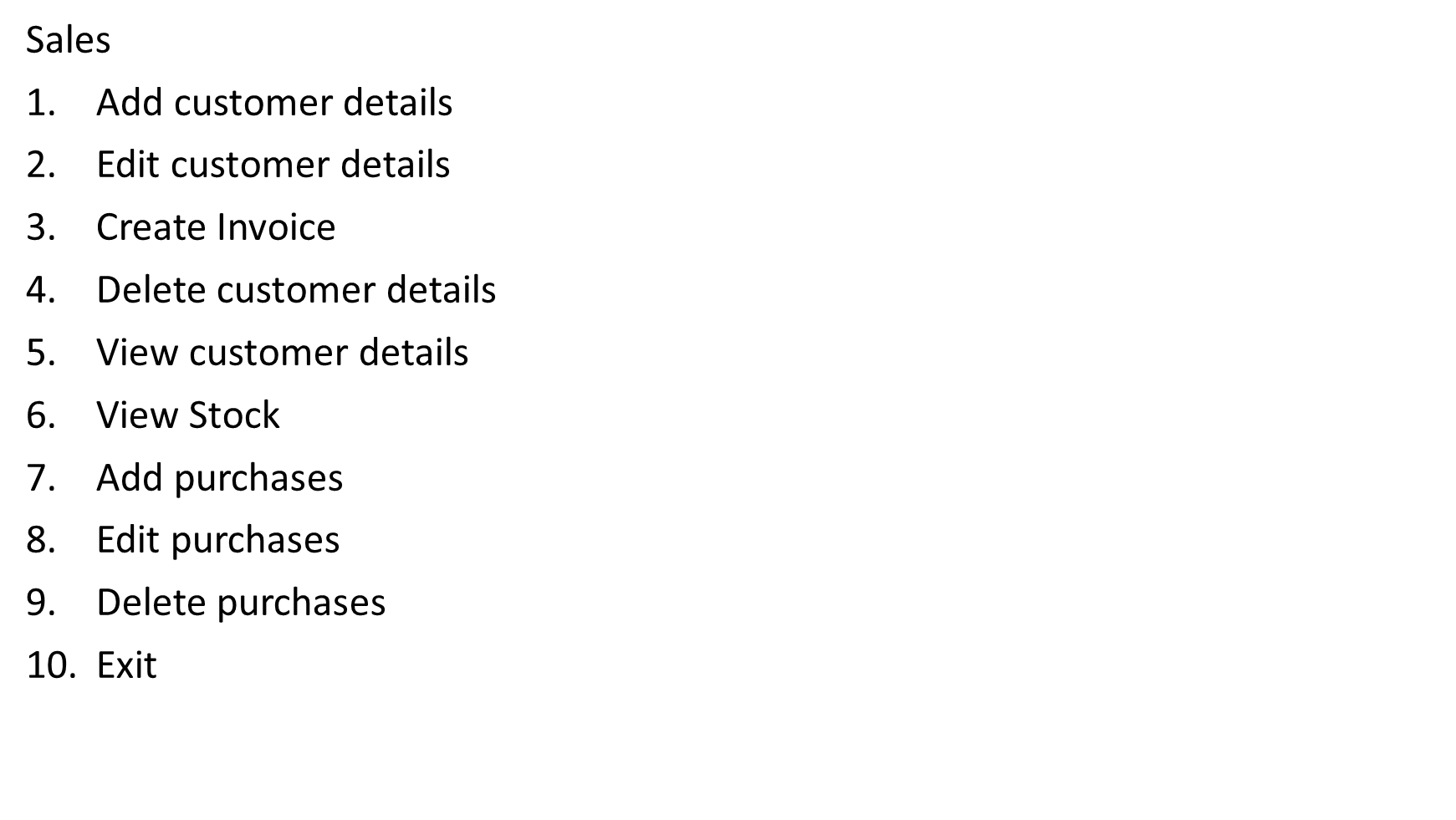
Presence checks have been used to ensure that the primary and foreign keys are not empty.

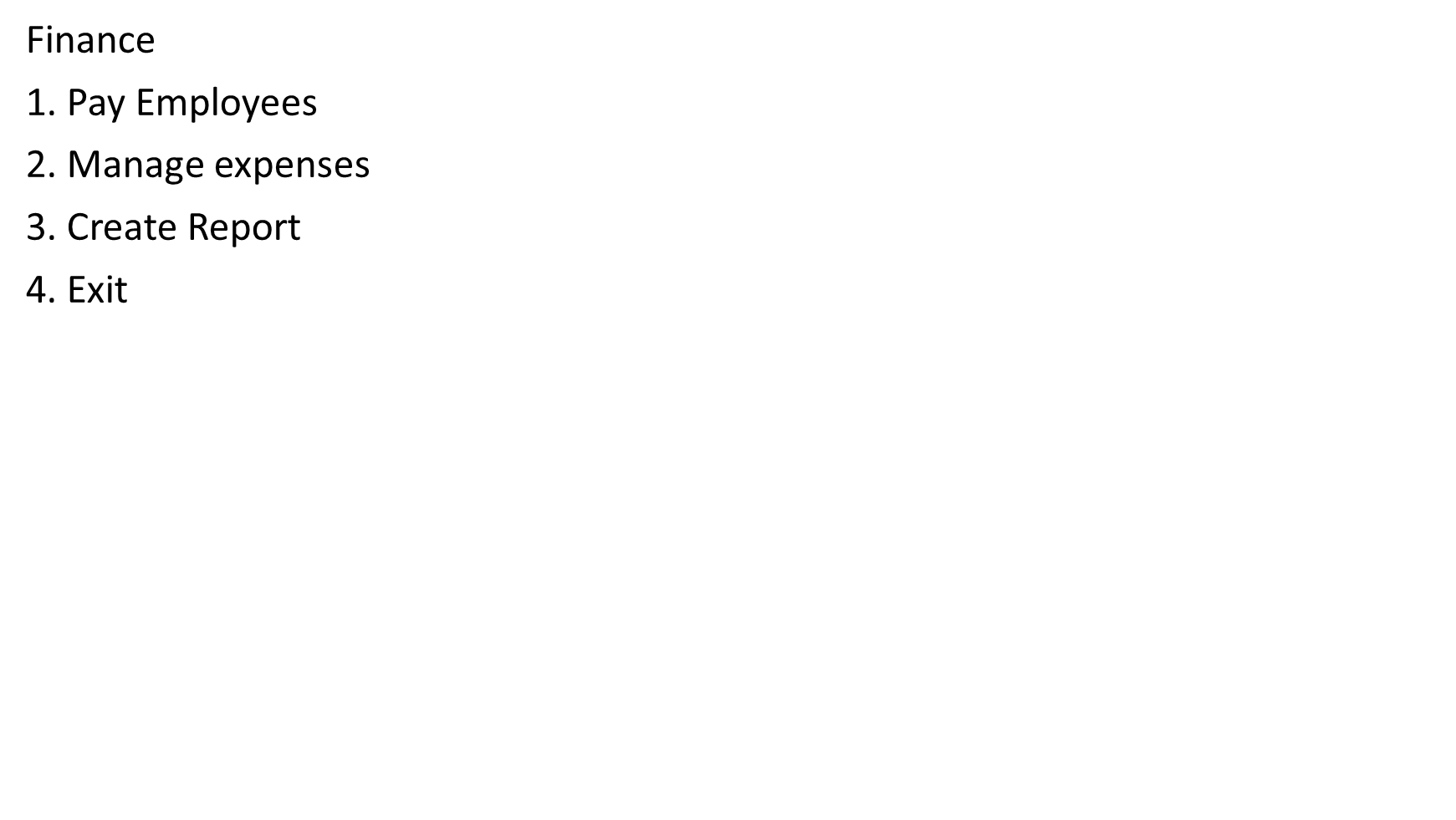
This method of access this table will be using is direct access. As stated with the items table, different items sell at different rates so therefore, more orders will be made for items that sell fast so the order information for those items will have to be searched more often than the items that sell slower.



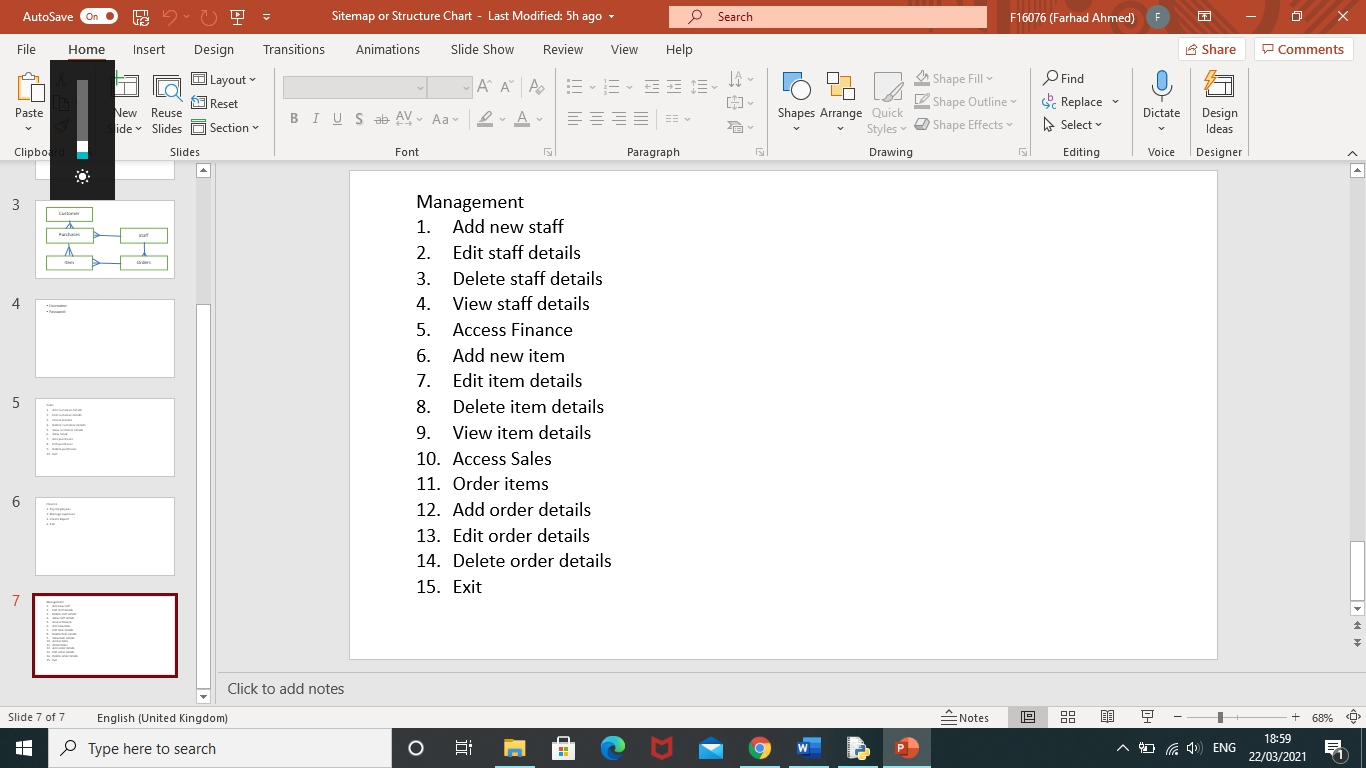
System Design



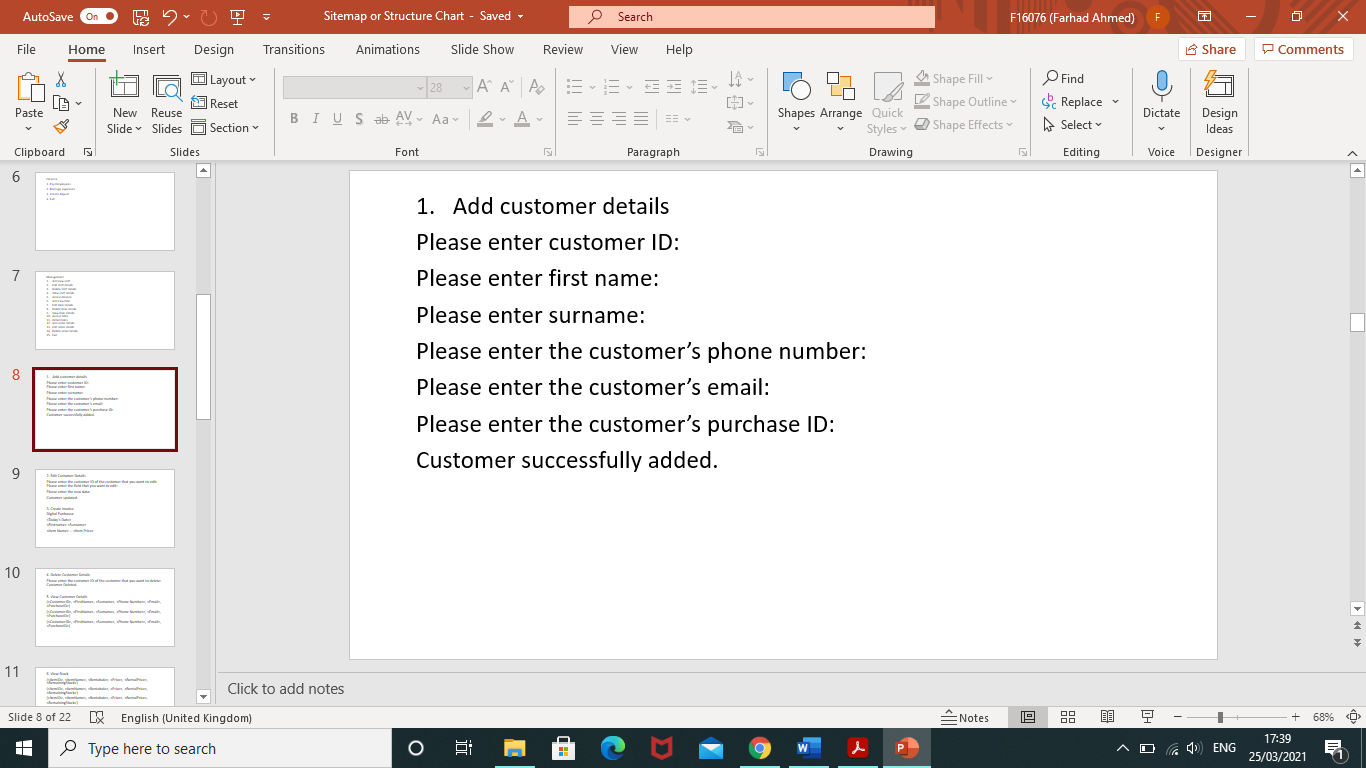
When the user uses the system, they are required to enter their username and password so that they can access the data that they need to do to do their job.

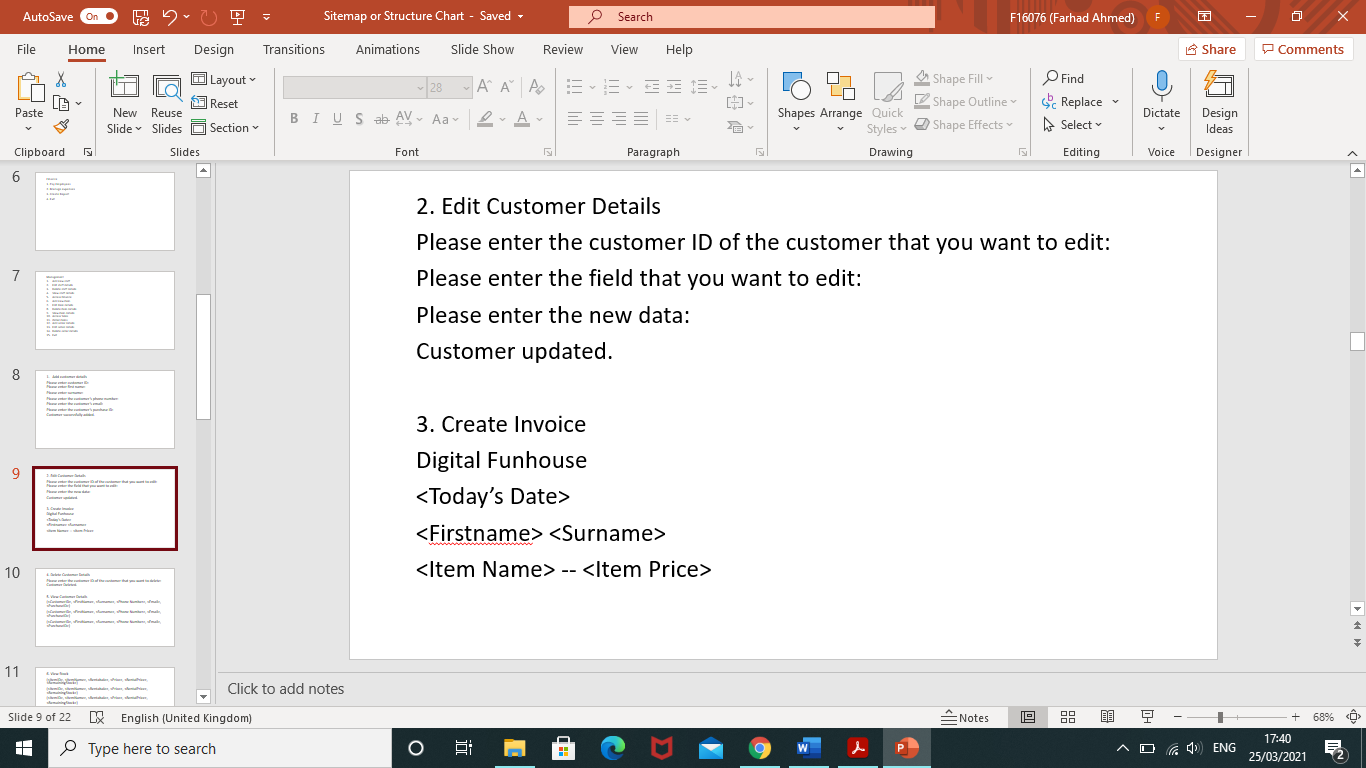
For the staff that are part of the sales team, they will have exclusive access to these functions which will allow them to manage customer details, view stock and create invoices.

For the finance staff, they will have access to these functions which will allow them to pay employees, manage their expenses and create reports.

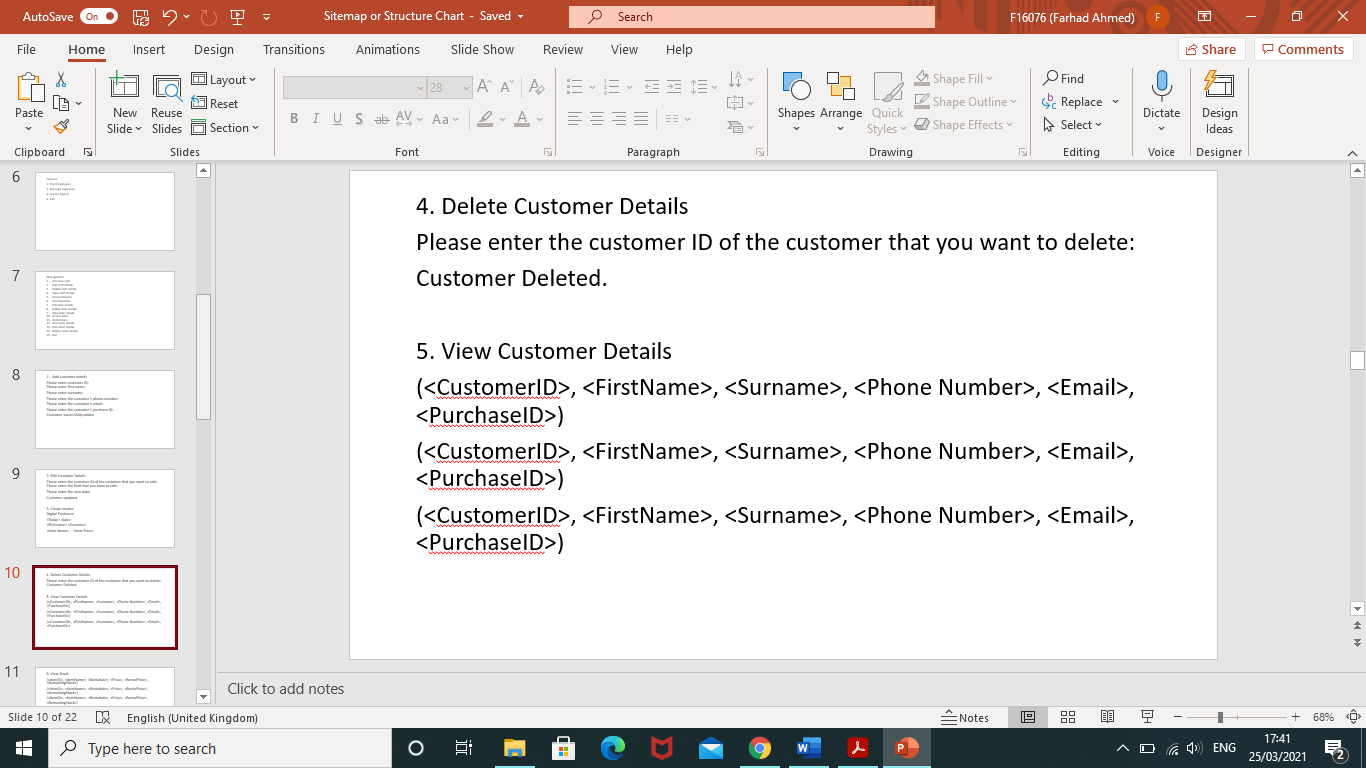


For the management staff, they will have access to these functions which allows them to manage staff details as well as accessing the functions used by the sales and finance team.

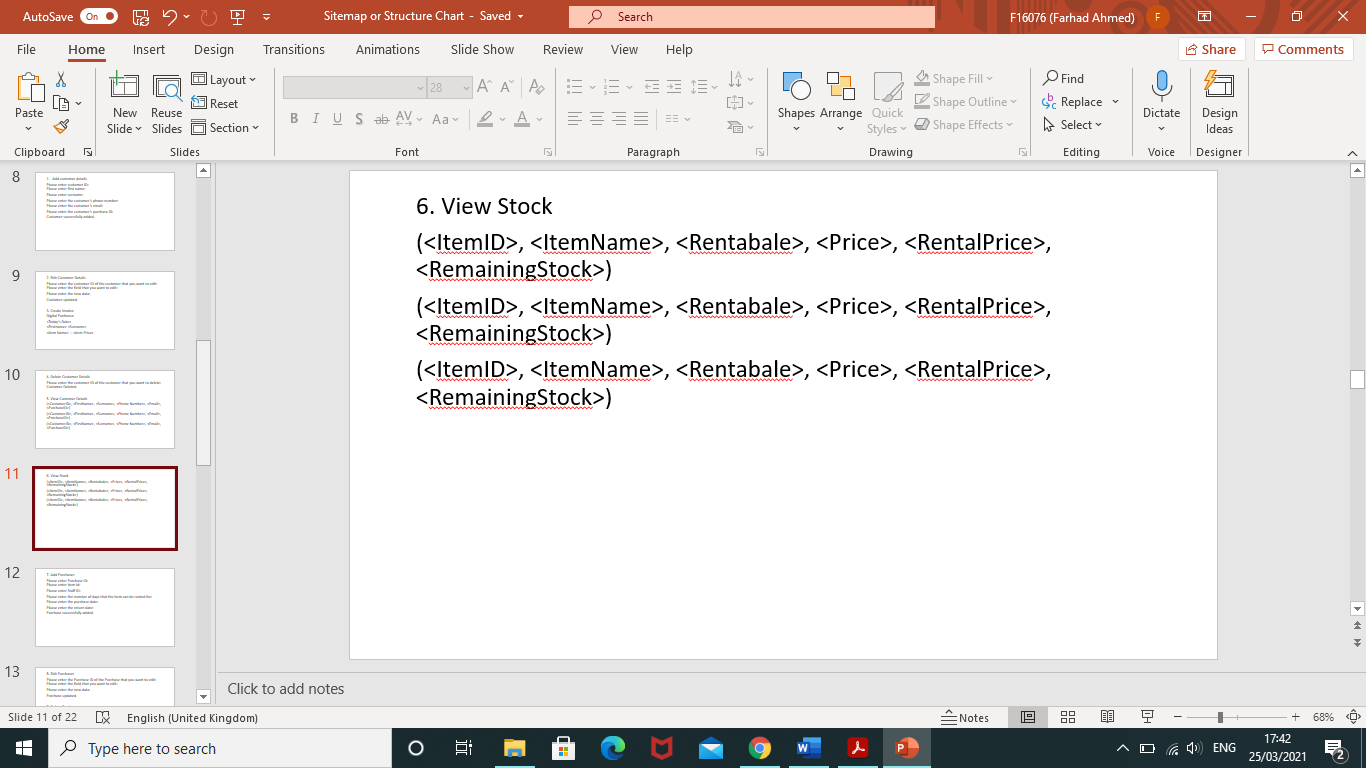




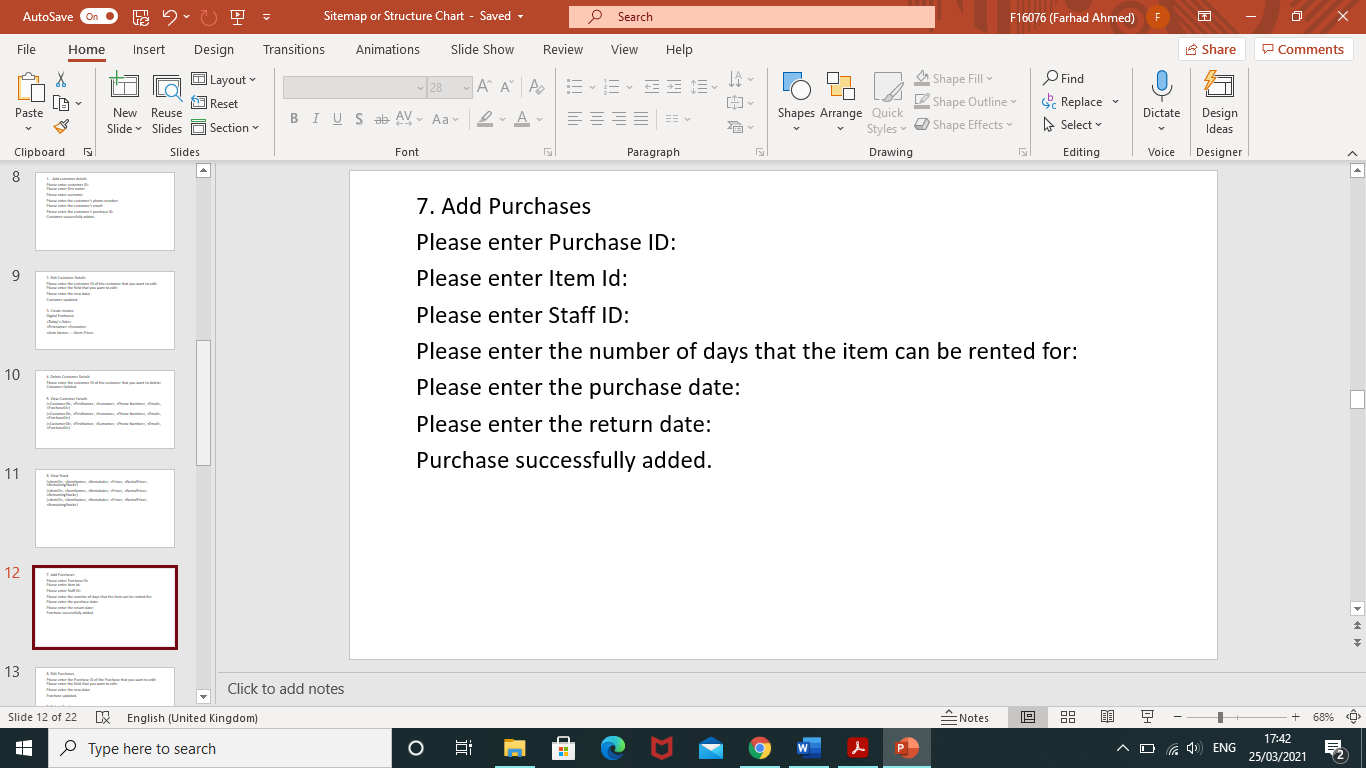
The data from the customer and items table is being used here to generate an invoice for the customer who purchased an item to rent. The invoice uses the customer’s first and last name obtained from the table and the item name and price from the item table.

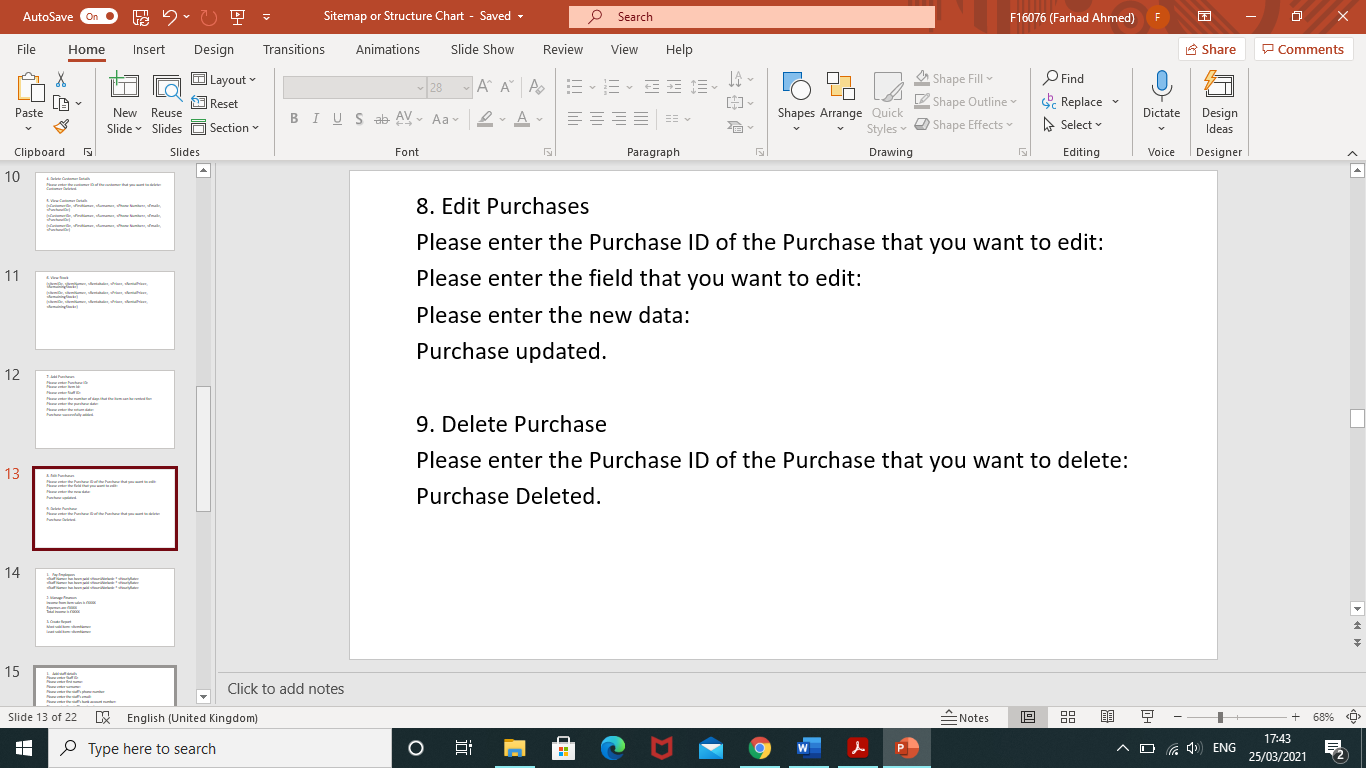


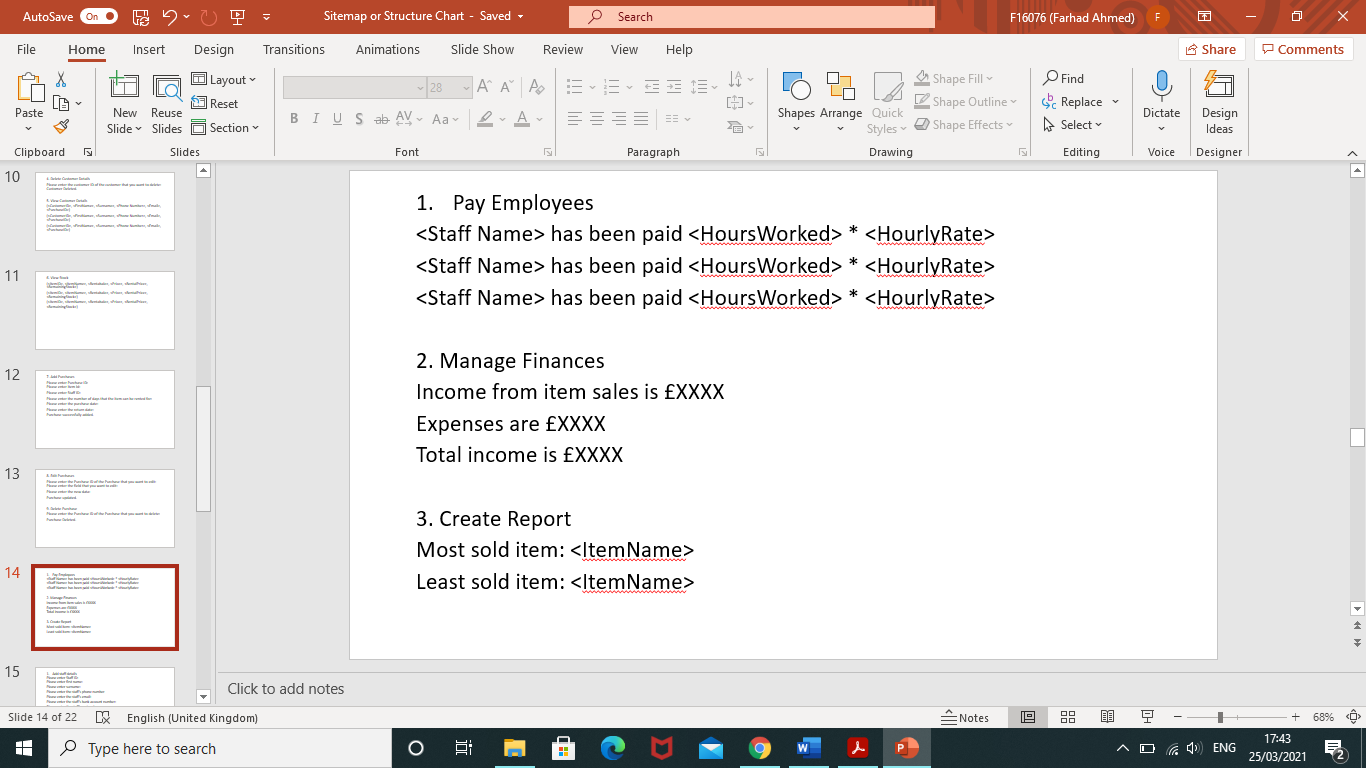
All of the data from each record and field in the customer table is being shown here so that the staff will be able to see every customer who has made a rental purchase.



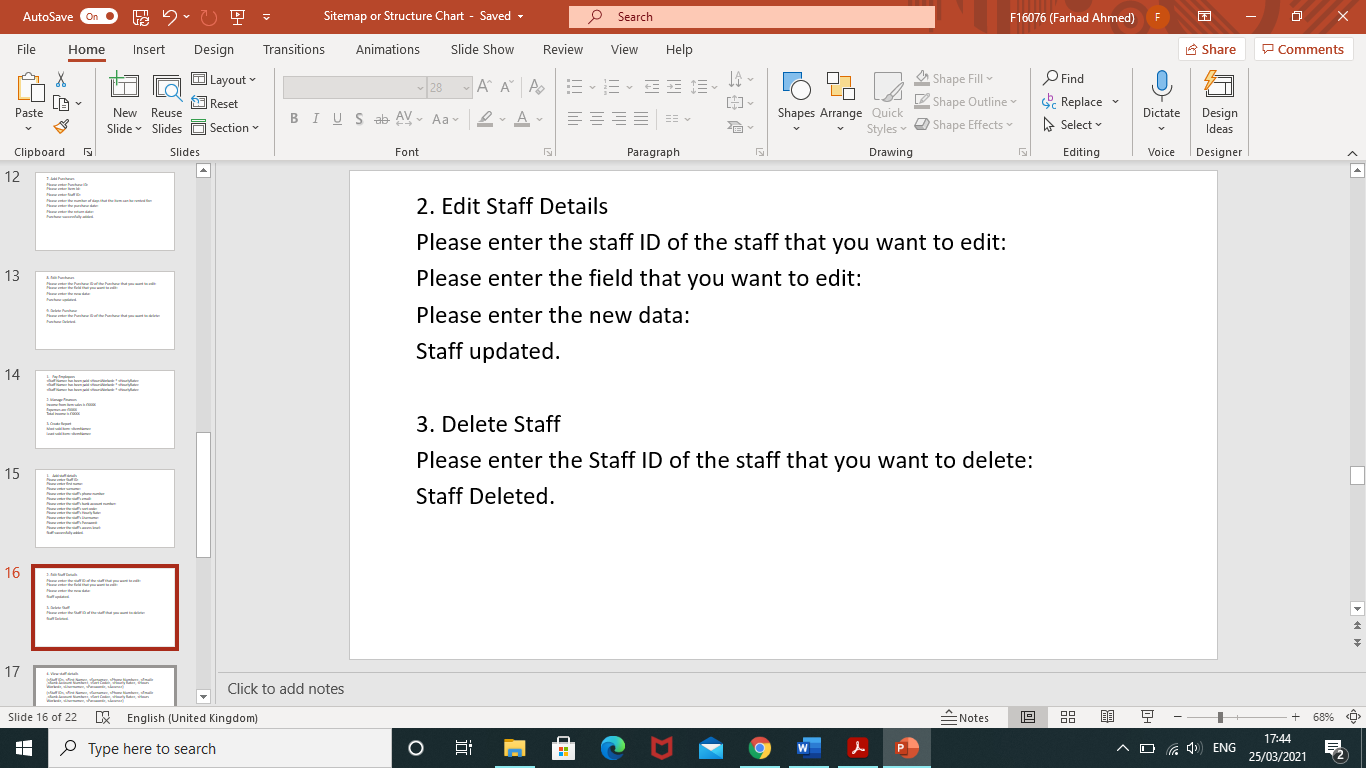
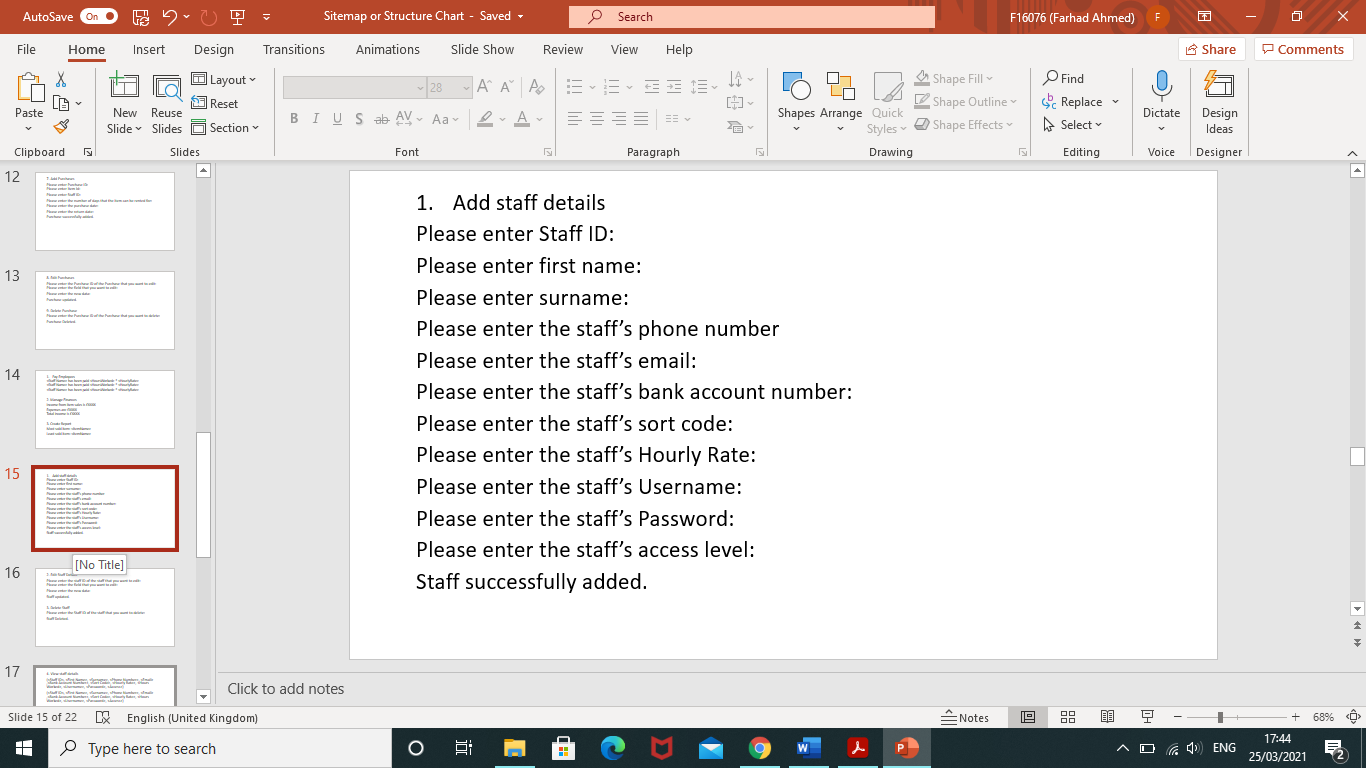
All of the data from each record and field in the items table is being shown here so that the staff will be able to see every item and how much of that item is remaining.

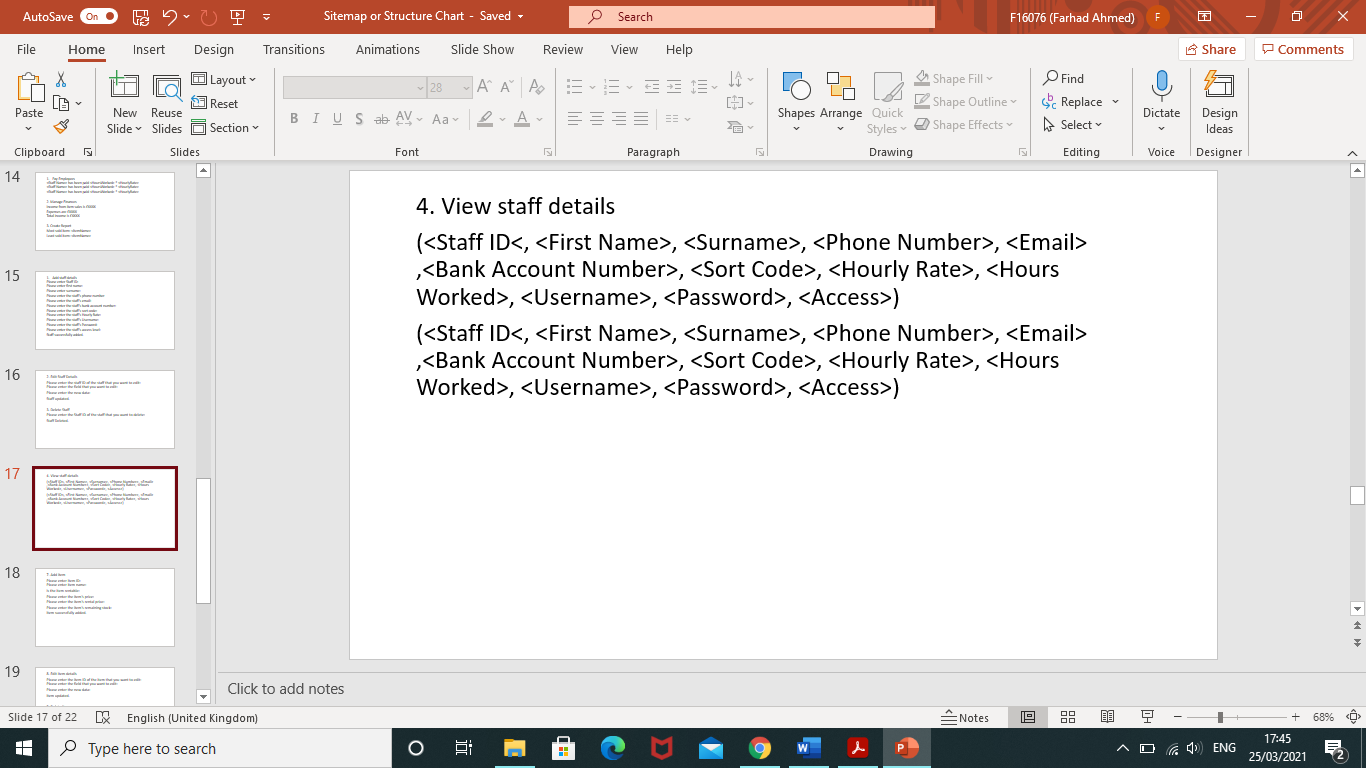




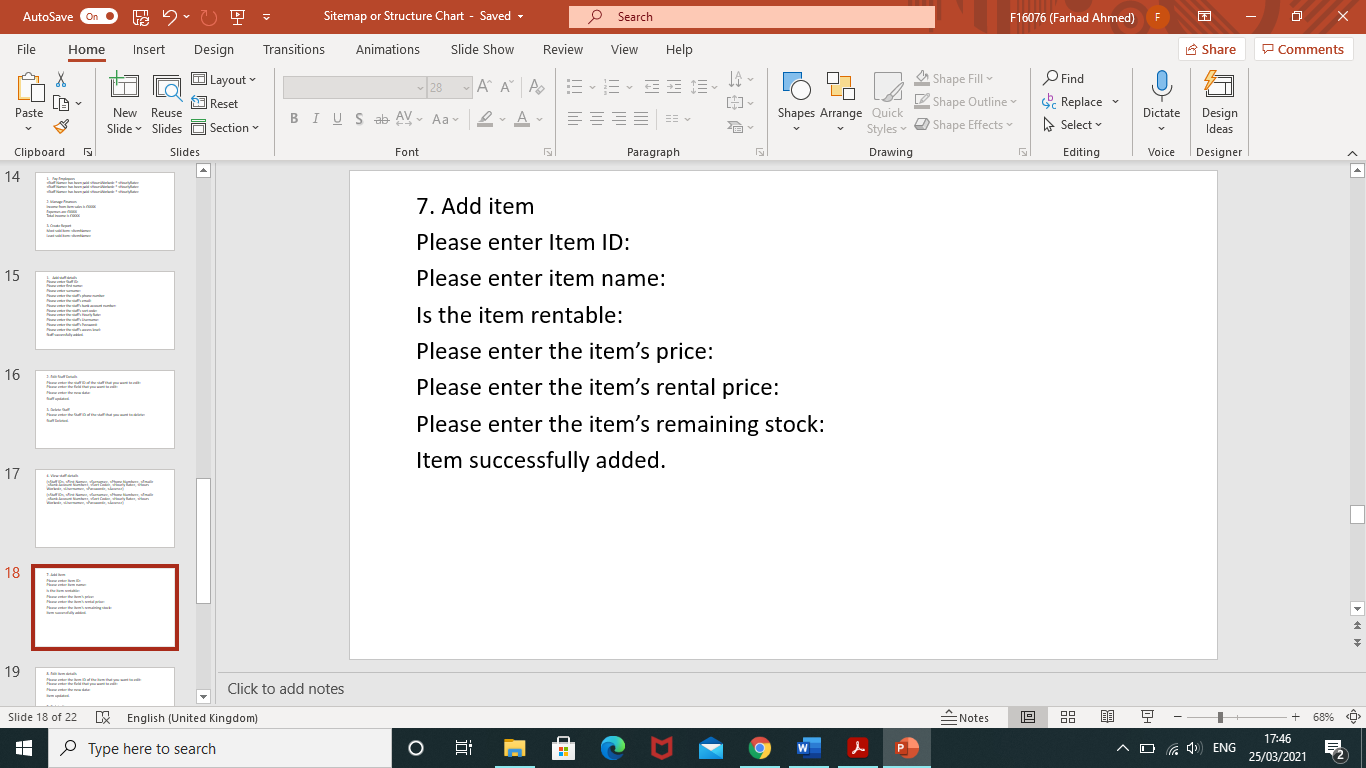


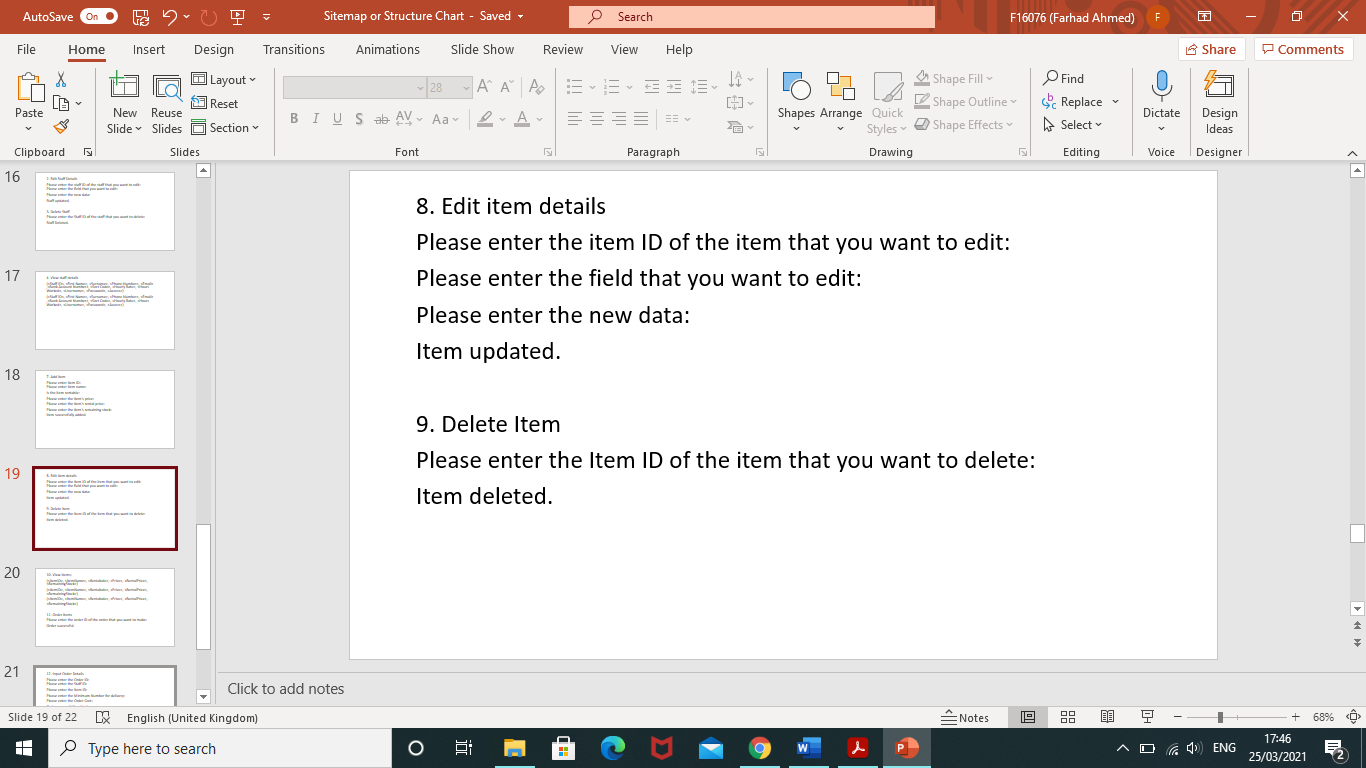
The staff name, hourly rate and hours worked from the staff table is being used here to show that each employee is being paid. Under the ‘Manage Finances’ section the system will perform calculations to determine the statistics regarding money. The item names from the items table are used to show the popularity of an item.

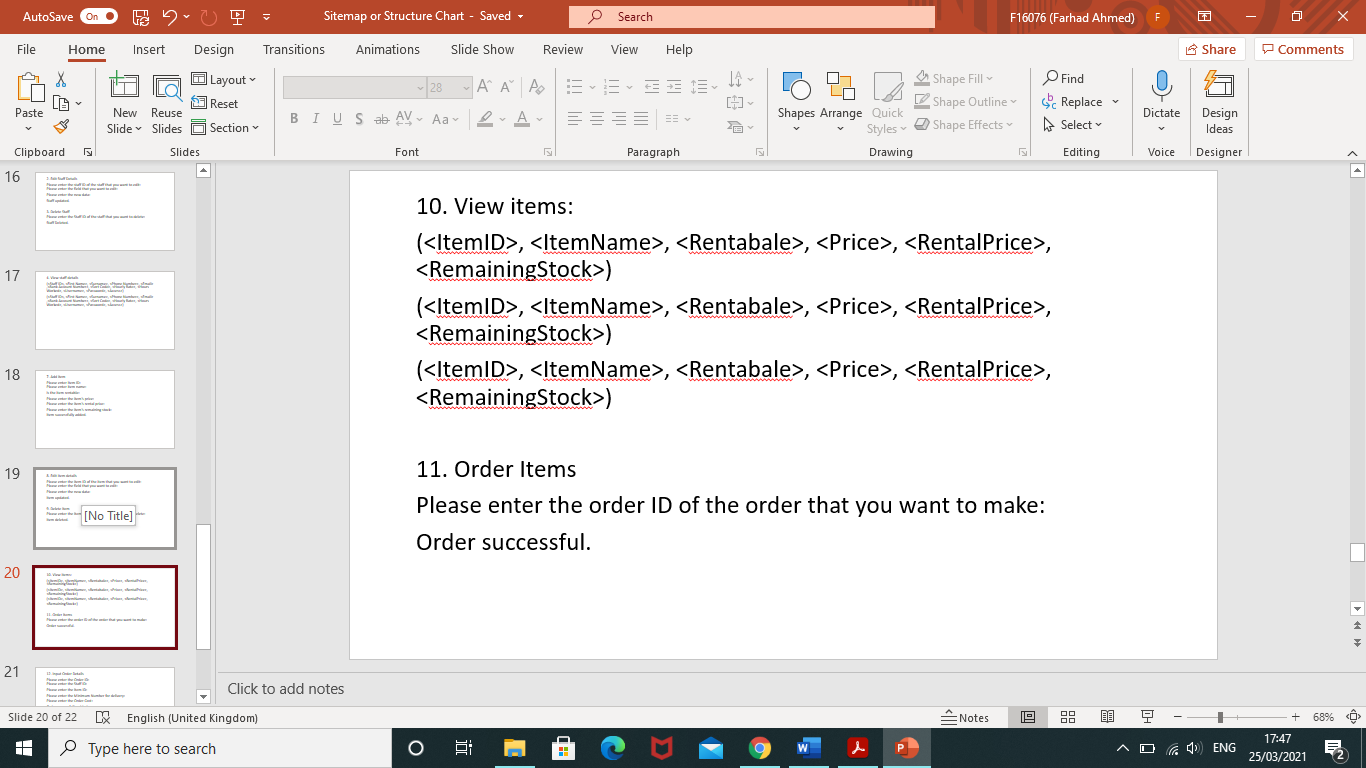




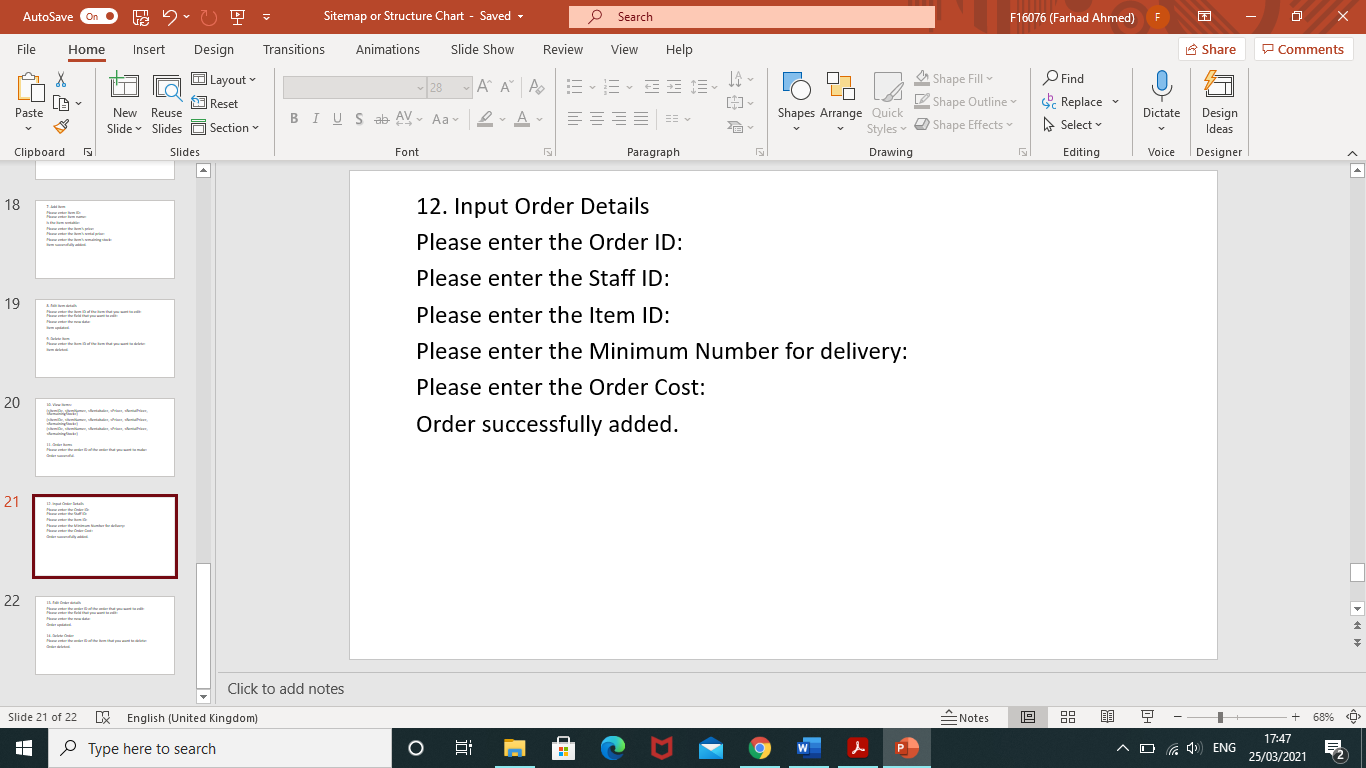
All fields and records from the staff table is being displayed here so that the management staff can have access to the staff that works for the store.

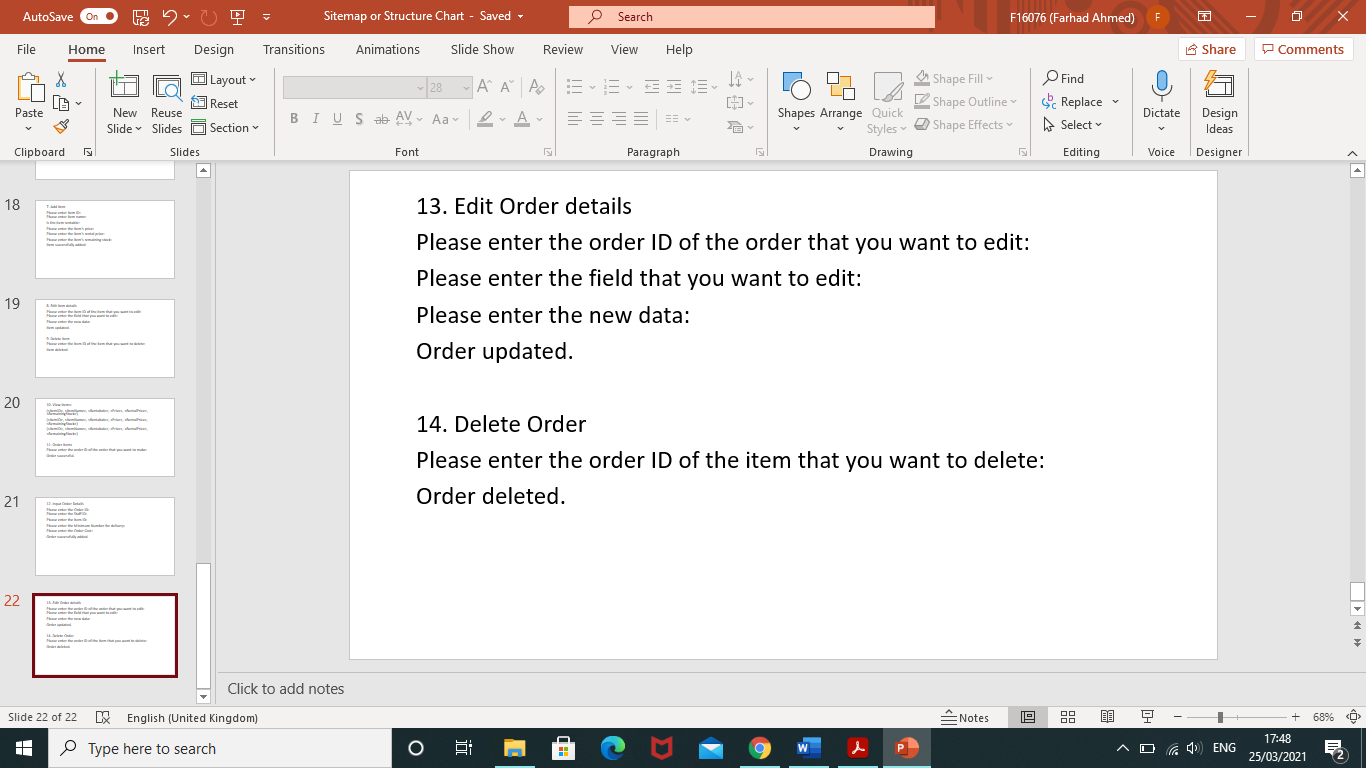






All data from the items table is being displayed here so that the management staff can see what items they own.





System Data and Processes

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **Process** |
| Customer ID | Customer | Add customer details  Edit customer details  Delete customer details |
| First Name | Customer | Add customer details  Edit customer details  Delete customer details |
| Surname | Customer | Add customer details  Edit customer details  Delete customer details |
| Phone Number | Customer | Add customer details  Edit customer details  Delete customer details |
| Email | Customer | Add customer details  Edit customer details  Delete customer details |
| Purchase ID | Customer | Add customer details  Edit customer details  Delete customer details |
| Purchase ID | Purchases | Add purchase  Edit purchase  Delete purchase |
| Item ID | Purchases | Add purchase  Edit purchase  Delete purchase |
| Staff ID | Purchases | Add purchase  Edit purchase  Delete purchase |
| Number of days | Purchases | Add purchase  Edit purchase  Delete purchase  Create Invoice |
| Purchase Date | Purchases | Add purchase  Edit purchase  Delete purchase  Create Invoice |
| Return Date | Purchases | Add purchase  Edit purchase  Delete purchase  Create Invoice |
| Staff ID | Staff | Add staff details  Edit staff details  Delete staff details |
| First Name | Staff | Add staff details  Edit staff details  Delete staff details  Pay employees |
| Surname | Staff | Add staff details  Edit staff details  Delete staff details  Pay Employees |
| Phone Number | Staff | Add staff details  Edit staff details  Delete staff details |
| Email | Staff | Add staff details  Edit staff details  Delete staff details |
| Bank Account Number | Staff | Add staff details  Edit staff details  Delete staff details  Pay employees |
| Sort Code | Staff | Add staff details  Edit staff details  Delete staff details  Pay employees |
| Hourly Rate | Staff | Add staff details  Edit staff details  Delete staff details  Pay employees |
| Hours Worked | Staff | Add staff details  Edit staff details  Delete staff details  Pay employees |
| Username | Staff | Add staff details  Edit staff details  Delete staff details |
| Password | Staff | Add staff details  Edit staff details  Delete staff details |
| Access | Staff | Add staff details  Edit staff details  Delete staff details |
| Item ID | Item | Add item details  Edit item details  Delete item details  Create report |
| Item Name | Item | Add item details  Edit item details  Delete item details  Create Invoice  Create report  Manage expenses |
| Rentable | Item | Add item details  Edit item details  Delete item details |
| Price | Item | Add item details  Edit item details  Delete item details  Create Invoice  Manage expenses  Create report |
| Rental Price | Item | Add item details  Edit item details  Delete item details  Create invoice  Manage expenses |
| Remaining Stock | Item | Add item details  Edit item details  Delete item details  Order items |
| Order ID | Orders | Add order details  Edit order details  Delete order details  Order items |
| Staff ID | Orders | Add order details  Edit order details  Delete order details  Order items |
| Item ID | Orders | Add order details  Edit order details  Delete order details  Order items |
| Minimum Number for delivery | Orders | Add order details  Edit order details  Delete order details  Order items |
| Order Cost | Orders | Add order details  Edit order details  Delete order details  Order items |

Pseudocode

Pseudocode

CREATE STAFF TABLE (

Staff ID (Primary Key)

First Name

Surname

Phone Number

Email

Bank Account Number

Sort Code

Hourly Rate

Hours Worked

Username

Password

Access)

CREATE ITEMS TABLE (

Item ID (Primary Key)

Item Name

Rentable

Price

Rental Price

Remaining Stock)

CREATE PURCHASES TABLE (

Purchase ID (Primary Key)

Item ID (Foreign Key)

Staff ID (Foreign Key)

Number of days

Purchase Date

Return Date)

CREATE CUSTOMER TABLE (

Customer ID(Primary Key)

First Name

Surname

Phone Number

Email

 Purchase ID (Foreign Key))

CREATE ORDERS TABLE (

Order ID (Primary Key)

Staff ID (Foreign Key)

Item ID (Foreign Key)

Minimum Number for delivery

Order Cost)

FUNCTION sales\_menu:

 ENTER WHILE LOOP:

    OUTPUT”

Sales

1. Add customer details
2. Edit customer details
3. Create Invoice
4. Delete customer details
5. View customer details
6. View Stock
7. Add purchases
8. Edit purchases
9. Delete purchases
10. Exit”

INPUT OPTION

IF OPTION ==1:

  INPUT CUSTOMER DETAILS: customerID, FirstName, surname, PhoneNumber, Email , purchaseID

  ADD CUSTOMER DETAILS TO CUSTOMER TABLE

END IF

IF OPTION == 2:

  SELECT CUSTOMER FROM CUSTOMER TABLE

INPUT FIELD

INPUT NEW DATA

  UPDATE CUSTOMER DETAILS INSERT NEW DATA AT FIELD

END IF

IF OPTION == 3:

  SELECT Item Name AND Item Price FROM ITEMS TABLE

SELECT Customer Name

INPUT Date

  CREATE INVOICE

‘Digital Funhouse

Date

Customer Name

Item Name -- Item Price’

END IF

IF OPTION == 4:

SELECT CUSTOMER FROM CUSTOMER TABLE

  DELETE CUSTOMER FROM CUSTOMER TABLE

END IF

IF OPTION == 5:

  SELECT ALL CUSTOMERS FROM CUSTOMER TABLE

END IF

IF OPTION ==6:

  SELECT ALL ITEMS FROM ITEMS TABLE

END IF

IF OPTION ==7:

  INPUT PURCHASE DETAILS: Purchase ID, Item ID, Staff ID, Number of days, Purchase Date, Return Date

  ADD PURCHASE DETAILS TO PURCHASES TABLE

END IF

IF OPTION == 8:

  SELECT PURCHASE FROM PURCHASES TABLE

INPUT FIELD

INPUT NEW DATA

  UPDATE PURCHASE DETAILS INSERT NEW DATA AT FIELD

END IF

IF OPTION == 9:

SELECT PURCHASE FROM PURCHASES TABLE

  DELETE PURCHASE FROM PURCHASES TABLE

END IF

IF OPTION == 10:

  EXIT WHILE LOOP

END IF

FUNCTION finance\_menu:

ENTER WHILE LOOP:

   OUTPUT “

Finance

1. Pay Employees

2. Manage expenses

3. Create Report

4. Exit”

   INPUT OPTION

   IF OPTION == 1:

     SELECT Bank Account Number, Sort Code, Hours Worked, Hourly Rate FROM STAFF TABLE

OUTPUT Hours Worked \* Hourly Rate

     PAY EMPLOYEES

 END IF

  IF OPTION == 2:

     CALCULATE ALL Hours Worked \* Hourly Rate FROM STAFF TABLE

CALCULATE COST FROM ORDERS

CALCULATE INCOME FROM ITEMS TABLE

OUTPUT INCOME – (COST FROM ORDERS AND EMPLOYEE COSTS)

END IF

   IF OPTION == 3:

     SELECT ITEMS TABLE

INPUT MOST SOLD ITEM

INPUT LEAST SOLD ITEM

     OUTPUT MOST SOLD ITEM AND LEAST SOLD ITEM

END IF

   IF OPTION == 4:

  EXIT WHILE LOOP

FUNCTION management\_menu:

ENTER WHILE LOOP:

  OUTPUT “

Management

1. Add new staff
2. Edit staff details
3. Delete staff details
4. View staff details
5. Access Finance
6. Add new item
7. Edit item details
8. Delete item details
9. View item details
10. Access Sales
11. Order items
12. Add order details
13. Edit order details
14. Delete order details
15. Exit”

INPUT OPTION

IF OPTION ==1:

  INPUT STAFF DETAILS: Staff ID, First Name, Surname, Phone Number, Email ,Bank Account Number, Sort Code, Hourly Rate, Username, Password, Access

ADD STAFF DETAILS TO STAFF TABLE

END IF

IF OPTION == 2:

  SELECT STAFF FROM STAFF TABLE

INPUT FIELD

INPUT NEW DATA

  UPDATE STAFF DETAILS INSERT NEW DATA AT FIELD

END IF

IF OPTION == 3:

SELECT STAFF FROM STAFF TABLE

  DELETE STAFF FROM STAFF TABLE

END IF

IF OPTION == 4:

  SELECT ALL STAFF FROM STAFF TABLE

END IF

IF OPTION == 5:

  ACCESS finance\_menu

END IF

IF OPTION == 6:

  ACCESS sales\_menu

END IF

IF OPTION ==7:

  INPUT ITEM DETAILS: Item ID, Item Name, Rentable, Price, Rental Price, Remaining Stock

  ADD ITEM DETAILS TO ITEM TABLE

END IF

IF OPTION == 8:

  SELECT ITEM FROM ITEM TABLE

INPUT FIELD

INPUT NEW DATA

  UPDATE ITEM DETAILS INSERT NEW DATA AT FIELD

END IF

IF OPTION == 9:

SELECT ITEM FROM ITEMS TABLE

  DELETE ITEM FROM ITEMS TABLE

END IF

IF OPTION == 10:

  SELECT ALL ITEMS FROM ITEMS TABLE

END IF

IF OPTION ==11:

  SELECT ORDER FROM ORDERS TABLE

INPUT Order ID

  ORDER ITEMS

UPDATE RemainingStock IN ITEMS TABLE

END IF

IF OPTION == 12:

  INPUT ORDER DETAILS: Order ID, Staff ID, Item ID, Minimum Number for delivery, Order Cost

  ADD ORDER DETAILS TO ORDERS TABLE

END IF

IF OPTION == 13:

  SELECT ORDER FROM ORDERS TABLE

INPUT FIELD

INPUT NEW DATA

  UPDATE ORDER DETAILS INSERT NEW DATA AT FIELD

END IF

IF OPTION == 14:

SELECT ORDER FROM ORDERS TABLE

  DELETE ORDER FROM ORDERS TABLE

END IF

IF OPTION == 15:

  EXIT WHILE LOOP

ENTER WHILE LOOP:

     INPUT USERNAME AND PASSWORD

     IF USERNAME AND PASSWORD IN STAFF TABLE:

            OUTPUT “Login Successful”

            LOGIN = TRUE

      ELSE:

            OUTPUT “Incorrect username or password”

END IF

      IF LOGIN == TRUE:

            SELECT Access FROM STAFF TABLE

            IF Access == Sales:

                   ACCESS sales\_menu

            ELSE IF Access == Finance:

                   ACCESS finance\_menu

            ELSE IF Access == Management:

                   ACCESS management\_menu

END IF

END IF