A2 UNIT 5

3. Design

Ryan Beattie
Candidate Number: 8002
Centre Number: 71741

Following my discussion with the stakeholders of Lee Opticians and the investigation carried out into their current system, problems and sub-problems have been identified and the design of the new system can be carried out.

Tables will be linked using foreign keys, each table containing a primary key. Validation methods such as type checks, presence checks, will be implemented into the system to ensure the data input to the system is reasonable.

Identification of Problems and Sub Problems

The system will be split into modules. This will bring advantages such as allowing the re-use of code and being able to easily correct programming errors, shortening the time required for development. Modular programming also improves the manageability of the program, making future maintenance much easier.

Tables will be linked using foreign keys, each table containing a primary key. Validation methods such as type checks, presence checks, will be implemented into the system to ensure the data input to the system is reasonable.

Login Screen

The login screen will ensure that data stored on the database system will be kept secure. Access levels will also be implemented into the system, ensuring only those permitted are able to access sensitive data. The login screen will be a simple window with three entry boxes for the user's StaffID, password and access level. A button will be included on the screen to run the login function, and a logo will be displayed at the top of the screen. When the user inputs their password, characters will be replaced with an asterisk to prevent their password from being seen and risking the security of the system.

The investigation previously carried out into Lee Optician's current system heavily emphasises the lack of security surrounding customer data, therefore, the use of a login system should eradicate this issue. Login details will be stored in a table in the database file and inputs in the login screen will be checked against this file to ensure the user is authorised to view data.

Menu Screen

If the user logs into the database system successfully, the main menu screen will be displayed. From here, they will be able to access each of the different forms to add, edit, view, and delete data in the database, provided they have the correct access level to do so. If a user is not authorised to view a specific piece of data, an error message will be displayed.

Another issue highlighted following the investigation was their current system being unorganised. The main menu clearly displays each of the different modules, making the system more user friendly and easy to navigate.

Customer Form

The customer form is used to add, update, view, and delete customer records in the database system. The user can input the CustomerID to search for a customer record and have the fields displayed in a popup window.

The investigation carried out into Lee Opticians' current system identified the problem of finding customer information. 33.3% of survey respondents said that they disagree that finding customer details is an easy task, with 13.3% saying they strongly disagree.

Appointments Form

The appointments form is used to add, update, view, and delete appointments made by customers. The user can input the AppointmentID to search for an appointment record and have the fields displayed in a popup window.

The investigation carried out into Lee Opticians' current system identified the problem of changing or cancelling appointments being a somewhat difficult task. 46.7% of survey respondents said that they disagree that changing or cancelling appointments is an easy task.

Order Form

The order form is used to add, update, view, and delete order records made for products e.g., frames. The user can input the OrderID to search for an order record and have the fields displayed in a popup window. Order details can be written to a text file to generate an order invoice, where it can be converted to a PDF or printed.

The investigation carried out into Lee Opticians' current system identified the problems surrounding the storage of order invoices, such as expenses because of storage and paper, and difficulty in searching for order invoices. The use of a digitised form should eliminate these issues.

Prescriptions Form

The prescriptions form is used to create, update, view, and delete prescriptions for customers. The user can input the PrescriptionID to search for a prescription and have the details displayed in a popup window. The prescription details can be written to a text file, where it can be converted to a PDF or printed for customers.

The investigation carried out into Lee Opticians' current system identified problems surrounding the storage of prescriptions, such as expenses because of storage and paper, and difficulty in searching for prescriptions. The use of a digitised form should eliminate these issues.

Products Form

The products form is used to add, updated, view, and delete product records. The user can input the ProductID to search for product details e.g., price, description, and have these details displayed in a popup window.

The investigation carried out into Lee Opticians' current system identified the problem of searching for product details. 46.7% of survey respondents said they strongly disagree that searching for

product details is an easy task, while 40% said they disagreed. The use of a digitised form should eliminate this issue.

Supplier Form

The supplier form is used to add, update, view, and delete supplier records in the database system. The user can input the SupplierID to search for a supplier record and have the fields displayed in a popup window.

The investigation carried out into Lee Opticians' current system identified the problem of finding supplier information or other records. The use of a digitised form should eliminate this issue.

Branches Form

The branches form is used to search branch records in the database system. The user can input the BranchID to retrieve the details of the branch including the address and the contact details. There is no need to include a feature to add, update, or delete records, as it is unlikely that these details will change. If they do require an update, I will be able to do so myself using a DB browser.

There was no requirement to be able to search for branch details highlighted in the investigation, however, I felt it would be a useful feature to improve communication between branches.

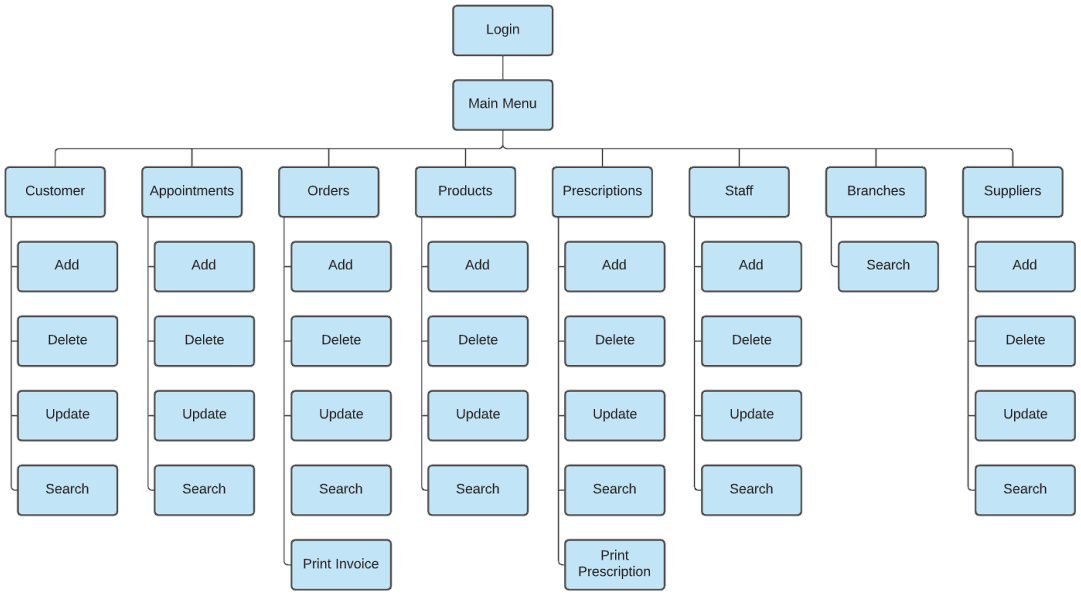
Staff Form

The staff form is used by the manager to add, update, view, and delete staff records. The manager can input the StaffID to retrieve the details of staff members if required e.g., their contact details. Any changes made to the StaffID or password using the staff form are automatically represented in the login table, where details required to log in to the system are stored.

The investigation carried out into Lee Opticians' current system identified the problem of finding staff information or other records. The use of a digitised form should eliminate this issue.

Menu Diagram

Below is a diagram displaying the layout of the new system:



Data Dictionaries

Customer Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
CustomerID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
BranchID	0	INTEGER	-	Foreign key to link to the branches table.	None required.
Name	{name}	VARCHAR	20	Customer's first name.	Length check, presence check, type check.
Surname	{surname}	VARCHAR	30	Customer's surname.	Length check, presence check, type check.
DateOfBirth	{YYYY-MM-DD}	DATE	10	Customer's date of birth.	Length check, presence check, type check.
Town	{town}	VARCHAR	30	Customer's town of residence.	Length check, presence check, type check.
Postcode	{postcode}	VARCHAR	7	Customer's postcode for their address.	Length check, presence check, type check.
EmailAddress	{emailaddress}	VARCHAR	30	Customer's email address.	Length check, presence check, type check.
TelephoneNo	0000000000000000	VARCHAR	15	Customer's telephone number.	Length check, presence check, type check.

MedicalConditions	{medicalconditions}	TEXT	-	Details about customer's medical conditions (if any).	None required.

Appointments Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
AppointmentID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
CustomerID	0	INTEGER	-	Foreign key to link to the customer table.	None required.
AppointmentDate	{YYYY-MM-DD}	DATE	10	Date the appointment is booked for.	Length check, presence check, type check.
AppointmentTime	{HH:MM}	TEXT	8	Time the appointment is booked for.	Length check, presence check, type check.
StaffID	0	INTEGER	-	Foreign key to link to the staff table.	None required.

Branches Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
BranchID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
Town	{town}	VARCHAR	30	Town the branch is located.	Length check, presence check, type check.
Postcode	{postcode}	VARCHAR	7	Postcode of the branch.	Length check, presence check, type check.
Email	{email}	VARCHAR	30	Email address to contact the branch.	Length check, type check, presence check.
Telephone	0000000000000000	VARCHAR	15	Telephone number to contact the branch.	Length check, presence check, type check.

Orders Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
OrderID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
OrderDate	{YYYY-MM-DD}	DATE	10	Date the order is made.	Length check, type check, presence check.
BranchID	0	INTEGER	-	Foreign key to link to the branches table.	None required.
SupplierID	0	INTEGER	-	Foreign key to link to the supplier table.	None required.
ProductID	0	INTEGER	-	Foreign key to link to the product table.	None required.
Quantity		INTEGER	-	Quantity of the product being ordered.	Type check, presence check.
OrderTotal	0.00	REAL	-	Total cost of the order being made in GBP.	Type check, presence check.

Prescriptions Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
PrescriptionID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
PrescriptionDate	{YYYY-MM-DD}	DATE	10	Date the prescription was created.	Length check, presence check, type check.
CustomerID	0	INTEGER	-	Foreign key to link to customer table.	None required.
PrescriptionDetails	{prescriptiondetails}	TEXT	-	Details of the prescription.	None required.
StaffID	0	INTEGER	-	Foreign key to link to staff table.	None required.

Products Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
ProductID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
SupplierID	0	INTEGER	-	Foreign key to link to the supplier table.	None required.
ProductName	{productname}	VARCHAR	30	Name of the product.	Length check, presence check, type check.
ProductDescription	{productdescription}	TEXT	-	Description of the product e.g., product features, brand.	None required.
Price	0.00	REAL	-	Price of the product in GBP.	Type check, presence check.

Staff Table

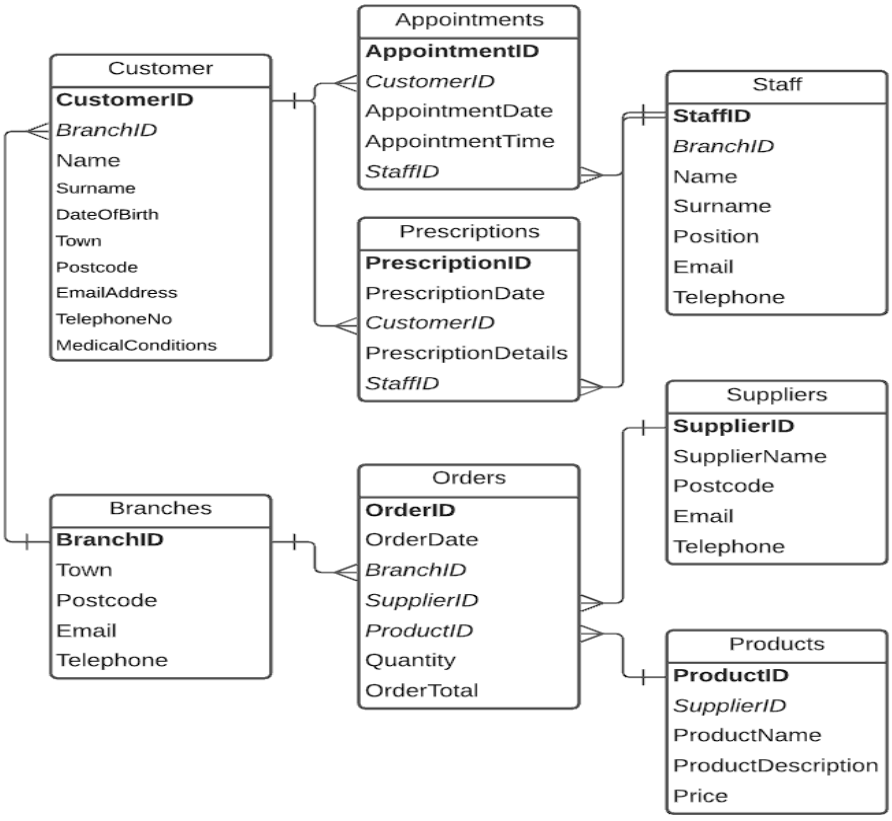
Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
StaffID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
BranchID	0	INTEGER	-	Foreign key to link to the branches table.	None required.
Name	{name}	VARCHAR	20	Staff member's name.	Length check, presence check, type check.
Surname	{surname}	VARCHAR	30	Staff member's surname.	Length check, presence check, type check.
Position	{position}	VARCHAR	15	Staff member's job position within the business.	Length check, presence check, type check.
Email	{email}	VARCHAR	30	Staff member's email address	Length check, presence check, type check.
Telephone	000000000000000	VARCHAR	15	Staff member's telephone number.	Length check, presence check, type check.

Suppliers Table

Field Name	*Default Value*	*Data Type*	*Field Length*	*Description*	*Validation*
SupplierID	0	INTEGER	-	Primary key to uniquely identify a record.	None required.
SupplierName	{suppliername}	VARCHAR	40	Name of the supplier.	Length check, presence check, type check.
Postcode	{postcode}	VARCHAR	8	Postcode of the supplier.	Length check, presence check, type check.
Email	{email}	VARCHAR	30	Supplier's email address.	Length check, presence check, type check.
Telephone	000000000000000	VARCHAR	15	Supplier's telephone number.	Length check, presence check, type check.

Entity-Relationship Diagram (ERD)

Below is an ERD highlighting the relationships between the tables in the database:



Normalisation

UNF

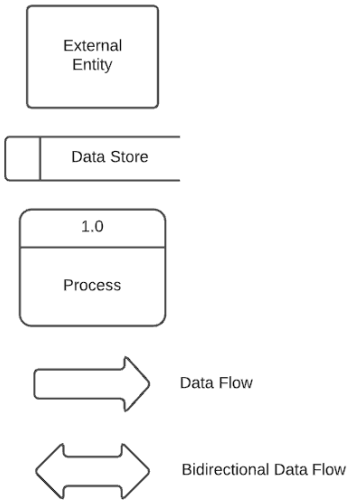
CustomerID, CustomerName, CustomerSurname, CustomerDOB, CustomerTown, CustomerPostcode, CustomerEmail, CustomerTelephone, MedicalConditions, AppointmentID, AppointmentDate, AppointmentTime, PrescriptionID, PrescriptionDate, PrescriptionDetails, StaffID, StaffName, StaffSurname, Position, StaffEmail, StaffTelephone, BranchID, BranchTown, BranchPostcode, BranchEmail, BranchTelephone, OrderID, OrderDate, Quantity, OrderTotal, SupplierID, SupplierName, SupplierPostcode, SupplierEmail, SupplierTelephone, ProductID, ProductName, ProductDescription, Price

1st Normal Form (1NF)

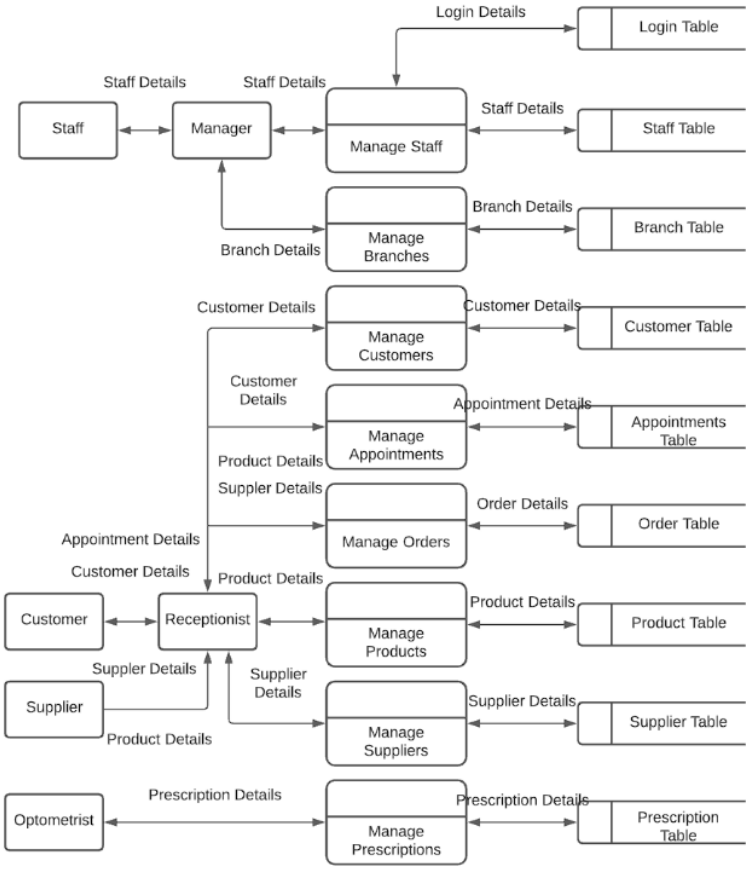
2nd Normal Form (2NF)

3rd Normal Form (3NF)

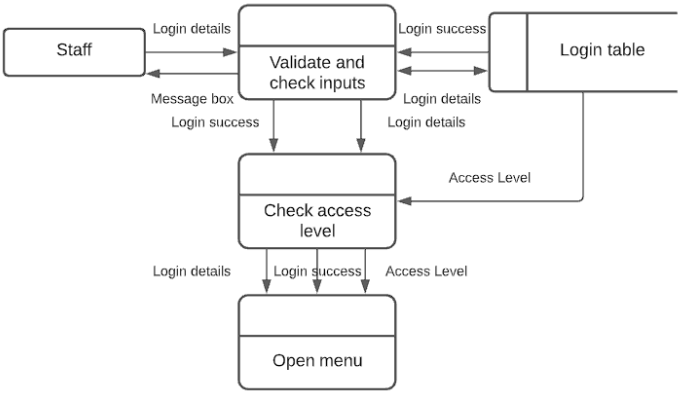
Data Flow Diagrams



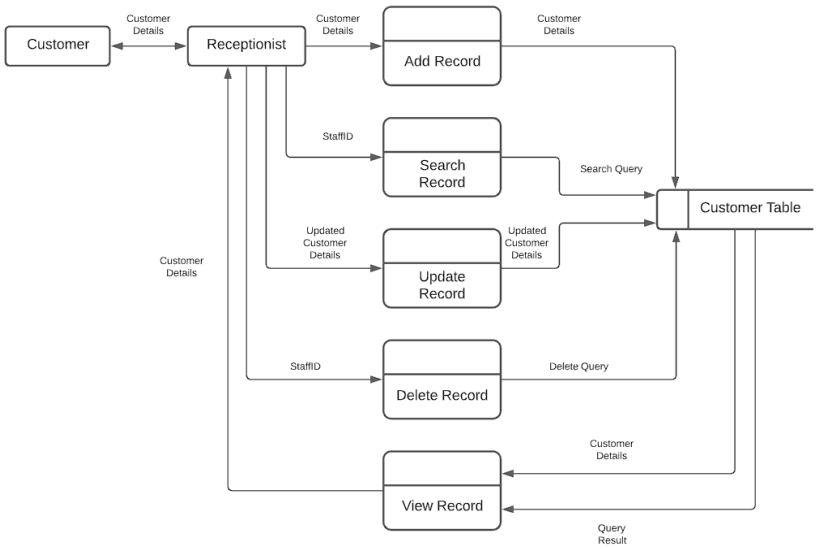
Level 0 DFD for the entire system



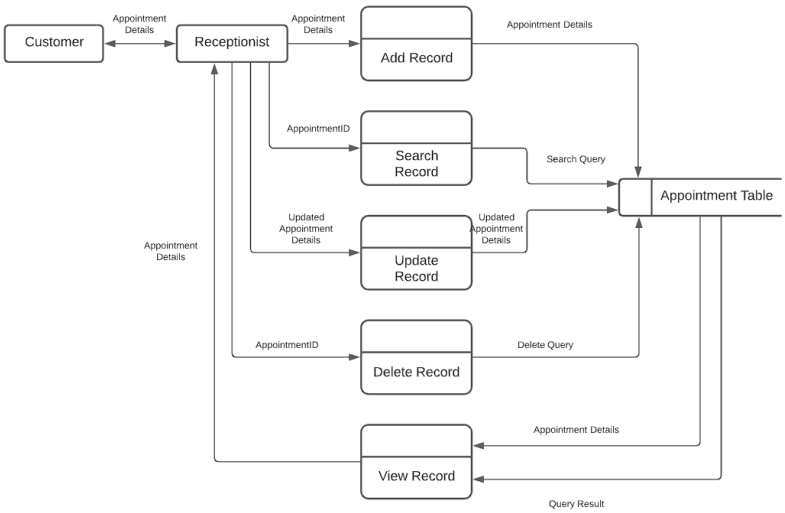
Level 1 DFD for login screen



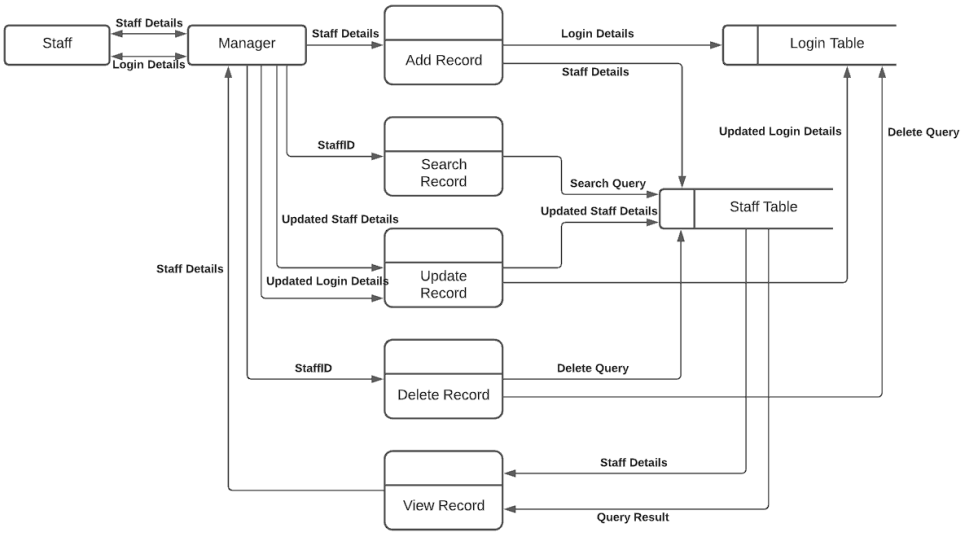
Level 2 DFD for customers



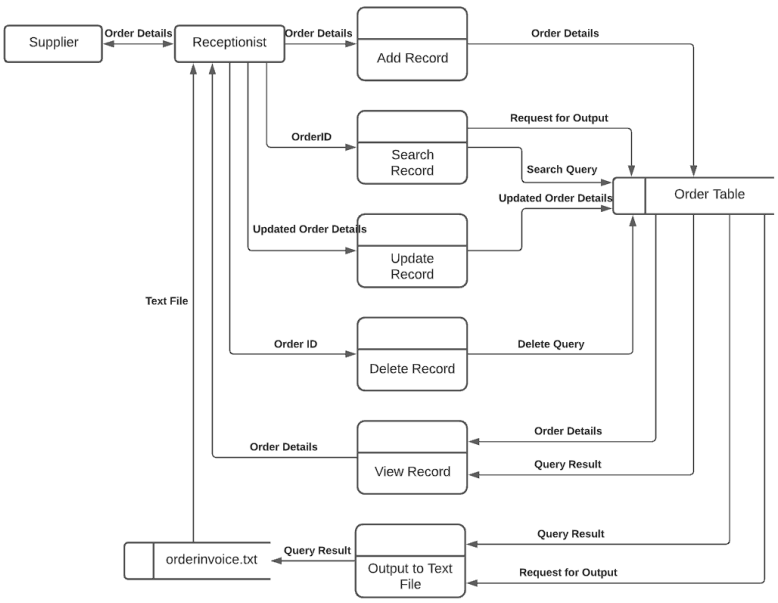
Level 3 DFD for appointments



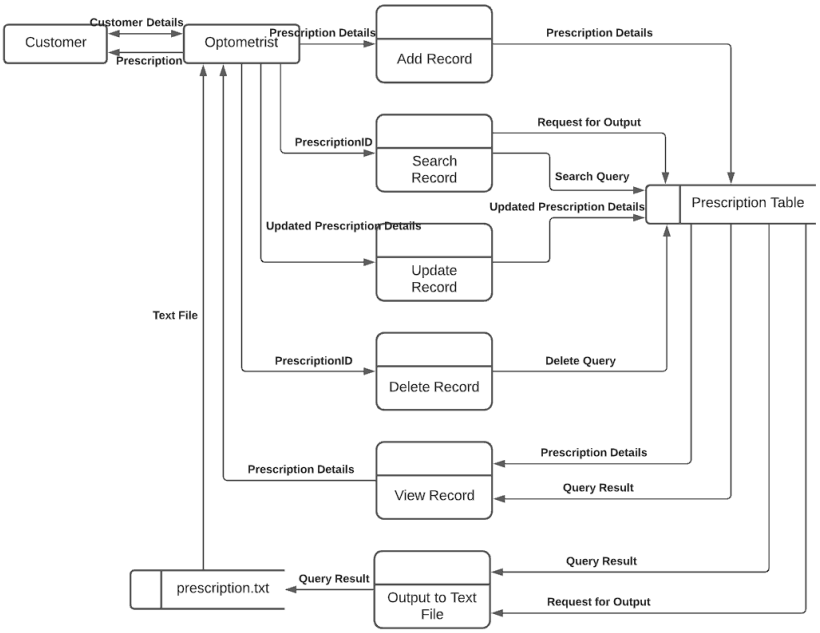
Level 4 DFD for staff



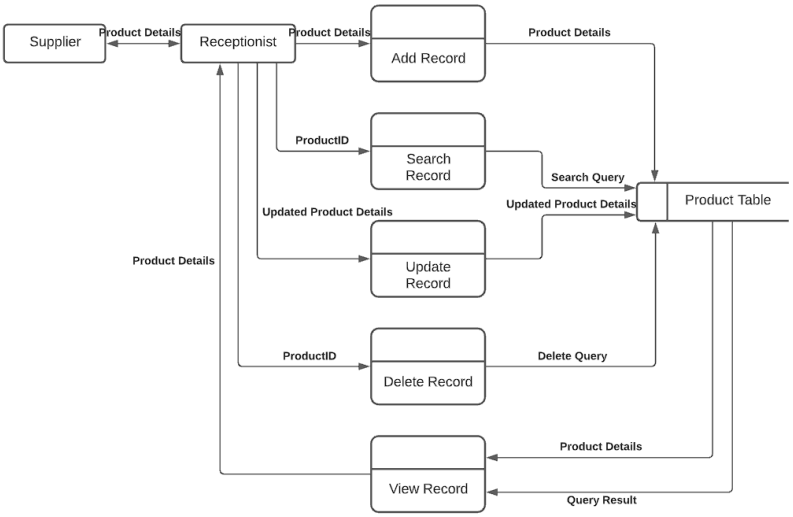
Level 5 DFD for orders



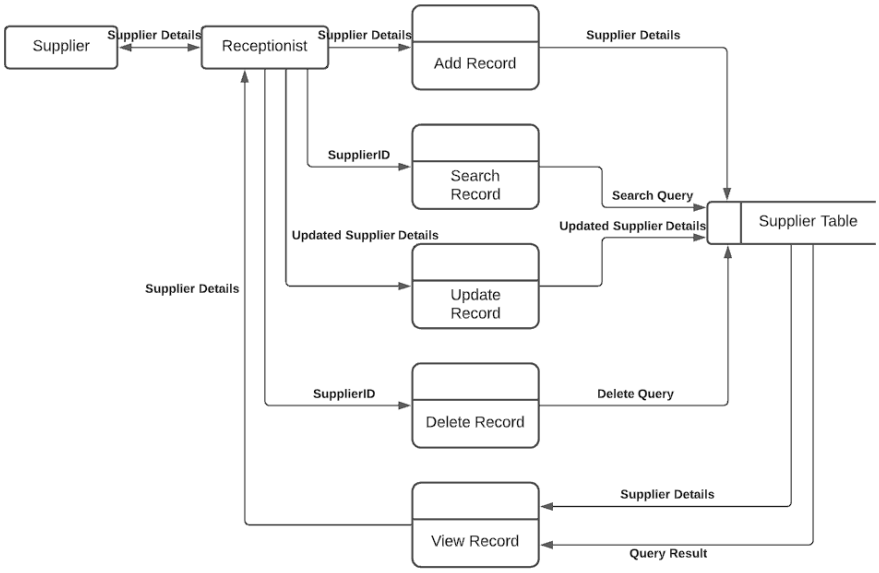
Level 6 DFD for prescriptions



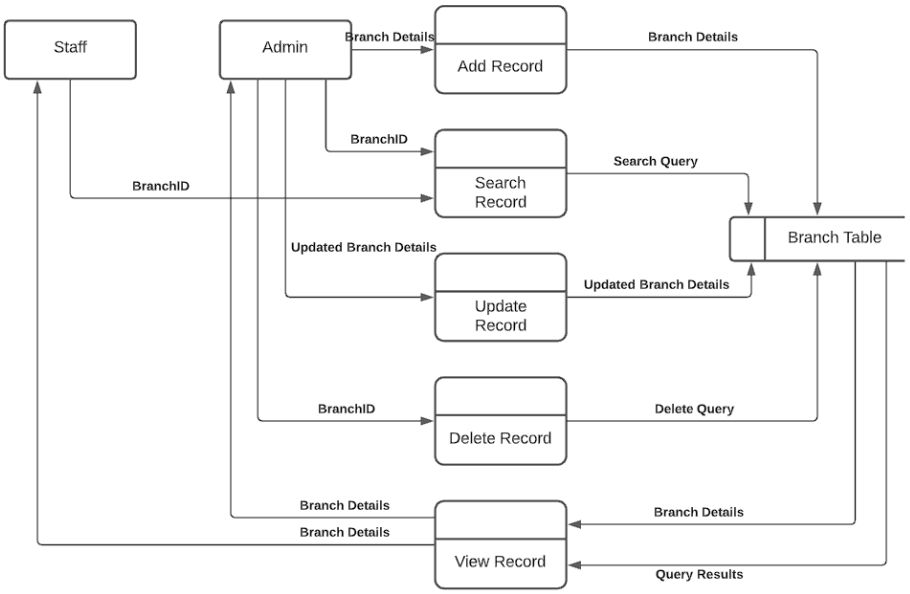
Level 7 DFD for products



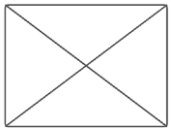
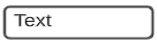
Level 8 DFD for suppliers



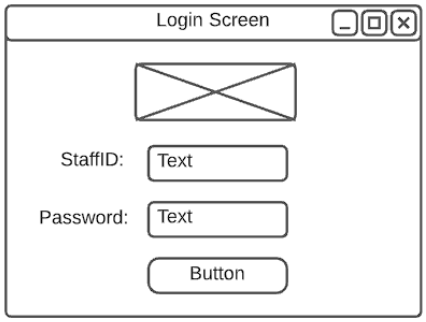
Level 9 DFD for branches



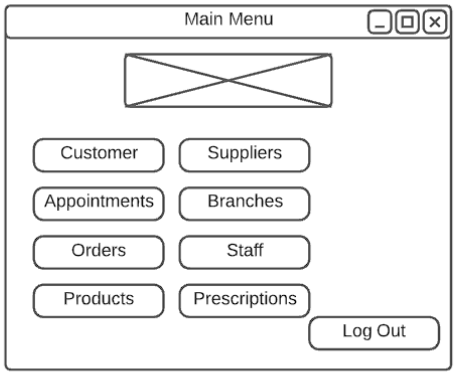
Screen Designs

Window	
Button	
Image	
Entry	

Login Screen

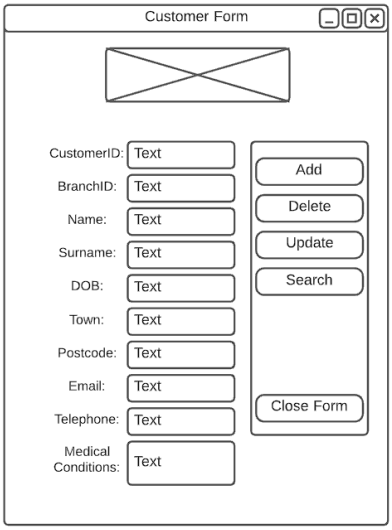


Main Menu Screen

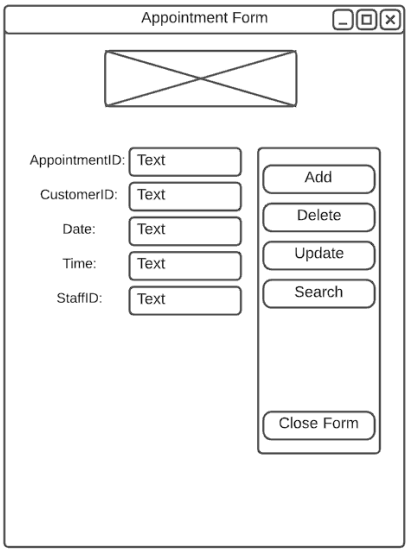


Input Forms

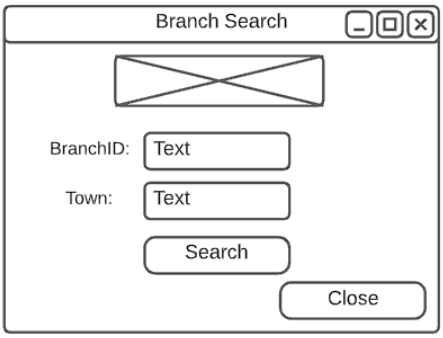
Customer Form



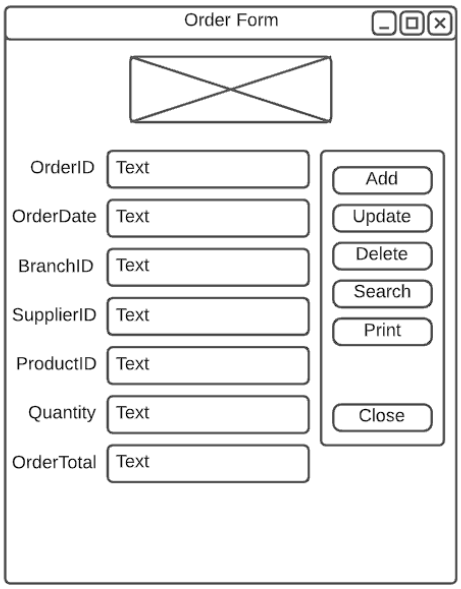
Appointments Form



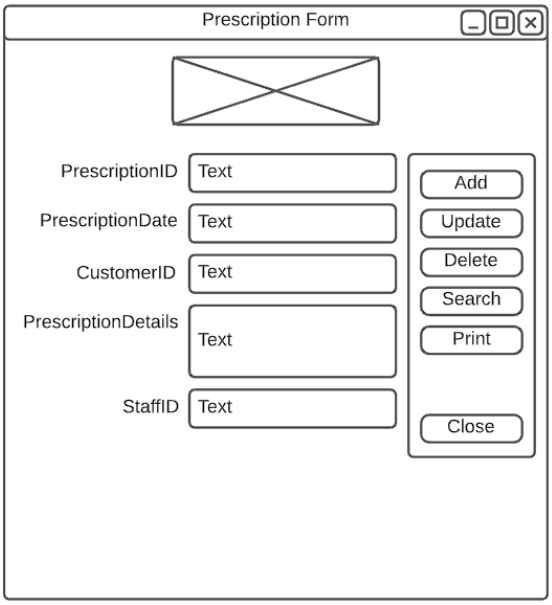
Branch Search Screen



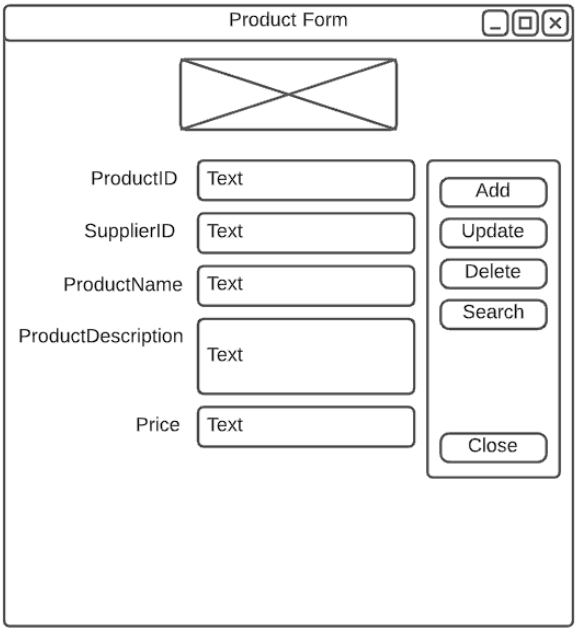
Order Form



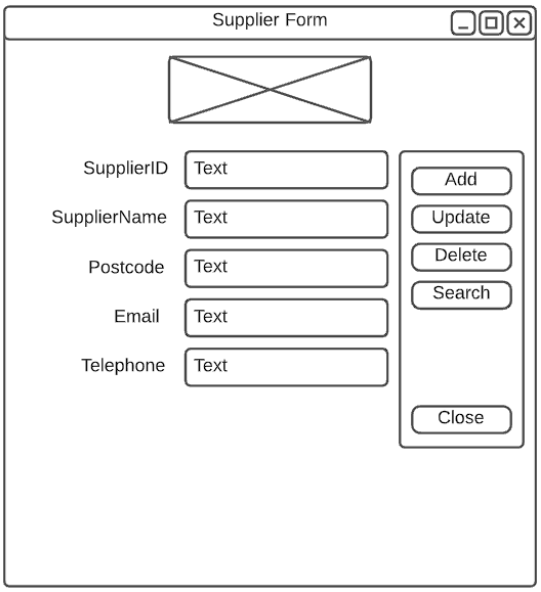
Prescription Form



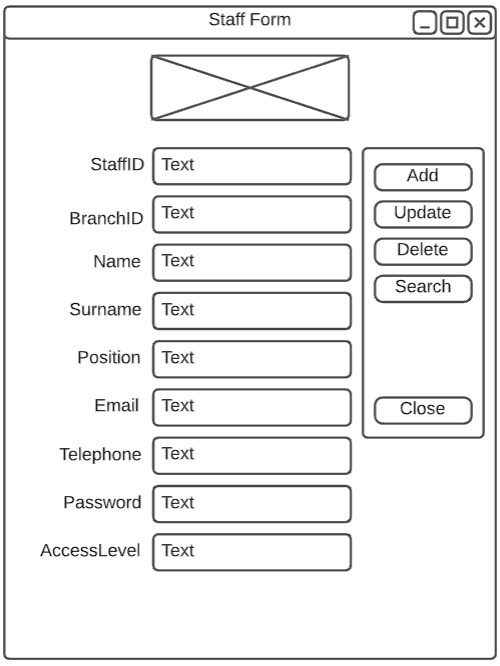
Product Form



Supplier Form



Staff Form



Outputs

Screen Outputs

Error Messages

Error messages will be displayed if an error occurs within the system. This could be a validation error for example. The error message will prompt the user on what went wrong and if applicable, how to fix it.

Notifications

Query Search Results

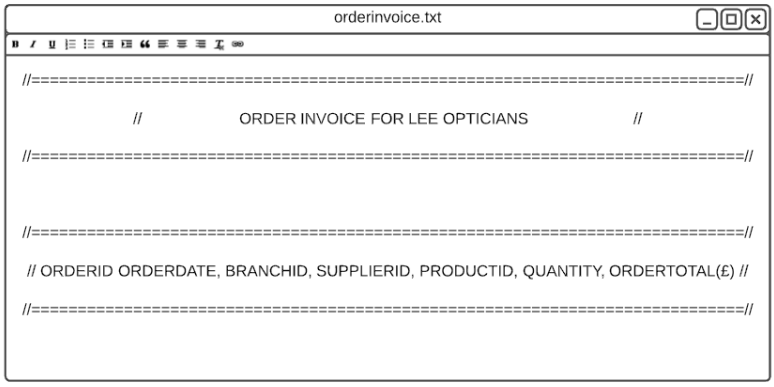
Justification of Output Data

A Tkinter module for message boxes will be used to create alerts for users. Alerts include error messages, notifications, and search results. Using message boxes as a screen output improves the user-friendliness of the system, as they are easy to understand and read prompts or instructions from.

Text Files

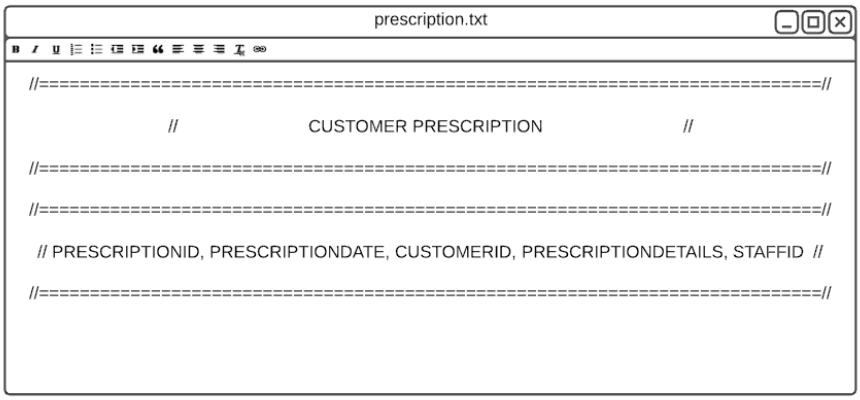
Order Invoice

Inputting order details enables the user to generate an order invoice using a text (.txt) file, which can then be converted to a PDF to be printed. Order invoices are useful for accounting purposes.



Prescription

Inputting prescription details enables the user to generate a prescription using a text(.txt) file, which can then be converted to a PDF to be printed. Prescriptions are required for customers to present when collecting what they are being prescribed.



Justification of Output Data

Using text files as an output for order invoices and prescriptions is suitable for the new system as it is very simple to program. My knowledge on Python is intermediate, therefore, anything more complex may take a longer amount of time. It is also easy to read and understand, improving the user-friendliness of the system, and does not require any additional software such as Microsoft Word or a PDF viewer.

Processing Routines

Access Levels

staffid = staffid entry

SELECT AccessLevels FROM Login WHERE StaffID = staffid

IF search results = 1 then

accesslevel = 1

OUTPUT access level

IF search results = 2 then

accesslevel = 2

OUTPUT access level

ELSE:

accesslevel = 3

OUTPUT access level

Login Screen

```
while TRUE:
    Try:
        staffid = staffid entry
        password = password entry
        SELECT * FROM Login WHERE StaffID = staffid AND Password = password
        IF results found then
            for i in results:
                Check access level
                OUTPUT access level
                Open main menu
                Clear entries
        ELSE:
            OUTPUT error message
    Except value error occurs:
        OUTPUT validation error message
```

Adding a Record

recordinput = record

Connect to database

INSERT INTO table

VALUES (recordinput)

OUTPUT message box

Clear entries

Updating a Record

recordinput = record

Connect to database

UPDATE table

SET

Fields = recordinput

Save changes

OUTPUT message box

Clear entries

Search Record

inputid = id

Connect to database

SELECT * FROM table WHERE ID = inputid

IF results found then

    OUTPUT message box with results

    Clear entries

ELSE:

    OUTPUT error message

    Clear entries

Deleting a Record

inputid = id

Connect to database

DELETE FROM table WHERE ID = inputid

OUTPUT message box

Clear entries

Writing to a Text File

```
inputid = id
Connect to database
SELECT * FROM table WHERE ID = inputid
IF results found then
        for i in results:
                open text file
                write record
                close text file
        OUTPUT message box
        Clear entries
ELSE:
        OUTPUT error message
        Clear entries
```

Validation Routines

Date of Birth Format Check

```
Inputdob = dateofbirth
Try:
        IF inputdob does not equal “DD/MM/YYYY”:
                Raise value error
        Return TRUE
Except value error occurs:
        OUTPUT error message
        Clear date entry
        Return FALSE
```

CustomerID Presence Check

```
Inputid = customerid
Try:
        IF inputid is blank:
                Raise value error
        Return TRUE
Except value error occurs:
        OUTPUT error message
        Clear ID entry
        Return FALSE
```

Appointment Date Format Check

```
appdate = appointmentdate
Try:
        IF appdate does not equal “DD/MM/YYYY”:
                Raise value error
        Return TRUE
Except value error occurs:
        OUTPUT error message
        Clear date entry
        Return FALSE
```

Appointment Time Format Check

```
apptime = appointmenttime
Try:
        IF apptime does not equal “HH:MM”:
                Raise value error
        Return TRUE
Except value error occurs:
        OUTPUT error message
        Clear time entry
        Return FALSE
```

AppointmentID Presence Check

```
appid = appointmentid
Try:
        IF appid is blank:
                Raise value error
        Return TRUE
Except value error occurs:
        OUTPUT error message
        Clear id entry
        Return FALSE
```

SQLite3 Integrity Check

```
Try:
        Add record
        Clear entries
Except sqlite3 integrity error occurs:
        OUTPUT error message
        Clear ID field
```