Hospital Appointment System

System Design

# Overall System Design

The new “Hospital Appointment” system will be created using Microsoft Access. The system will include a multitude of tables namely: Appointments, Clinic Records, Clinic Hours, Doctor Records, Medical Records, Medicine Usage, Patients, Test Results and Login Details.

Before someone can use the system, they must login through the login form which gives them their user group rights. I.e. Admin is able to edit all contents of the system where as a doctor gets read-only options. There must also be an option to create a new user with new login details which are validated through the login form.

To enter new patients into the system, the user must go to the patients form and then enter in all the details of the patient through the text boxes or combo boxes depending on the type of data i.e. entering a name through textbox and Gender through the use of a combo box. Once all the data fields are filled in, the user must press a button which saves the record. In this form, these must be an option to transverse through the records and order the records through the name or the primary key: the NHS Number. The user must also be able to do an advanced search which single out records based on Gender, Town/City and other variables. Doctor Records, Test Results Records, Medicine Usage Records and Clinic Records are created in a similar way to this.

The “Clinic Records” form must have 2 sub forms. The first is “Clinic Hours” which will contain the opening hours and dates of the clinic and the second is “Employed Doctors” which contains the doctors Name, Gender, Speciality, and their holiday dates and return dates.

The “Appointments” Form must contain a reference number (the primary key) for the appointment and must contain the NHS number of the patient involved as well as the clinic he or she must see. The reference number should be generated through the use of a button which generates a random number from 1 to 100,000. The appointment date and time must be checked such that the date and time does not conflict with a date and time that has been previously entered for the same clinic. Also stated on the appointments form must be the treatment the patient is to receive and two check boxes which indicate if it is the patients first appointment and if the patient attended the appointment or not. Similar to that of the rest of the forms, the form must have an option to save and create a new record as well as to navigate through the different appointment records.

In addition, on the appointment form should be a button which enables the user to generate a letter for the patient which takes into account the patients appointment details and the patient’s personal details and merges those details with a word document. Furthermore the form should allow the user to create letter labels for the letter in generates. Both the letter and label must present an option to print.

A “Summary” Form should also be created which presents statistics such as the number of: Clinics, Doctors, Patients, amount of appointments attended etc. The form must also have an option to print the summary and on the summary form should be the date it was generated.

All of the forms should be accessible through one main form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inputs | Processes | | Storage | Outputs |
| Patient Details | Sorts Data | Login details | | Did Not Attend Appointment Letter |
| Doctor Details | Patients Advanced Search | New User Info | | First Appointment Letter |
| Clinic Details | Login User Rights | Patient Details | | Normal Appointment Letter |
| Appointment Details | Generate Summary | Doctor Details | | Summary Report |
| Login Details | Did Not Attend Appointment Query | Clinic Details | | Letter Labels |
| Search Parameters | First Appointment Query | Appointment Details | | Summary Report |
| Medical Record Details | Normal Appointment Query | Medical Record Details | | Appointments Report List |
| Medicinal Usage Details | Generate Reference Number | Medicinal Usage Details | | Clinic Address Book/Phonebook |
| Test Result Details | Generate applicable Letter | Test Result Details | | Patient Address/Phonebook |
| Holiday/Leaves | Generate applicable letter labels | Holiday/Leaves | |  |
| New User Details | Date & Time Available Check |  | |  |
|  | Check Security Code |  | |  |
|  | Appointments Query |  | |  |
|  | Create New User |  | |  |

Definition of Data Requirements

|  |  |
| --- | --- |
| **Data Item** | **Purpose** |
| **Forename** | Used to identify a person |
| **Surname** | Used to identify a person |
| **Gender** | Used to identify gender specific health problems and regulate gender specific medication. |
| **Date of Birth (D.O.B)** | Used to restrict certain medication with an age limit or restricting operations with an age limit |
| **Address** | Used to contact a person |
| **Postcode** | Used to contact a person |
| **Town/City** | Used to contact a person |
| **Phone Number** | Used to contact a person |
| **Mobile Number** | Used to contact a person |
| **Email Address** | Used to contact a person |
| **NHS Number** | Used to uniquely identify each patient’s record |
| **Known Allergies** | Used to restrict certain medication which people are allergic too. |
| **Emergency Contact Name** | Used for contacting a person in case of an emergency. |
| **Emergency Contact Relationship** | Used to identify what relation the patient has to their emergency contact |
| **Emergency Contact Phone 1** | Used for contacting the patients emergency contact |
| **Emergency Contact Phone 2** | Used for contacting the patients emergency contact |
| **Attachment(s)** | Used to provide a visual representation of the patient/clinic |
| **Tested For** | Used to establish what a patient has previously been tested/screened for |
| **Dates of Test** | Used to identify the date of a patients latest test/screening and to decide if a patient needs another test/screening |
| **Result of Test** | Used to establish if the patient has been diagnosed with any problems |
| **Past Major Illnesses** | Used to verify if a current illness/medical problem may be caused by a past illness. |
| **Past Surgeries** | Used to identify whether or not a medical problem has been resolved |
| **Weight** | Used to determine the patients weight which can affect specific surgeries |
| **Height** | Used to determine the patients Height which can affect specific surgeries |
| **Notes** | Used to give additional information which may be vital for a surgery/Test |
| **Past Letters** | Used to give additional information of past test/surgeries/problems the patient may have suffered |
| **Medicine** | Used to determine which medication a patient has taken and regulate which medication a patient cannot take other medication due to this. |
| **Medicine Dosage** | Used to ascertain the strength of the medication used by a patient and whether this will effect operations or the effect of other medication. |
| **Medicine Usage End Time** | Used determine if a patient can have an operation/use a certain medicine or if it will conflict with what the patient is currently taking. |
| **Medicine Usage Start Time** | Used determine if a patient can have an operation/use a certain medicine or if it will conflict with what the patient is currently taking. |
| **Known Chronic Diseases** | Used to establish if a patient is suffering from a new medical problem or whether the problem is chronic |
| **Genetic Illnesses** | Used to check if a person has an illness that runs through the family or is something completely new |
| **Blood Type** | Used to regulate what blood a patient can’t receive. |
| **Disabilities** | Used to identify if there needs to be any accommodations for the patient’s disability i.e. sign language interpreter, large print etc. |
| **Current Doctor/GP** | Used to contact a patients GP |
| **Current Doctor/GP Phone** | Used to contact a patients GP |
| **Speciality** | Used to establish what the doctor and/or clinic specializes in. |
| **Works for Clinic (ID)**  **(i.e. Doctor Works for Clinic)** | Used to determine which clinic a doctor works for so that they can easily be contacted. |
| **Clinic ID** | Used as a unique identifier of a clinic |
| **Booked Holiday Dates** | Used to determine when specific doctors will be unavailable. |
| **Name of Clinic** | Used to identify the clinic. |
| **Opening Hours** | Used to determine whether or not an appointment can be made for a patient at a specific time. |
| **Day** | Used to determine whether or not an appointment can be made for a patient on a specific day at a clinic |
| **Start Time** | Used to determine the earliest time an appointment can be made for a clinic |
| **End Time** | Used to determine the latest time an appointment can be made for a clinic |
| **Appointment Date** | Used to determine what date an appointment is supposed to take place at. |
| **Appointment Time** | Used to determine what time an appointment is supposed to take place at. |
| **First Appointment** | Used to establish if the patient is attending the first appointment or not and the correct procedures are carried out for both circumstances |
| **Did Not Attend** | Used to establish if the patient if the patient attended the appointment or not and the correct procedures are carried out for both circumstances |
| **Appointment Created Date** | Used to identify when an appointment was made as to avoid confusion about whether or not an appointment needs a letter to be generated or not. |
| **Ref Number** | Used to uniquely identify an appointment |
| **Treatment** | Used to establish what a patient is being treated for |
| **Assigned Clinic** | Used to establish which clinic a patient will be attending for their treatment. |
| **Booked Leave Date** | Used to identify on which dates a doctor will be unavailable. |
| **Return Date** | Used to ascertain on which date a doctor will be returning from their day(s) off. |
| **Absence Reason** | Used to ascertain why a doctor was absent i.e. Holiday, illness etc. |
| **Login Name** | One of the variables used to determine what right(s) a user receives once logging in |
| **Password** | Used to verify if the correct details have been entered |
| **Admin** | Used to establish if the user is an admin or a doctor and then give them their respectful rights to the system. |
| **Duration (Minutes)** | Used to identify how long an appointment is. |
| **End Time** | Used to establish when the appointment will end. |

Description of the Modular Structure of the System

## Data Flow Diagram

NHS number, date and time of appointment and name of preferred clinic and date and time of appointment

Patient NHS number, PMH, Contact Details, Reference Letter etc

Phone Patient’s Choice Centre

Email or Phone Centre or Book Online

GP Sends Reference/ Books Appointment Online

Patient

Patient NHS number, PMH, Contact Details, Reference Letter etc

NHS number, date and time of appointment and name of preferred clinic and date and time of appointment

Verify details, appointment date & time available for clinic and doctor

Patient’s Choice Centre

Online Booking System

Patient’s Record

Updates Patient Details/ New Patient Record

Sends patient’s details if available

Date, Time, Clinic name, patient NHS number, Doctor Name

Verify if first appointment. If it is, it creates details for patient

Clinic open time, clinic name

Doctor’s operating hours & Name

Clinic’s Record

Date, Time, Clinic name, patient NHS number, Doctor Name

Doctor’s Record

Appointment Details

Create Appointment

Appointments Record

Appointment details

Appointment details

Patient’s details

Clinic

Appointment details

Letter

Create follow up appointment letter

Create first appointment letter if applicable.

Letter

Patient

Appointment details

Letter

Verify patient attended appointment

Appointments Record

If DNA, Create “Did not Attend” appointment letter

Patient’s details

## Hierarchical Structure Diagram

Login Form

Login Module

Create Account

Navigation Form

Patient

Information

Appointments

Clinics

Summary Report

Appointments Query

Patients Advanced Search

Test

Results

Medicine Usage

Medical Records

DNA Query

Normal Appoint Letter

First Appoint Letter

Did Not Attend Letter

Clinic Hours

Doctors

Summary Report

Norm Query

First Query

Systems Flowchart

Patient Requests appointment

Check if patient’s first appointment

Patient Details

Create if Patients Details non existent

Appointment Details

Check available appointments

Clinic Details

Details

Doctor Details

Details

Create Appointment

First Time Appointment

Create appropriate appointment letter

Normal Appointment

Description of the Record Structure

#### Entity Relationship Diagram

Below is a proposed entity Relationship diagram of the system.

Medicine Usage

Medical Record

Patients

Test Results

Appointments

Clinic Records

Holiday/Leaves

Doctor Records

Clinic Hours

#### Entity Relationship Table

Below is a table to propose what relationships the entities should have and why they should have them.

|  |  |  |
| --- | --- | --- |
| Entity | Relationship | Why |
| Medical Record and Medicine Usage | **1 :** ∞ | Because many of the patients can take more than piece of medication. |
| Medical Record and Test Results | **1 :** ∞ | Because many of the patients can have more than one test and therefore, more than one test results. |
| Medical Record and Patients | **1 : 1** | Because one patient can only have one medical record (one set of medical details) |
| Patients and Appointments | **1 :** ∞ | Because one patient can have many appointments. |
| Clinic Records and Appointments | **1 :** ∞ | Because one Clinic can take many appointments. |
| Clinic Records and Doctor Records | **1 :** ∞ | Because one Clinic can have many doctors |
| Clinic Records and Clinic Hours | **1 :** ∞ | Because one Clinic has many days that it’s open for. |
| Doctor Records and Holiday/Leaves | **1 :** ∞ | One doctor can have many holidays/leaves |

#### Entity Description

Below I have proposed what attributes each entity should have as well as their primary key; what foreign keys are in which entity and what primary keys are composite keys.

**MEDICINE USAGE (NHS Number, Medicine, Medicine Dosage, Medicine Usage Start Time, Medicine Usage End Time)**

**Note: -** The Primary key should be a composite key of NHS Number and Medicine.

**MEDICAL RECORD (NHS Number, Past Major Illnesses, Past Surgeries, Known Chronic Diseases, Genetic Illnesses, Disabilities, Blood Type, Weight (Kg), Height (cm), Notes, Past Letters)**

**Note: -** The NHS Number should act as a foreign key for the following Entities: Medicine Usage and Test Result.

**TEST RESULT (NHS Number, Tested For, Result of Test, Date of Test)**

**Note:** - The primary key should be a composite key of NHS Number and Tested For

**PATIENTS (NHS Number, Forename, Surname, Date of Birth, Address, Gender, City/Town, Postcode, Home Phone Number, Mobile Phone Number, Emergency Contact Name, Emergency Contact Relationship, Emergency Contact Phone 1, Emergency Contact Phone 2, Current Doctor/GP, Current Doctor/GP Phone, Attachments)**

**Note: -** The NHS Number should act as a foreign key for the following Entities: Medical Record and Appointments.

**APPOINTMENTS (NHS Number, Ref Number, Appointment Date, Appointment Time, Did Not Attend, Assigned Clinic, First Appointment, Treatment, Appointment Created Date, Duration (Minutes))**

**Note:** - The primary key should be a composite key of NHS Number and Ref Number

**CLINIC HOURS (Clinic ID, Day, Start Time, Finish Time)**

**Note:** - The primary key should be a composite key of Clinic ID and Day

**CLINIC RECORDS (Clinic ID, Name of Clinic, Speciality, Address, Postcode, Phone Number, Attachments)**

**Note: -** The Clinic ID should act as a foreign key for the following Entities: Clinic Hours and Doctor Records.

**DOCTOR RECORDS (NHS Number, Work for Clinic (ID), Forename, Surname, Gender, Speciality)**

**Note: -** The NHS Number should act as a foreign key for the Holiday/Leaves Entity. The Primary key should also be a composite key of Work for Clinic (ID) and NHS Number.

**HOLIDAY/LEAVES (NHS Number, Booked Leave Date, Return Date, Absence Reason)**

**LOGIN DETAILS (Login Name, Password, Admin)**

#### Data Requirements

Below is a list of all the tables proposed to be in the system along with their field names, Descriptions and types.

##### **Medicine Usage Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | NHS Number | Foreign Key  Primary Key | Number  (Long Integer) | Uniqueness Presence |
| 2 | Medicine | Primary key  Default: None, Required | Text  (String(60)) | Presence |
| 3 | Medicine Dosage | Dosage of Medicine tables/hour | Text  (String(80)) | Presence |
| 4 | Medicine Usage  Start Time | Required | Date/Time  (General Date) | Presence |
| 5 | Medicine Usage  End Time | Required | Date/Time  (General Date) | Presence |

##### **Medical Record Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | NHS Number | Primary Key | Number  (Long Integer) | Uniqueness Presence |
| 2 | Past Major Illnesses | Default: None, Required | Text  (String(200)) | Presence |
| 3 | Past Surgeries | Default: None, Required | Text  (String(200)) | Presence |
| 4 | Known Chronic Diseases | Default: None, Required | Text  (String(200)) | Presence |
| 5 | Genetic Illnesses | Default: None, Required | Text  (String(200)) | Presence |
| 6 | Disabilities | Default: None, Required | Text  (String(200)) | Presence |
| 7 | Blood Type | Required | Text  (List Box(4)) | Presence |
| 8 | Weight (Kg) | Required | Number  (Long Integer) | Presence |
| 9 | Height (cm) | Required | Number  (Long Integer) | Presence |
| 10 | Notes | Extra Notes for Patient | Memo | Presence |
| 11 | Past Letters | Past GP Letters | Attachment |  |

##### **Test Results Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | NHS Number | Foreign Key  Primary Key | Number  (Long Integer) | Uniqueness Presence |
| 2 | Tested For | Primary key  Default: Nothing, Required | Text  (String(60)) | Presence |
| 3 | Result of Test | Required | Text  (List Box (40)) | Presence |
| 4 | Date of Test | dd/mm/yyyy | Date/Time  (General Date) | Range  <= Date() |

##### **Patients Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Uniqueness Presence |
| 1 | NHS Number | Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Forename | First Name  Required | Text  (String(30)) | Presence |
| 3 | Surname | Last Name  Required | Text  (String(30)) | Presence |
| 4 | Date of Birth | Required | Date/Time  (General Date) | Presence  Range |
| 5 | Address | Street Name  Required | Text  (String(80)) | Presence |
| 6 | Gender | Male or Female  Required | Text  (List Box(7)) | Presence |
| 7 | City/Town | Required | Text  (String(40)) | Presence |
| 8 | Postcode | Required | Text  (String(8)) | Presence |
| 9 | Home Phone Number | Required | Text  (String(13)) | Presence |
| 10 | Mobile Phone Number | Required | Text  (String(13)) | Presence |
| 11 | Emergency Contact Name | First Name + Last Name | Text  (String(60)) |  |
| 12 | Emergency Contact Relationship | Selection | Text  (List Box(25)) |  |
| 13 | Emergency Contact  Phone 1 | Phone number | Text  (String(13)) |  |
| 14 | Emergency Contact  Phone 2 | Mobile phone number | Text  (String(13)) |  |
| 15 | Current Doctor/GP | Required | Text  (String(40)) | Presence |
| 16 | Current Doctor/GP Phone | Required | Text  (String(13)) | Presence |
| 17 | Attachments | Picture of Person | Attachments | Presence |

##### **Appointments Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Uniqueness  Presence |
| 1 | NHS Number | Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Ref Number | Required | Number  (Long Integer) | Presence |
| 3 | Appointment Date | dd/mm/yy  Required | Date/Time  (Short Date) | Presence  Range |
| 4 | Appointment Time | Required | Date/Time  (Short Time) | Presence |
| 5 | Did Not Attend | Check Box | Yes/No |  |
| 6 | Assigned Clinic | Required | Text  (String(200)) | Presence |
| 7 | First Appointment | Check box | Yes/No |  |
| 8 | Treatment | Required | Text  (String(60)) | Presence |
| 9 | Appointment Created Date | Required | Date/Time  (General Date) | Presence |
| 10 | Duration (Minutes) | Duration of Appointment | Number  (Long Integer) | Presence |

##### **Clinic Records Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type |  |
| 1 | Clinic ID | Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Name of Clinic | Required | Text  (String(60)) | Presence |
| 3 | Speciality | Required | Text  (String(60)) | Presence |
| 4 | Address | Required | Text  (String(80)) | Presence |
| 5 | Postcode | Required | Text  (String(8)) | Presence |
| 6 | Phone Number | Required | Text  (String(13)) | Presence |
| 7 | Attachments | Picture of Clinic | Attachment |  |

##### **Login Details Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | Login Name | Primary Key | Text  (String(30)) | Uniqueness  Presence |
| 2 | Password | Input Masked  Required | Text  (String(24)) | Format  Presence |
| 3 | Admin | Check Box | Yes/No |  |

##### **Clinic Records Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | Clinic ID | Foreign Key  Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Day | Primary key  Required | Text  (String(60)) | Uniqueness  Presence |
| 3 | Start Time | Required | Date/Time  (Short Time) | Presence |
| 4 | Finish TIme | Required | Date/Time  (Short Time) | Presence |

##### **Doctor Records Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | NHS Number | Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Forename | Required | Text  (String(30)) | Presence |
| 3 | Surname | Required | Text  (String(30)) | Presence |
| 4 | Gender | Male or Female  Required | Text  (List Box(7)) | Uniqueness  Presence |
| 5 | Speciality | Required | Text  (String(60)) | Presence |
| 6 | Work for Clinic (ID) | Primary Key  Foreign Key, Required | Number  (Long Integer) | Uniqueness  Presence |

##### **Holiday/Leaves Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref No. | Field Name | Description | Type | Validation? |
| 1 | NHS Number | Foreign Key  Primary Key | Number  (Long Integer) | Uniqueness  Presence |
| 2 | Booked Leave Date | Required | Date/Time  (Short Date) | Presence |
| 3 | Return Date | Required | Date/Time  (Short Date) | Presence |
| 4 | Absence Reason | Reason for Absence  Required | Memo | Presence |

Validation

For this system, there should be multitude of validation checks. Below are the validation checks suggested for this system:

Length Check: - Length checks are going to be used for all fields possible. The length limit for each field is present in the data requirements tables

Type Check: - Each and every field has a type check already implemented as the system is designed in access which has its own simple type check. The type checks for my fields are present in the data requirements tables.

Check Digit: - This can be set up through having a security code which all admins must enter when logging in to enable to make changes to the system. The security code’s characters are then checked through a module.

If the security code was invalid, an error message would popup stating that the code was invalid.

Uniqueness Check: - Each table should have a primary key and therefore each record should be unique. The unique fields are located in the Data Requirements tables

Format Check: - Format checks can be used for the Password field with asterisks as the input mask

(Refer to Login Details Table Ref No. 2)

Range Check: - Range check can be used for the Date of Birth i.e. <= 01/01/1900

If the D.O.B year is smaller than 1900 (such as 1765) then an error message will state that the D.O.B must be larger than 01/01/1900

Presence Check: - Presence checks can be set up by applying the required attribute to the NHS Number field as it is a primary key. Almost all my fields will have a presence check to make sure that the data is valid.

Database Design including Normalised Relations

#### File Organisation

Since I am using Microsoft Access to create my system, the file organisation is conducted in the background by Microsoft Access and is determined for me.

#### Normalised Entity Relationship Model

F = Foreign Key P = Primary Key

|  |
| --- |
| Medicine Usage |
| NHS Number (FP)  Medicine(P)  Medicine Dosage  Medicine Usage Start Time  Medicine Usage End Time |

1: ∞

|  |
| --- |
| Medical Record |
| NHS Number(FP)  Past Major Illnesses  Past Surgeries  Known Chronic Diseases  Genetic Illnesses  Disabilities  Blood Type  Weight (Kg)  Height (cm)  Notes  Past Letters |

|  |
| --- |
| Test Result |
| NHS Number (FP)  Tested For (P)  Result of Test  Date of Test |

|  |
| --- |
| Appointments |
| NHS Number(FP)  Ref Number(P)  Appointment Date  Appointment Time  Did Not Attend  Assigned Clinic  First Appointment  Treatment  Appointment Created |

|  |
| --- |
| Patients |
| NHS Number(P)  Forename  Surname  Date of Birth  Address  Gender  City/Town  Postcode  Home Phone Number  Mobile Phone Number  Emergency Contact Name  Emergency Contact Relationship  Emergency Contact Phone 1  Emergency Contact Phone 2  Current Doctor/GP  Current Doctor/GP Phone  Attachments |

∞:1

1:1

1: ∞

|  |
| --- |
| Clinic Hours |
| Clinic ID (FP)  Day(P)  Start Time  Finish Time |

|  |
| --- |
| Doctor Records |
| NHS Number(P)  Forename  Surname  Gender  Speciality  Work for Clinic (ID)(FP) |

|  |
| --- |
| Holidays/Leaves |
| NHS Number(FP)  Booked Leave Date  Return Date  Absence Reason |

|  |
| --- |
| Clinic Records |
| Clinic ID(P)  Name of Clinic  Speciality  Address  Postcode  Phone Number  Attachments |

1: ∞

1: ∞

∞:1

The main entities in the above system are the Clinic Records, Doctor Records, Medical Records, Patients and Appointments.

All the tables should be in 3rd Normal Form as no table should have a dependency outside the primary keys. This is because the values of the attributes should not changes unless the primary key is changed (meaning a user has navigated to another record).

Sample of planned SQL or QBE Queries

Here are some queries that are planned to be in the system.

## Query #1 – Did Not Attend Query

**Included Tables**: - Patients, Appointments, and Clinic Records

**Fields**: - NHS Number, Appointment Date, Appointment Time, Assigned Clinic, Forename, Surname, Address (Patients), City/Town, Postcode (Patients), Did Not Attend, Treatment, Address (Clinic Records), Postcode (Clinic Records)

**Criteria**: - Assigned Clinic (Appointments) = Name of Clinic (Clinic Records) and Did Not Attend = True

**Calculated Fields**: - None

**Sort Criteria**: - NHS Number (Ascending)

**Purpose of Query**: - To produce the detail necessary for the “Did Not Attend” mail merged letter.

## Query #2 – First Appointment Query

**Included Tables**: - Patients, Appointments, and Clinic Records

**Fields**: - NHS Number, Appointment Date, Appointment Time, Assigned Clinic, Forename, Surname, Address (Patients), City/Town, Postcode (Patients), Did Not Attend, First Appointment, Treatment, Address (Clinic Records), Postcode (Clinic Records)

**Criteria**: - Assigned Clinic (Appointments) = Name of Clinic (Clinic Records), Did Not Attend = False, First Appointment = True

**Calculated Fields**: - None

**Sort Criteria**: - NHS Number (Ascending)

**Purpose of Query**: - To produce the detail necessary for the “First Appointment” mail merged letter.

## Query #3 – Normal Appointment Query

**Included Tables**: - Patients, Appointments, and Clinic Records

**Fields**: - NHS Number, Appointment Date, Appointment Time, Assigned Clinic, Forename, Surname, Address (Patients), City/Town, Postcode (Patients), Did Not Attend, First Appointment, Treatment, Address (Clinic Records), Postcode (Clinic Records)

**Criteria**: - Assigned Clinic (Appointments) = Name of Clinic (Clinic Records), Did Not Attend = False, First Appointment = False

**Calculated Fields**: - None

**Sort Criteria**: - NHS Number (Ascending)

**Purpose of Query**: - To produce the detail necessary for the “Normal Appointment” mail merged letter.

## Query #4 – Appointments Query

**Included Tables**: - Appointments

**Fields**: - NHS Number, Appointment Date, Appointment Time, Assigned Clinic, Did Not Attend, First Appointment, Treatment, Duration (Minutes), Ref Number

**Criteria**: - None

**Calculated Fields**: - End Time: Add Duration (Minutes) to Appointment Time

**Sort Criteria**: - NHS Number (Ascending)

**Purpose of Query**: - To produce an End Time for the Appointment and then use this time to avoid conflicting appointments.

Identification of Storage Media and Format

The storage medium for the system will be a magnetic disc; this is because a magnetic disc has sufficient space for my database system as it is not a large program. The file size for the program is approximately 5-6mb and with additional records (1000’s) it could very well reach 40mb-50mb.

This piece of code initially takes the security code entered by the user and checks it with the algorithm. If it turns out to be a valid security code then the code takes the username, password and user group and checks them against the login details stored in the login table.

If the details are correct and the user is an admin, the code then makes visible the “Create User” label. The code also changes the properties of every form in the system such that they can be modified by the admins.

If the details are correct and the user is a doctor, the code keeps the “Create User” label hidden. The code also changes the properties of all the forms such that the forms become read only.

If the security code is incorrect, then the program stops and brings up a message box stating that the user had entered an incorrect security code. If the security code is correct and the username and/or the password are wrong then the code stops and brings up an error message stating that they have entered an incorrect username and password combination.

Identification of Processes and Suitable Algorithms for Data Transformation (Pseudo code)

## Process #1 – Logging into the System (For Login Form)

### Task

1. Take Security Code
2. Take Login Name
3. Take Password Value
4. Take User Group Value
5. Check Security Code with algorithm
   1. If it is false, end the program and display error message
   2. If it is true then continue and set Security Code to true
6. Check Login Name, password and User group with the details in the table.
   1. If Login Name, password or user group are invalid then display error message
   2. If login Name and password are true then continue
      1. If user group is admin
         1. Display message welcome admin
         2. Set Valid Details to true
         3. Set Admin Account to true
         4. Close and Save Login Form.
      2. If user group is doctor
         1. Display welcome doctor message
         2. Set Valid Details to true
         3. Set Admin Account to False
         4. Close and Save Login Form
7. If Admin Account is true then call the procedure “Rights” with True
   1. Right sets all forms to allow edits, additions, deletion and opens the main form
8. If Admin Account is False then call procedure “rights” with False
   1. Right sets all forms to NOT allow edits, additions, deletion and opens the main form

### Pseudo code

This function will test if the security code is valid

Function Check Digit (SecNo as text) As Boolean

Declare Valid As Boolean

Declare AsciValues (20) As Number

Declare CheckLetters As Text

Declare IDString (20) As text

Declare IDNum (20) As Number

Declare Total (5) As Number

Here the function takes each value of the security code separately and tests if it is numeric or not. If it is numeric, then AsciiVal(C ) is equal to Idstring(c) . If the idstring(C ) is a letter, than the AsciiVal(c) converts it into a number

Declare Remainder As Number

k= Number

Check Digit = False

For k = 1 to 20

StringID (k) = Mid (SecNo, k, 1)

If IsNum(IDString (k)) = True then AsciValues (k) = IDString (k)

Elseif IDString (k) = “-“ then AsciValues (k) = 0

Else AsciValues (k) = ascivalue(IDString (k))

End If

Next k

Here it tests if the values of the characters are greater than 9. If it is, it takes 64 away from the numbers. If the numbers are less than or equal to 9, they will remain the same.

For k = 1 to 20

If AsciValues (k) > 9 then IDNum (k) = AsciValues (k) – 64

Else IDNum (k) = AsciValues (k)

End If

Next k

Total (1) = (IDNum (1) + IDNum (2) + IDNum (3) + IDNum (4)) x 2

Total (2) = (IDNum (6) + IDNum (7) + IDNum (8) + IDNum (9)) x 3

Total (3) = (IDNum (11) + IDNum (12) + IDNum (13) + IDNum (14)) x 4

Total (4) = (IDNum (16) + IDNum (17) + IDNum (18) + IDNum (19)) x 5

Total (5) = ((Total (4) + Total (3)) - (Total (2) + Total (1)))

Then the totals are added up and a remainder is found. If the first character of the security code is a number then check letter remains to be a number otherwise it is converted into a letter.

The security code then checks the 20th character in the security code to see if it’s equal to the check letter. If it is, then Check Digit is set to true.

Remainder = Total (5) mod 9

If IsNum(IDString (1)) = True then CheckLetter = Remainder

Else CheckLetter = Character(Remainder +65)

End If

If the CheckLetter = Mid (SecNo, 20, 1) then Check Digit = true

Private Sub btnLogin\_Click()

Dim rs As DAO.Recordset

Set rs = CurrentDb.OpenRecordset("Select \* From [Login Details]")

Dim rcl As Integer

This is the login in code.

First validation checks are done.

Dim C As Integer

Dim e As Boolean ' Record Valid

Dim A As Boolean ' Admin Account

Dim S As Boolean ' Security Code

e = False

S = False

If IsNull(SecCode) = True Then

Checks if the security code is null or if the length is length than 20 and brings up a message accordingly.

MsgBox ("Enter your Security Code")

Exit Sub

ElseIf Len(SecCode) < 20 Then

MsgBox ("Incorrect Security Code Length")

Exit Sub

End If

If CheckDigit(SecCode) = True Then

MsgBox ("Valid Security Code")

S = True

Else

MsgBox ("Invalid Security Code")

Exit Sub

End If

rs.MoveLast

rcl = rs.RecordCount

rs.MoveFirst

If S = True Then

Checks if login field is empty

If IsNull(Me.login) = True Then

MsgBox ("Please Enter Your Login Name")

Exit Sub

Checks if password field is empty

ElseIf IsNull(Me.Password) = True Then

MsgBox ("Please Enter A password")

Exit Sub

ElseIf Len(Me.Password) < 6 Then

MsgBox ("Please Enter a Password with 6 to 24 Characters" & vbNewLine & "Your Password is Currently Less than 6 Characters")

Exit Sub

ElseIf Len(Me.Password) > 24 Then

MsgBox ("Please Enter a Password with 6 to 24 Characters" & vbNewLine & "Your Password is Currently More than 24 Characters")

Exit Sub

ElseIf IsNull(Me.UserGroup) = True Then MsgBox ("Please Select a UserGroup")

Exit Sub

Else

For C = 1 To rcl

If UserGroup = "Admin" And Me.login = rs("Login Name") And Me.Password = rs("Password") And rs("Admin") = True Then

MsgBox ("Welcome Admin")

A = True

e = True

DoCmd.Close acForm, "Login Form", acSaveYes

C = rcl

ElseIf UserGroup = "Doctor" And Me.login = rs("Login Name") And Me.Password = rs("Password") And rs("Admin") = False Then

e = True

A = False

MsgBox ("Welcome Doctor")

DoCmd.Close acForm, "Login Form", acSaveYes

Here is when the details of the login are checked with the records in the login details table. If they are correct, a welcome message is sent to the Admin.

C = rcl

If the record is found and it’s a doctor login, then a welcome message is sent to the doctor.

Else

End If

rs.MoveNext

Next C

If e = False Then

MsgBox "Username, Password and Usergroup combination Not Recognized"

Exit Sub

End If

If the record is not found it states the above message.

If A = False Then

Call Rights(False)

ElseIf A = True Then

Call Rights(True)

End If

Procedure Rights (Value as Boolean)

DoCommand Open all forms, Design View

DoCommand Change, all forms, Allow Additions= Value

DoCommand Change, all forms, Allow Edits = Value

DoCommand Change, all forms, Allow Deletions = Value

DoCommand close,all the forms, save them.

All records are changed according to the value of the Boolean (which is of course dependant on the user group)

DoCommand Open MainForm, NormalView

## Process #2 – Sort Records

### Task

1. Takes the value selected from the combo box
2. Code checks if all the records are ordered in descending order
   1. If they are, then order them by the value selected from the combo box
   2. Otherwise, order them in descending order with the value form the combo box.
3. Make sure the form allows the records to be ordered.

### Pseudo code

If Me.FormOrder Is like “ALL DESCENDING” then

Me.FormOrder = Me.combobox

Else

Me.FormOrder = Me.combobox & “ DESCENDING”

End If

Me.FormOrderEnabled = True

This code allows an entire form to be ordered by whatever fields that is in the combo box. In either ascending or descending order

## Process #3 – Check if Date is Free (For Appointment Form)

### Task

1. Take the date, clinic, time from the form text box called date
2. Search the appointments query from the first record to the last for the same date, time, surgery
   1. If it finds that the date, clinic and time values are identical to one already stored then produces a message stating the date and time is already used for a specific clinic. It goes on to state the appointment reference number, the date of the appointment, time, duration and end time.
   2. Otherwise, it states that the Date is free to book.
3. Once the it has looped until it reaches the end of the records, it closes the database.

### Pseudo code

Private Sub checkDate.Click

Declare Message As Text

Declare FoundDate As Boolean

Declare db As DAO.Database

Declare rsAppointments As DAO.RecordSet

Set rsAppointments = CurrentDatabase.OpenRecordSet(“Select ALL from Apppointments Query)

FoundDate = False

rsAppointments .MovetoBeginning

Do

If Me.AssignedClinic = rsAppointments.AssingnedClinic AND Me.AppointmentDate = rsAppointments.AppointmentDate AND

(Me.EndTime <= rsAppointments.EndTime OR Me.StartTime >= rsAppointments.StartTime) Then

Message = “Date and Time is booked for “ & rsAppointments.AssignedClinic & newline

Message = Message & “Reference Number” rsAppointments.RefNo & newline

Message = Message & “Start Time” rsAppointments.StartTime & newline

Message = Message & “Duration” rsAppointments.Duration & newline

Message = Message & “EndTime” rsAppointments.EndTime & newline

FoundDate = True

This piece of ensures no double booking can take place. It first checks if the clinics are the same as well as the appointment date. It then does a series of checks to ensure that the appointment times cannot conflict. This is to ensure that no two appointments can be at the same clinic on the same date and have their appointment dates conflict. If the appointments do conflict, a message is produces as seen above.

End If

rsAppointments.MovetoNextRecord

Loop Until rsAppointments.LastRecord

This piece of ensures no double booking can take place. It first checks if the clinics are the same as well as the appointment date. It then does a series of checks to ensure that the appointment times cannot conflict. This is to ensure that no two appointments can be at the same clinic on the same date and have their appointment dates conflict. If the appointments do conflict, a message is produces as seen above.

rsAppointments.Close

If FoundDate = True then

MessageBoxpopup(Message)

Else MessageBoxpopup(This Date is Free)

## Process #4 – Next Record

**Note**: This can be used for all the Next Record buttons.

### Task

1. Check record position
   1. If this is the last record then go to the first record
   2. Else go to the next record

### Pseudo code

If record position = last position then go to first record

Else go to next record

End If

## Process #5 – Previous Record

**Note**: This can be used for all the Previous Record buttons.

### Task

1. Check record position
   1. If this is the first record then go to the last record
   2. Else go to the previous record

### Pseudo code

If record position = first position then go to last record

Else go to previous record

End If

## Process #6 – Save and Close

**Note**: This can be used for all the Save and Close buttons.

### Task

1. Check if the form has been changed
   1. If it has then save the record
2. Close the Form

### Pseudo code

If form = changed then

Save record

End If

DoCommand Close the form

## Process #7 – Save and New Record

**Note**: This can be used for all the Save and New Record buttons.

### Task

1. Check if the forms has been changed
   1. If it has then save the record
2. Create a New Record

### Pseudo code

If form = changed then

save form

End If

DoCommand Create New Record

## Process #8 – Open (Any Form)

**Note**: This can be used for all Open Form buttons. Note that Formname implies that any form name can be set as long as that form exists.

### Task

1. Take NHS Number from current form and store it in a variable
2. If the form has changed the save the record
3. If there is no NHS Number specified then open form “Any Form” to a new record
   1. If there is an NHS Number specified, then open “Any Form” to that NHS number.

### Pseudo code

Declare variable = String

Declare Formname = String

Set Formname to “Any form”

Set variable to “Any form” [NHS Number] must equal current form’s NHSNumber

If the form has been changed then save the record

If the current forms NHSNumber is empty then

DoCommand Open “Any form” to a new record

Else

DoCommand Open “Any form” to when it is equal to variable.

## Process #9 – Generate Label (For Appointment Form)

### Task

1. Save all changes to the form and records
2. OpenReport “Patients” in preview mode where NHSNumber is equal to current form’s NHS Number

### Pseudo code

Do Command Save Record

Do Command Open Report “Patients” in preview mode where NHSNumber = current form’s NHS Number

## Process #10 – Generate Reference Number (For Appointment Form)

### Task

1. Check the length of the Appointments form Refnumber text box
   1. If it is 0, then create a random number for the Refnumber textbox from 100 to 999999 by using a “Random” function

### Pseudo code

If Length(RefNumber) is 0 then

Refnumber = Random(100,999999)

End If

Function Random (lowerbound, upperbound) as long number

Random = integer value of ((upperbound – lowerbound add 1) x Random function add lowerbound))

## Process #11 – Close Form Window

**Note**: This can be used for all Forms.

### Task

1. Close current form window and save changes

### Pseudo code

DoCommand Close Form, “Any Form” and SaveChanges

## Process #12 – Create Account (For Create User Form)

### Task

1. Open a recordset (all records) with all the details from table “Login Details”
2. Open a second recordset (admin) with all the details which have admin set to true from the table “Login Details”
3. Set valid to False
4. Count the amount of records in both record sets and store them in separate variables
5. Check if the details the admin has entered into the form for his login name and password are correct by comparing them with the details from them admin recordset.
   1. If they are a match then set valid to true.
6. If the details are correct then check the new user account details ie. The new login name and password for the user with the recordset(all records).
   1. If the new user account username matches with one in the record set then display an error message and set valid to false
7. If valid is true then if the usergroup is set to admin, then create a new account with the login name and password that was specified and then set the accounts rights to: Admin is = true
   1. Else input the login name and password specified and set the account’s rights to: Admin = False

### Pseudo code

Declare rsAdmin = Recordset

Declare rsEveryRecord = Recordset

Declare rsARecordsCount = Integer

Declare rsERRecordsCount = Integer

Declare C = Integer

Declare Valid = Boolean

Set RsAdmin to an openrecordset with all details from “login details” table where admin is = true

Set rsEveryRecord to an openrecordset with all details from “login details”

Valid = False

This is the code that is used to create user accounts.

rsAdmin.MoveRecordstoEnd

rsARecordsCount = rsAdmin.Countallrecords

rsAdmin.MoveRecordstoBeginning

rsEveryRecord.MoveRecordstoEnd

rsERRecordsCount = rsEveryRecord.Countallrecords

rsEveryRecord.MoveRecordstoBeginning

For C from 1 to rsARecordsCount

If AdminLogin = rsAdmin(AdminLogin) and AdminPass = rsAdmin(AdminPass) then

Valid is true

End If

rsAdmin.MovetoNextRecord

Next value for C

For C from 1 to rsERRRecordsCount

If NewUsername = rsEveryRecord(LoginName) then

Messagebox (Choose a Different Name)

This code will not allow non-unique user names

Valid is false

End If

This bottom section prevents two usernames from being the same

This section checks if the details typed into the form exist in the login details table.

rsEveryRecord.MovetoNextRecord

Next value for C

If valid is true then if NewUserGroup is Admin then

rsEveryRecord.addnewrecord

rsEveryRecord(LoginName) = NewLoginName

rsEveryRecord(Password) = NewPassword

If valid is true and all fields contain valid data and the admin details were logged in correctly, then an admin or doctor account will be created depending on which user group was chosen

rsEveryRecord(Admin) is set to true

Messagebox(“Admin Account Created”)

ElseIf NewUserGroup is Doctor then

rsEveryRecord.addnewrecord

rsEveryRecord(LoginName) = NewLoginName

rsEveryRecord(Password) = NewPassword

rsEveryRecord(Admin) is set to False

Messagebox(“Doctor Account Created”)

End If

End If

## Process #13 – Advanced Search (Patient’s Advanced Search Form)

### Task

1. If user clicks on “Single Item Search” then set all relevant labels and radio buttons visible property to true.
   1. Once user selects the criteria that they want to search by, take their inputted criteria from the single textbox.
      1. If the text box is not empty then open the form “Patient Details” where it is equal to the specified criteria.
      2. If no criteria is entered then show an error message
2. If the user clicks on “Multiple item search” then hide all the single item search labels, textboxes and radio buttons and set visible all the relevant labels, textboxes and checkboxes.
   1. Take the variables the user enters into the multiple text boxes.
      1. Add together the filter terms for every variable the user wants to search by and then open the “Patients Details” form where it is equal to the criteria specified.

### Pseudo code

Searchoption button.click

Select case of radiooptionsgroupname

Case 1

Call SearchVisible procedure(True, False)

Case 2

Call SearchVisible procedure(False, True)

End Select

Sub SearchVisible( singlesearch as boolean, multisearch as Boolean)

Me.Single.Visible = singlesearch

Me.SS.Visible = singlesearch

Me.SSForename.Visible = singlesearch

Me.SSSurname.Visible = singlesearch

Me.SSAdd.Visible = singlesearch

This procedure sets either the multi search options to visible or single search options through the use of a radio button.

Me.SSPost.Visible = singlesearch

Me.SSDate.Visible = singlesearch

Me.optForename.Visible = singlesearch

Me.optSurname.Visible = singlesearch

Me.optAdd.Visible = singlesearch

Me.optPost.Visible = singlesearch

Me.optDate.Visible = singlesearch

Me.btnSS.Visible = singlesearch

Me.SSearch.Visible = singlesearch

Me.SM.Visible = multisearch

Me.SMForename.Visible = multisearch

Me.SMSurname.Visible = multisearch

Me.SMAdd.Visible = multisearch

Me.SMPost.Visible = multisearch

Me.SMDate.Visible = multisearch

Me.chkForename.Visible = multisearch

Me.chkSurname.Visible = multisearch

Me.chkAdd.Visible = multisearch

Me.chkPost.Visible = multisearch

Me.chkDate.Visible = multisearch

This procedure runs the single search

Me.btnSM.Visible = multisearch

Me.Forename.Visible = multisearch

Me.Surname.Visible = multisearch

Me.Add.Visible = multisearch

Me.Post.Visible = multisearch

Me.Date.Visible = multisearch

Me.SR.Visible = multisearch

End Sub

btnSingleSearch Click

Declare Where As String

Select Case SingleSearch

Case 1

SingleSearch ("[Forename]")

Here, the Single Search has many cases statements as Me.Single could be Forename or Surname or any of the listed cases. These in turn, run the single search procedure.

Case 2

SingleSearch ("[Surname]")

Case 4

SingleSearch ("[Address]")

Case 5

SingleSearch ("[Postcode]")

Case 6

SingleSearch ("[Date of Birth]")

End Select

End Sub

Sub SingleSearch(Field as String)

If textboxcriteria is not empty then

DoCommand Open Form "Patient Details” where Field is = textboxcriteria

Else

MessageBox "No Criteria Entered"

End If

End Sub

Here for multi search the fields are tested for validation purposes again to ensure correct data entry.

btnSearchMultiple Click

Declare Where = String

Dim StringLength As Long

If Forename Field is not empty And Forename checked = True Then

Where = Where& "([Forename] = """ & Me.Forename & """) AND "

End If

If Surname Field is not empty And Surname checked True Then

Where= Where& "([Surname] = """ & Me.Surname & """) AND "

End If

If Address Field is not empty And Address checked = True Then

Where= Where& "([Address] = """ & Me.Add & """) AND "

End If

If Postcode Field is not empty) And Postcode checked = True Then

Where= Where& "([Postcode] = """ & Me.Post & """) AND "

End If

If Date of Birth Field is not empty) And Date of Birth checked = True Then

Where= Where& "([Date of Birth] = """ & Me.Date & """) AND "

Left$(Where,StringLength) removes 5 off the length of the field Where because the AND ” needs to be removed for the Where to work when opening a form.

End If

StringLength = Len(Where) - 5

If StringLength <= 0 Then

MsgBox "No criteria"

Else

Where= Left$(Where, StringLength)

DoCommandd OpenForm where "Patient Details” opens to criteria of variable Where

End If

End sub

## Process #14 – Generating Relevant Appointment Letter (Appointments Form)

### Task

1. The program checks the form to see if the “Did not attend” button is checked or if the “First Appointment” button is checked or if they are neither checked.
2. Following this, the system runs a procedure which links each of the options in step 1 to a query and then sets up a query definition for any of the three applicable queries.
3. The program then sets up the name for the new letter which is dependent on each query ie. The DNA Query (Did not attend query) will have a file name called “DNA” followed by the date added to the file name.
4. The program then calls another procedure which proceeds to merge the details it has gathered from the queries to a template letter located in a specified directory.
5. The program then executes a mail merge function and saves the letter in the same directory as the template letters.

### Pseudo code

Sub SetQuery(QueryName As String, SQL As String)

Declare NewQueryDef = QueryDef

If Me.chkAttend isTrue Then

QueryName is "DNA Query"

ElseIf Me.chkFirst isTrue Then

QueryName is"First Query"

Else

QueryName is"Norm Query"

End If

Set NewQueryDef to CurrentDb.QueryDefs(QueryName)

NewQueryDef.SQL is strSQL

Close NewQueryDef

End Sub

GenLetter Click

Declare DocumentName = String

Declare SQL = String

Declare NewName = String

If Did Not Attend checked = True Then

SQL = "SELECT Appointments.[NHS Number], Appointments.[Appointment Date], Appointments.[Appointment Time], Appointments.[Assigned Clinic], Patients.Forename, Patients.Surname, Patients.Address, Patients.[City/Town], Patients.Postcode, Appointments.[Did Not Attend], Appointments.Treatment, [Clinic Records].Address, [Clinic Records].Postcode FROM [Clinic Records], Patients INNER JOIN Appointments ON Patients.[NHS Number] = Appointments.[NHS Number] WHERE (((Appointments.[Assigned Clinic])=[Name of Clinic] And (Appointments.[Assigned Clinic])=[Name of Clinic] And (Appointments.[Assigned Clinic])=[Name of Clinic]) AND ((Appointments.[Did Not Attend])=True));"

DocumentName = "\DNA.dotx"

Call SetQuery("DNA Query", SQL)

NewName = "DNA " & Format((Date), "yyyymmdd")

ElseIf First Appointment checked = True Then

SQL = "SELECT Appointments.[NHS Number], Appointments.[Appointment Date], Appointments.[Appointment Time], Appointments.[Assigned Clinic], Patients.Forename, Patients.Surname, Patients.Address, Patients.[City/Town], Patients.Postcode, Appointments.[First Appointment], Appointments.[Did Not Attend], Appointments.Treatment, [Clinic Records].Address, [Clinic Records].Postcode FROM [Clinic Records], Patients INNER JOIN Appointments ON Patients.[NHS Number] = Appointments.[NHS Number] WHERE (((Appointments.[Assigned Clinic])=[Name of Clinic] And (Appointments.[Assigned Clinic])=[Name of Clinic]) AND ((Appointments.[First Appointment])=True) AND ((Appointments.[Did Not Attend])=False));"

DocumentName = "\First.dotx"

Call SetQuery("First Query", SQL)

NewName = "First Appointment " & Format(CStr(Date), "dd MMM yyyy")

Else

SQL = "SELECT Appointments.[NHS Number], Appointments.[Appointment Date], Appointments.[Appointment Time], Appointments.[Assigned Clinic], Patients.Forename, Patients.Surname, Patients.Address, Patients.[City/Town], Patients.Postcode, Appointments.[Did Not Attend], Appointments.[First Appointment], Appointments.Treatment, [Clinic Records].Address, [Clinic Records].Postcode FROM [Clinic Records], Patients INNER JOIN Appointments ON Patients.[NHS Number] = Appointments.[NHS Number] WHERE (((Appointments.[Assigned Clinic])=[Name of Clinic] And (Appointments.[Assigned Clinic])=[Name of Clinic]) AND ((Appointments.[Did Not Attend])=False) AND ((Appointments.[First Appointment])=False));"

DocumentName = "\Appointment.dotx"

Call SetQuery("Norm Query", SQL)

NewName = "Appointment " & Format(CStr(Date), "dd MMM yyyy")

End If

Call OpenMergedDoc(DocumentName, SQL, NewName)

Sub OpenMergedDoc(DocName As String, SQL As String, MergedDocName As String)

Const Dir As String = "C:\Users\Kaeze PhoeniX\Desktop\Coursework"

Declare objWord = New Word.Application

Declare objDoc = Word.Document

Set objWord.Application.Visible = True

Set objDoc = objWord.Documents.Open(Dir & DocName)

objWord.Application.Visible = True

If Did Not Attend checked = True Then

objDoc.MailMerge.OpenDataSource \_

Name:=" Coursework\Computing Coursework Source.accdb", \_

LinkToSource:=True, AddToRecentFiles:=False, \_

Connection:="QUERY [DNA Query]", \_

SQLStatement:="SELECT \* FROM [DNA Query]"

ElseIf First Appointment checked = True Then

objDoc.MailMerge.OpenDataSource \_

Name:=" Coursework\Computing Coursework Source.accdb", \_

LinkToSource:=True, AddToRecentFiles:=False, \_

Connection:="QUERY [First Query]", \_

SQLStatement:="SELECT \* FROM [First Query]"

Else

objDoc.MailMerge.OpenDataSource \_

Name:=" Coursework\Computing Coursework Source.accdb", \_

LinkToSource:=True, AddToRecentFiles:=False, \_

Connection:="QUERY [Norm Query]", \_

SQLStatement:="SELECT \* FROM [Norm Query]"

End If

objDoc.MailMerge.Destination = wdSendToNewDocument

objDoc.MailMerge.Execute

objWord.Application.Documents(1).SaveAs (Dir & "\" & MergedDocName & ".docx")

objWord.Application.Documents(2).Close wdDoNotSaveChanges

Set objWord = Nothing

Set objDoc = Nothing

User Interface Design (HCI) Rational

The UI will mainly consist of a series of forms and reports since I am using Microsoft Access. Below is a table of forms listed with their names, purpose and reference code. Below the table is a list of all the potential designs of the forms and all are annotated.

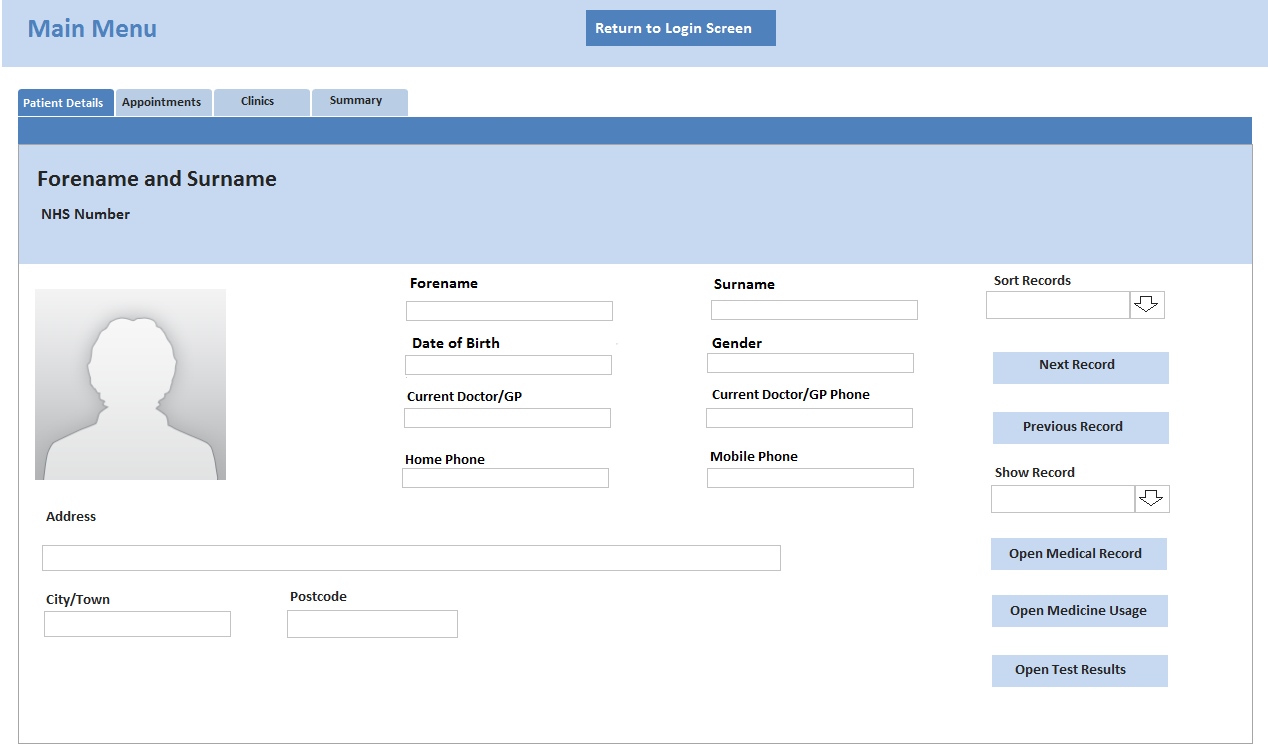
|  |  |  |
| --- | --- | --- |
| Form Name | Purpose | Reference Code |
| Main Menu + Patient Details | This form provides the user a means to navigate through the many forms in the program | F1 |
| Login Form | This form allows a user to login and therefore the program will then allocate the proper rights to the user group that has logged in. | F2 |
| Create User Form | This form allows an admin to create a new account which is able to login to the system. | F3 |
| Clinic Records | Clinics details are added, edited or deleted. | F4 |
| Appointments | Appointments are made, edited or deleted. Allows you to generate letters for each appointment and letter labels. | F5 |
| Medical Record | Patient’s medical record details are added, edited or deleted. | F6 |
| Medical Usage | Patient’s medical usage details are added, edited or deleted. | F7 |
| Test Results | Patient’s test result’s details are added, edited or deleted. | F8 |
| Summary Form | For a summary form to be generated indicating amount of appointments and those that attended and more. | F9 |
| Clinics List | To easily cycle and select a clinic when searching for a specific clinic. | F10 |
| Patients List | To easily cycle and select a patient when searching for a specific patient. | F11 |
| Appointments List | To easily cycle and select an appointment when searching for a specific appointment. | F12 |
| Patient’s Advanced Search | To do a more thorough search of the Patient’s Records | F13 |
| Holiday/Leaves Form | To Add holiday’s/leaves for the doctors | F14 |
| Clinics Phonebook | Report shows contact details for clinic for easy contact. | R1 |
| Patients Address Book | Report shows all patients’ addresses. | R2 |
| Patients Phonebook | Report shows all patients’ contact details. | R3 |
| Summary Report | Summarizes information from the system such as amount of clinics, appointments attended etc. | R4 |
| Did Not Attend Letter | To send the letter to a patient who has missed an appointment | L1 |
| Normal Appointment Letter | To send the letter to a patient to inform them of their appointment. | L2 |
| First Appointment Letter | To send the letter to a patient who is having their first appointment | L3 |

## F1 – Main Menu (With Patient Details Form)

Light blue back colour for Header. Dark blue for button on header with white text, font Calibri, size 14

Form Name: Main Menu Form No: F1

Main Menu in darker light blue font Calibri size 18 and bold.



Tab on focus dark blue background, white text font Calibri size 10. Tabs out of focus light blue back colour Calibri size 10 black colour texts.

Combo box approx. same size as buttons. Text font Calibri size 12 colour black

Light blue back colour buttons with font Calibri size 12 and colour black.

Grey thin border to stand out from white background.

Text size 16 font Calibri and Bold with colour black.

Placeholder image with no borders.

Background plain white

All boxes have grey thin borders.

All box texts are text size 12 font Calibri and colour black.

## F2 – Login Form

Form Name: Login Form Form Ref No: F2



Text font Calibri, size 24, colour dark grey. Heading background colour light blue

NHS Logo top left of screen under the header

Border is thin and light blue

All boxes have grey thin borders.

All box texts are text size 12 fonts Calibri and colour black.

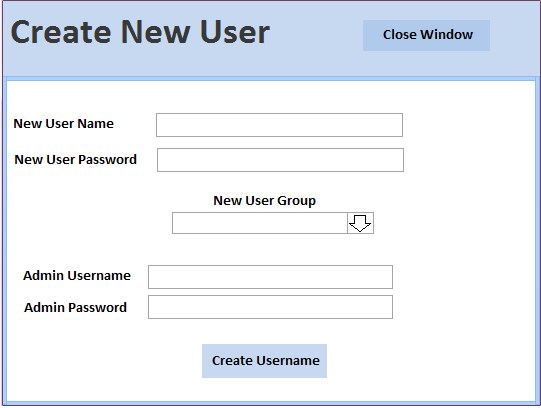
Light blue back colour buttons whose text has font Calibri size 12 and colour black.

Combo box has thin grey border. Text font Calibri, size 12 and colour black.

Background plain white

## F3 – Create User Form

Form Name: Create User Form Form Ref No: F3



Light blue back colour button whose text has font Calibri size 12 and colour black.

Combo box has thin grey border. Text font Calibri, size 12 and colour black.

Background plain white

Border is thin and light blue

Light blue back colour button whose text has font Calibri size 12 and colour black.

All box texts are text size 12 fonts Calibri and colour black.

All boxes have grey thin borders.

Text font Calibri, size 24, colour dark grey. Heading background colour light blue

## F4 – Clinic Records

Light blue back colour for Header. Darker light blue for button on header with black text, font Calibri, size 14

Form Name: Clinic Records Form Ref No: F4

Text font Calibri, size 24, colour dark grey.



Placeholder image with no borders

All box texts are text size 12 font Calibri and colour black.

Combo box has thin grey border. Text font Calibri, size 12 and colour black.

Light blue back colour button whose text has font Calibri size 12 and colour black.

Plain white background

All boxes have grey thin borders.

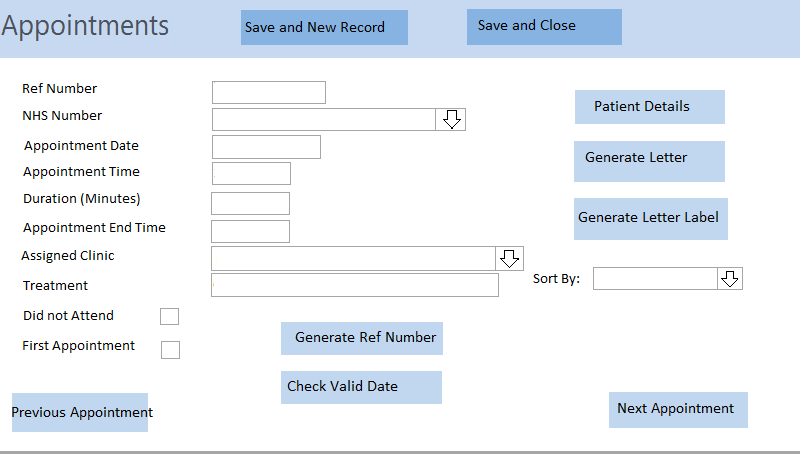
Light blue back colour button whose text has font Calibri size 12 and colour black.

Text font Calibri, colour black and size 10.

## F5 – Appointments

Form Name: Appointments Form Ref No: F5

Light blue back colour for Header. Darker light blue for button on header with black text, font Calibri, size 14



All boxes have grey thin borders.

All box texts are text size 12 fonts Calibri and colour black.

All Combo boxs has thin grey border. Text font Calibri, size 12 and colour black.

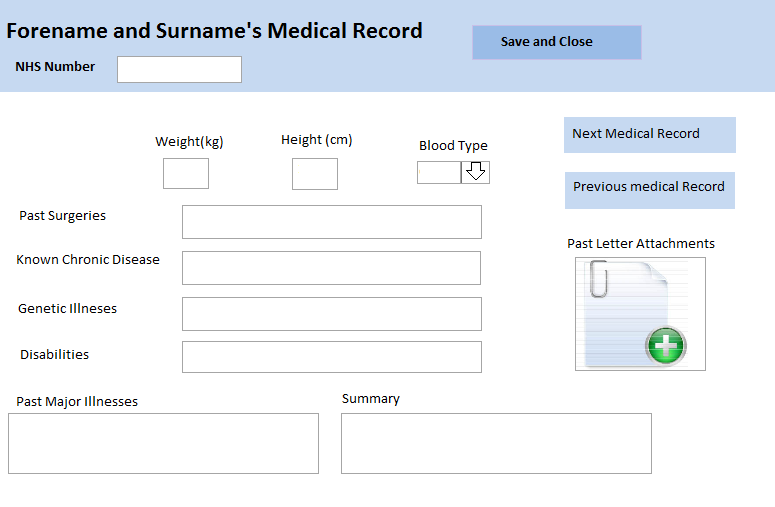
All buttons in main area are light blue back colour button whose text has font Calibri size 12 and colour black.

Text font Calibri, size 24, colour dark grey.

## F6 – Medical Record

Form Name: Medical Record Form Ref No: F6

Light blue back colour for Header. Darker light blue for button on header with black text, font Calibri, size 14



All box texts are text size 12 fonts Calibri and colour black.

All boxes have grey thin borders.

Text font Calibri, size 24, colour dark grey.

Combo box has thin grey border. Text font Calibri, size 12 and colour black.

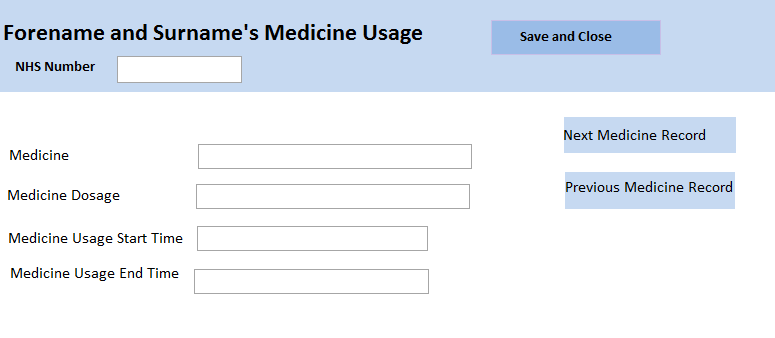
Light blue back colour button whose text has font Calibri size 12 and colour black.

Text font Calibri, size 18, colour black and bold.

Placeholder image with no borders

## F7 – Medical Usage

Form Name: Clinic Records Form Ref No: F7



Light blue back colour for Header. Darker light blue for button on header with black text, font Calibri, size 14

Light blue back colour button whose text has font Calibri size 12 and colour black.

All boxes have grey thin borders.

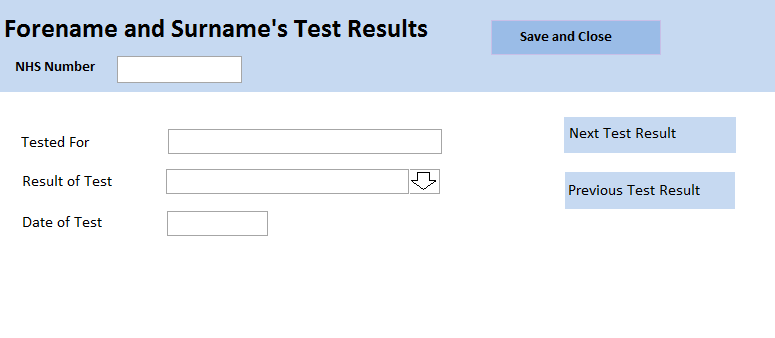
All box texts are text size 12 fonts Calibri and colour black.

Text font Calibri, size 18, colour black and bold.

Text font Calibri, size 24, colour dark grey.

## F8 – Test Results

Form Name: Test Results Form Ref No: F8



All boxes have grey thin borders.

Combo box has thin grey border. Text font Calibri, size 12 and colour black.

Light blue back colour button whose text has font Calibri size 12 and colour black.

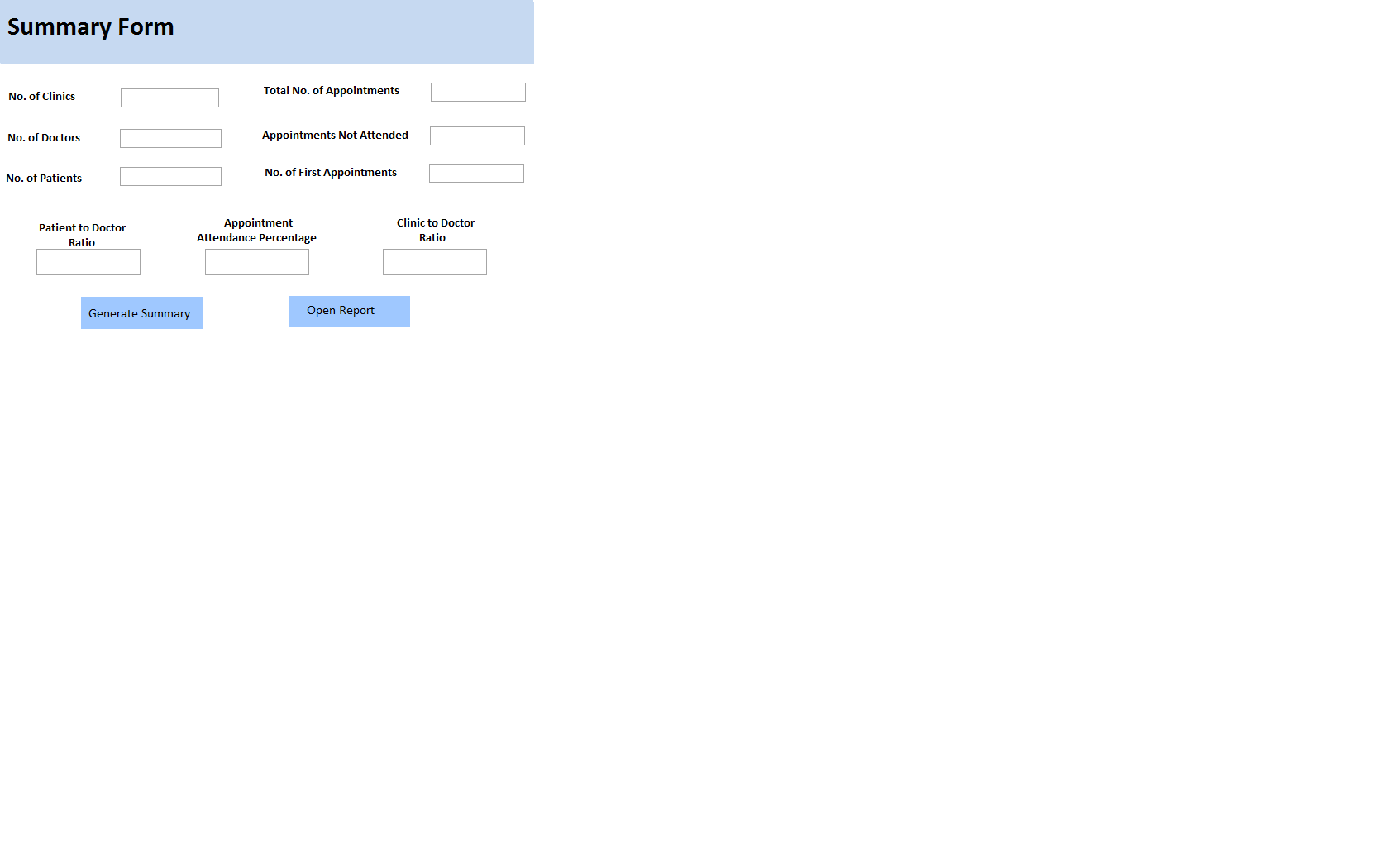
Text font Calibri, size 24, colour dark grey.

Text font Calibri, size 18, colour black and bold.

All box texts are text size 12 fonts Calibri and colour black.

Light blue back colour for Header. Darker light blue for button on header with black text, font Calibri, size 14

## F9 – Summary Form

Form Name: Summary Form Form Ref No: F9

Plain white background

All box texts are text size 12 fonts Calibri and colour black.

Light blue back colour button whose text has font Calibri size 12 and colour black.

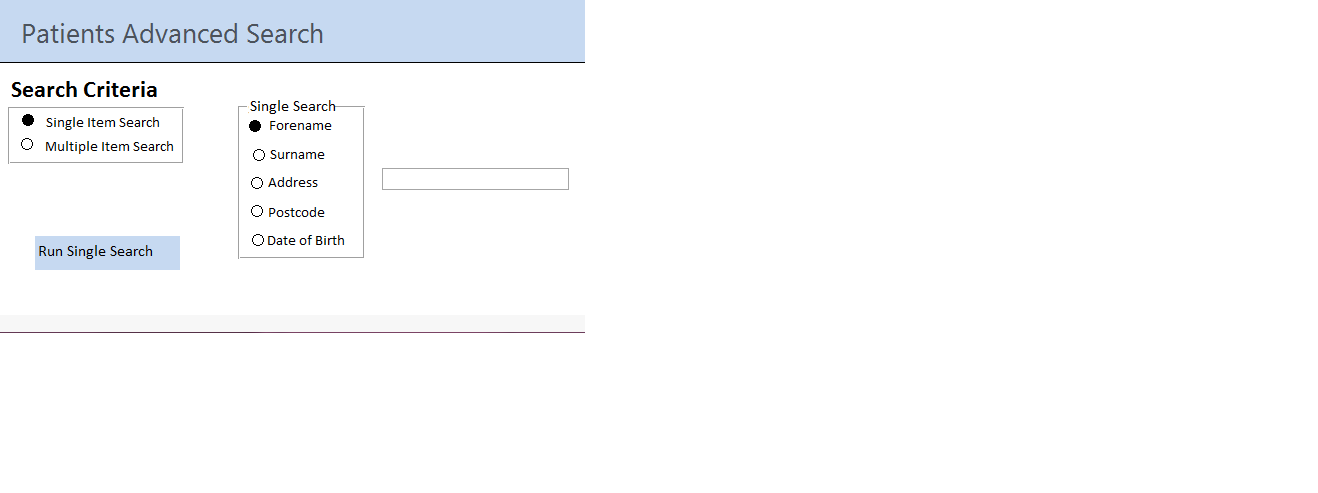
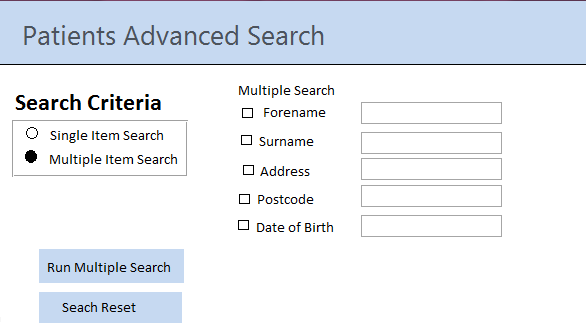
All boxes have grey thin borders.

Light blue back colour for header

Black text, font Calibri, size 24 for title in header

## F13 – Patients Advanced Search

Form Name: Patients Advanced Search Form Ref No: F13



Black text, font Calibri, size 20 for subheader

All box texts are text size 12 fonts Calibri and colour black.

Grey text, font Calibri, size 24 for title in header

Light blue back colour for header

All boxes have grey thin borders.

Light blue back colour button whose text has font Calibri size 12 and colour black.

Light blue back colour button whose text has font Calibri size 12 and colour black.

## R1 – Clinics Phonebook

Dark blue text, font Calibri, size 20 and Bold.

Report Name: Clinic Phonebook Report Ref No: R1



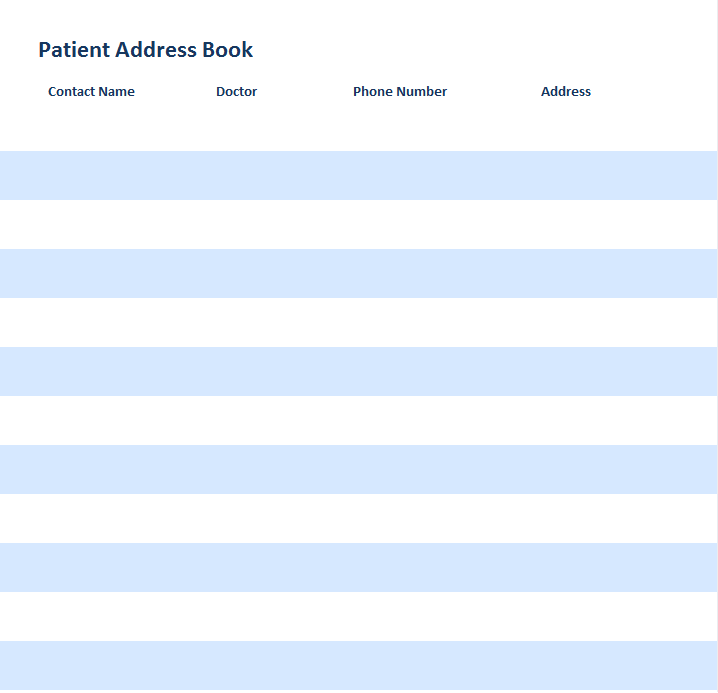
Dark blue text, font Calibri, size 24.

Dark blue text, font Calibri, size 18.

Light blue and White evenly spaced lines

## R2 – Patients Address Book

Report Name: Patients Address Book Report Ref No: R2



Dark blue text, font Calibri, size 24.

Dark blue text, font Calibri, size 18.

Light blue and White evenly spaced lines

## R3 – Patients Phone Book

Report Name: Patients Phonebook Report Ref No: R3

Dark blue text, font Calibri, size 20 and Bold.



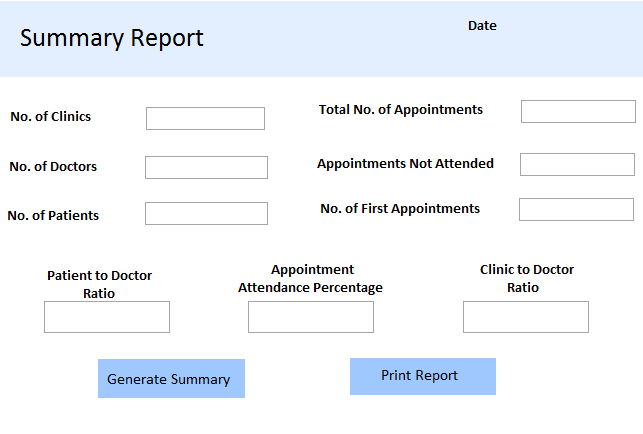
Dark blue text, font Calibri, size 24 for title in header

Dark blue text, font Calibri, size 18 for headings

Light blue and White evenly spaced lines

## R4 – Summary Report

Report Name: Summary Report Report Ref No: R4



All boxes have grey thin borders.

Plain white background

Black text, font Calibri, size 18

Black text, font Calibri, size 24 for title in header

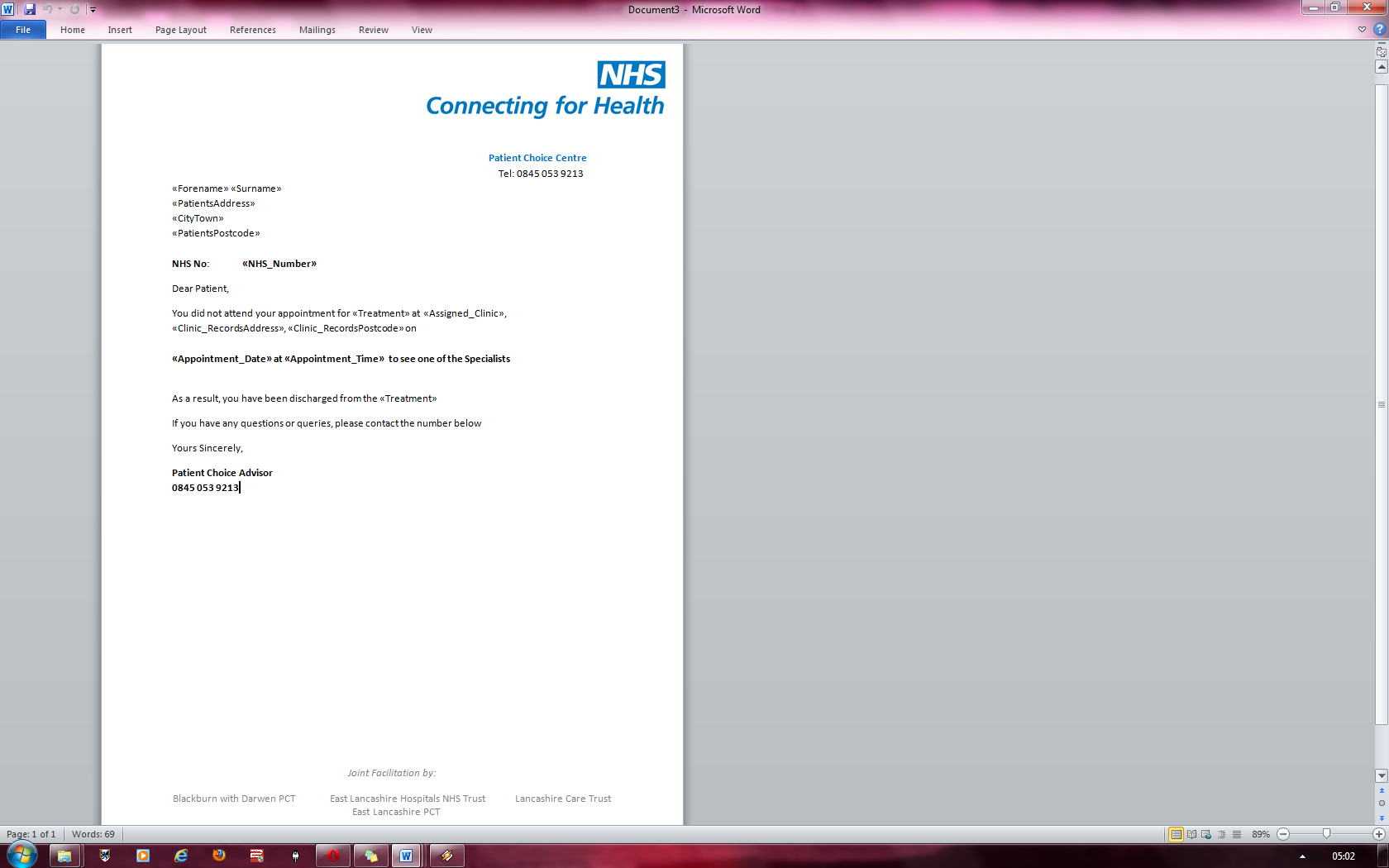
Light blue back colour for header

All box texts are text size 12 fonts Calibri and colour black.

Light blue back colour button whose text has font Calibri size 12 and colour black.

## L1 – Did Not Attend Letter

Letter Name: DNA Letter Ref No: L1



Logo for NHS on Letter Blue and white variation on text

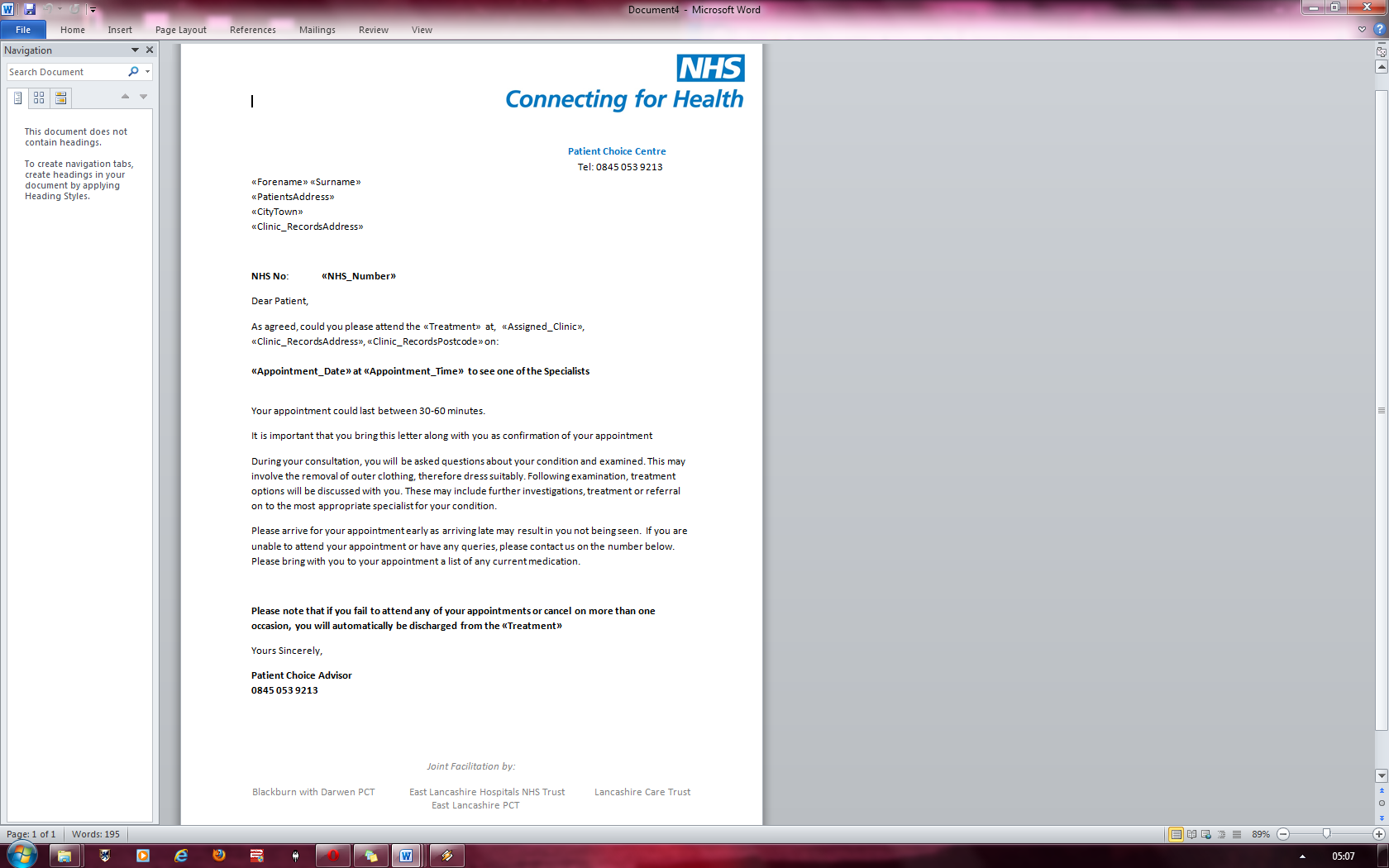
Light grey text footer with size 12 Calibri.

Bold text font Calibri size 12. Also coloured black

Normal text is Calibri font, size 12 and coloured black

## L2 – Normal Appointment Letter

Letter Name: Norm Letter Ref No: L2



Normal text is Calibri font, size 12 and coloured black

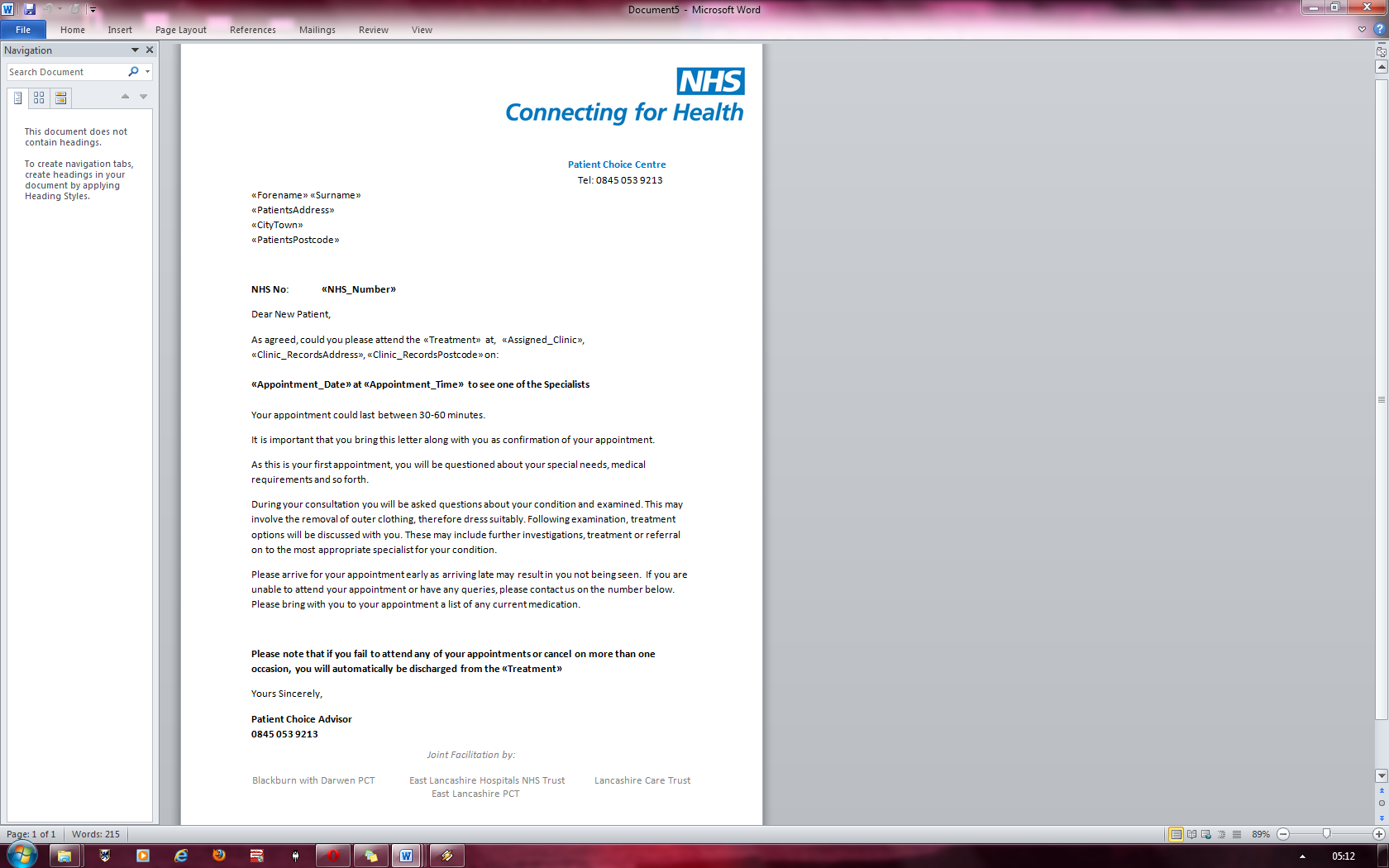
Light grey text footer with size 12 Calibri.

Bold text font Calibri size 12. Also coloured black

Logo for NHS on Letter Blue and white variation on text

## L3 – First Appointment Letter

Letter Name: First Letter Ref No: L3



Light grey text footer with size 12 Calibri.

Normal text is Calibri font, size 12 and coloured black

Bold text font Calibri size 12. Also coloured black

Logo for NHS on Letter Blue and white variation on text

User Interface Design Summary

I chose the light blue/white colour scheme for my system because the text was easy to read off both colours. In addition, it avoids the problem caused by the red/green colours which is the most prevalent type of colour blindness which would make black text hard to read.

My report all have dark blue headings and title with the addition of alternating blue and white segments. The reason this was done was to keep the style consistent throughout the systems/reports.

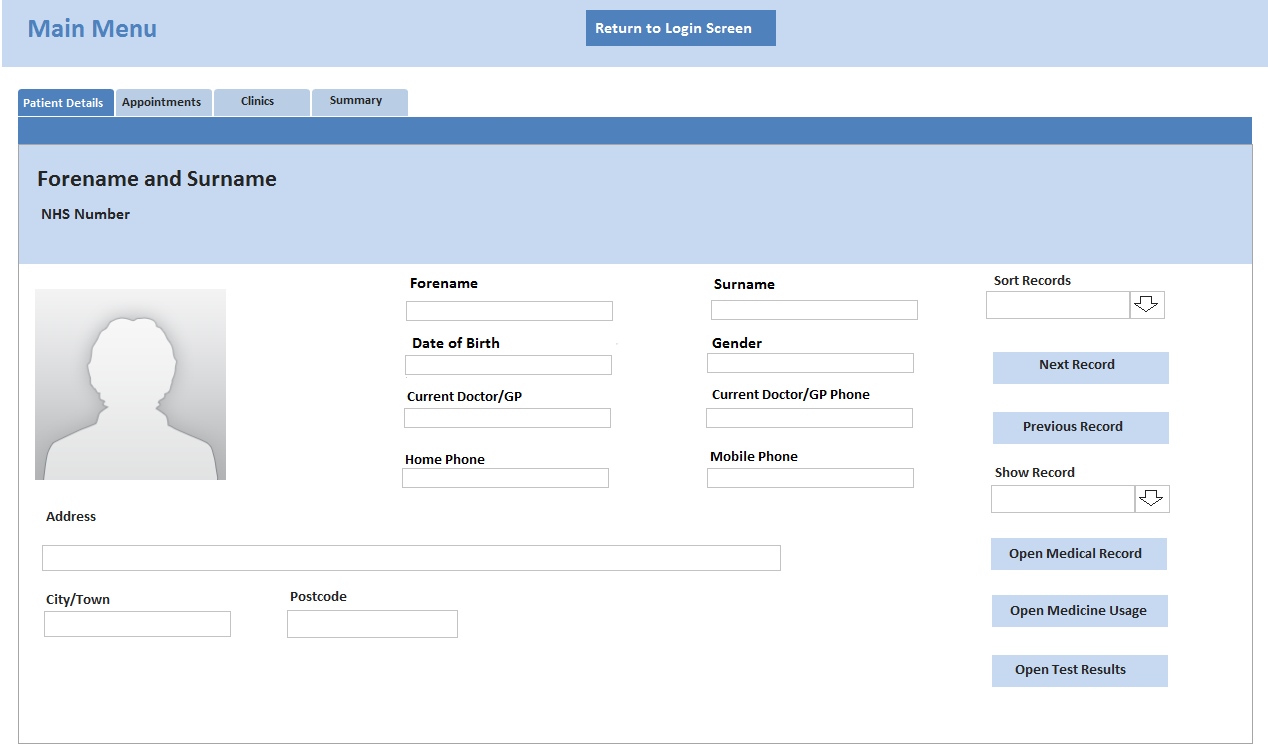
All the buttons on the main form should have a light blue colour where as those in the headers should have a darker blue such that the colour of the buttons in the header does not mesh in with the background colour of the headers themselves. This style is consistent throughout the system.

For the main menu in the navigation/tab system, I opted to have the tab that was in focus change to a dark blue colour and white text to show that it was the tab currently being selected. The white text was used because it was easier to read the white text than the black text on such a dark blue shade.

User Interface – Sample of Planned Data Capture

## F1 – Main Menu

Form Name: Main Menu Form Ref No: F1



I designed a combo box for “Show Record” as not only could I see a drop down list of all the patients’ names, but I could also type in their name if I knew it and it would bring me to that person’s record. The combo box would drop down the patient’s forename, surname and NHS Number such as: Katy Smith 02435082

I chose a combo box for “Sort Records” as well as there would be a limited amount of option for which the records could be sorted by and so for ease of accessibility, I opted to use a combo box.

I designed my main menu as a tab system as it would create less of a cluster than ordinary buttons would do on a main menu screen.

This is because instead of putting on 20 buttons on one screen, I have put 4 tabs which then categorize the buttons and so you only see the buttons which are of use to the part of the system you are viewing

For example, the “Patients” tab would show all the buttons relevant to the form.

## F2 – Login Form

Form Name: Login Form Form Ref No: F2

User Interface - Sample of Planned Valid Output Designs

In this text box, some data would be entered such as

CHK4-HJKD-843H-A896L

The system would then check the security code after the login button is pressed

In the login box, some data would be entered such as: Bob

In the Password box, some data would be entered such as: 1234567

This would be hidden with the input mask of asterisks

The system would then check if the login name was valid in combination with its password after the login button is pressed

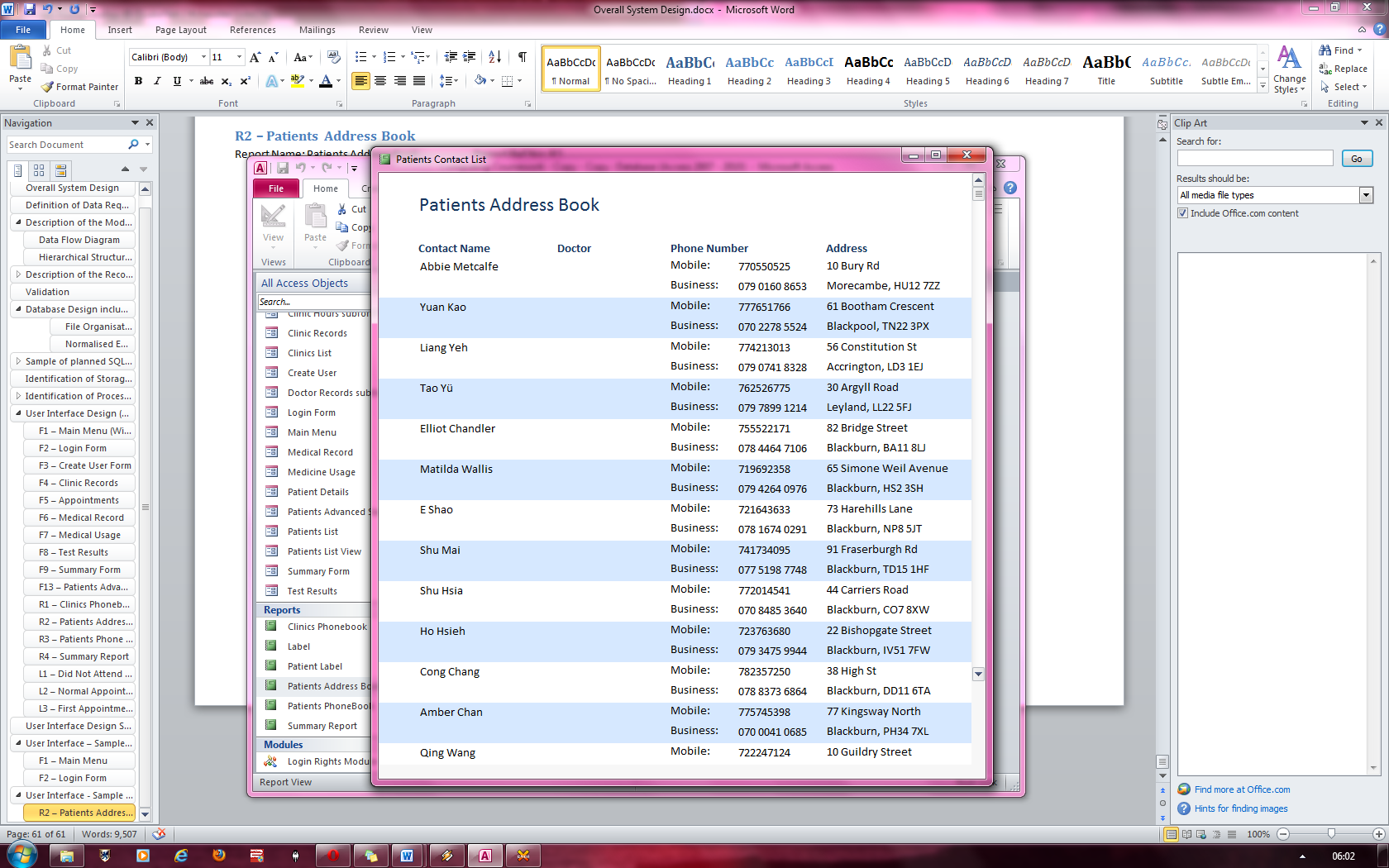
In the User Group combo box, the user would select one of the User Group’s listed such as:

Admin or Doctor

I would use a combo box as there would be only a few limited options for the user group list.

## R2 – Patients Address Book

Report Name: Patients Address Book Report Ref No: R2



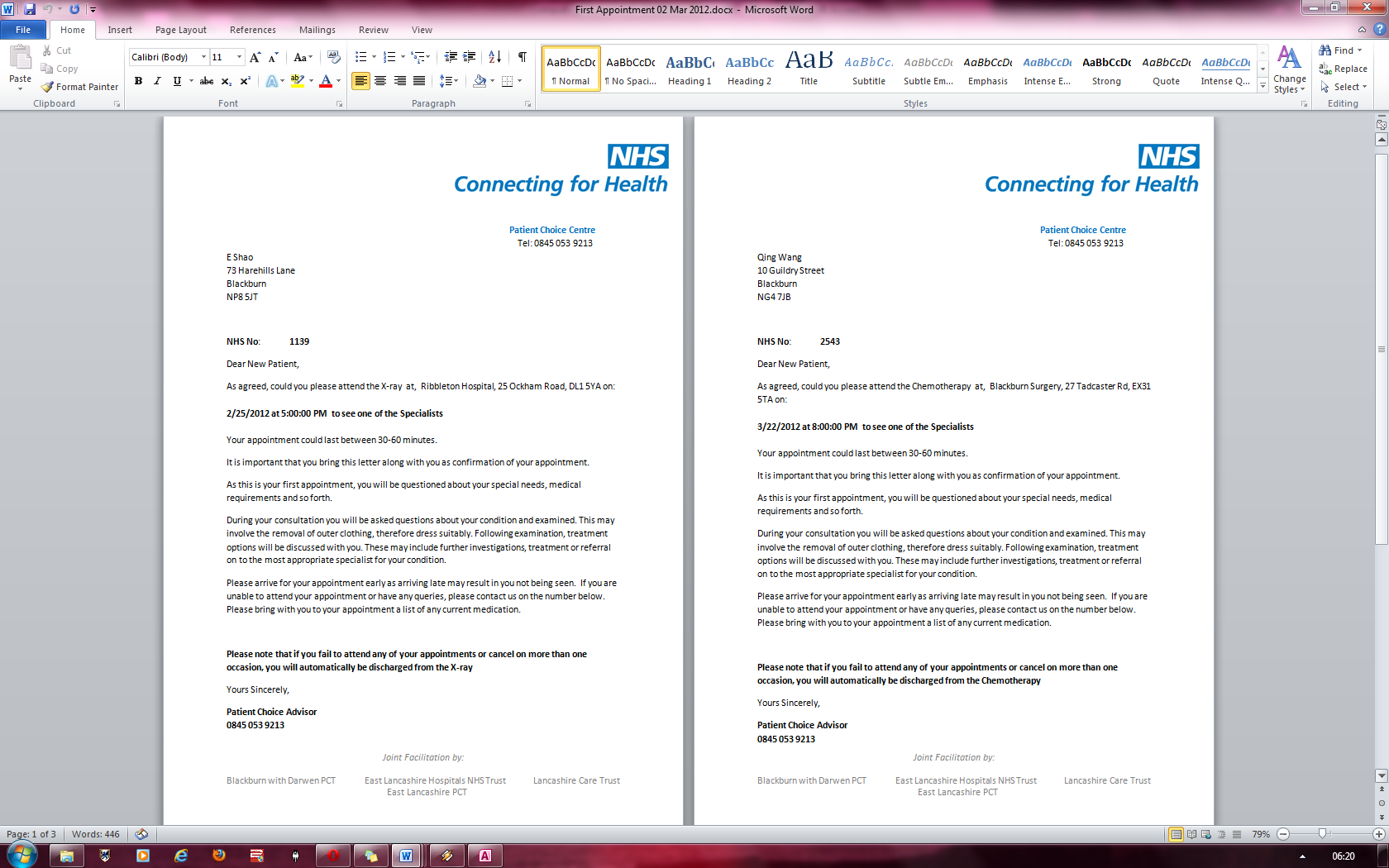
The black text is easily readable off the light blue segments on the report.

The use of the alternating blue and white segments adds to the house style of the system and keeps it consistant.

All the data easily fits into the space

## L3 – First Appointment Letter

Letter Name: First Letter Ref No: L3



Sample text inputted into the letter

Blue and white logo consistent with house style of the system.

Measures Planned for Security and Integrity of Data

Since the software I am designing will hold confidential information about patients, the data must follow the data protection act.

My system already includes a login feature which requires you to login before accessing any other forms, reports or tables and with the traditional login system, there comes extra security in the form of a security code which must be entered to gain access to your account.

My system will reside on the hard drive mainly but the system will be backed up onto a CD-R. The database size, although initially small, can grow to a very large amount and so the data could be compressed through the use of software compression utilities such as WinRAR.

The backup will have to be made every day and can be made through a built in backup procedure. The backups will also be kept offsite to avoid losing the data and it will need to be encrypted so if someone managed to get a hold on one of the backup discs, they wouldn’t be able to extract any information regarding the patients. I would recommend a senior administrator to back up the changed data daily and then store it in a secure place. The backups would be kept in the password protected backup directory and all the backups should have their filenames named after the date and time that the backups were made.

Backups should be kept for as long as they are relevant and mostly, they will only stay relevant for a month as the database is constantly being updated with new information.

The data will also be hard to corrupt as there are many validation checks for each data entered. As I am using Microsoft access, the system itself allows you to give each field a data type which already initiates a type check whenever something is written into that field. There are also range checks for dates of birth for example and presence checks for some of the most important data. There are also length checks for specific pieces of information such as phone numbers and names which are set up using the field size property and the validation rule property in the tables of data.

Description of measures planned for System Security

My software has rigorous amounts of security as it must protect very confidential data. Therefore, all passwords must be 6 characters or. There is also a security code which you must enter before you login for additional layers of protection.

Each user when logging on must also choose their user group which gives them different access rights to different parts of the program.

However, I would also recommend also keeping the database in a secure location for additional layers of security.

Overall Test Strategy

For my testing strategy, I will be starting by testing each and every individual component of my system. For this, I will be working with three types of data: normal, limit and exceptional data. In addition, I will be testing the maintenance of all the forms tables and reports in the system.

## Unit testing

The main components of my system which I will be testing are: the login system, the patient details form/tables, the appointment form/tables and the clinic/form/tables.

## Login Section

For the login section, I will be testing the security of it which is integral for my system.

For the Security code, I will be testing erroneous data such as combinations of characters that should not work with the system ie. ^%^”-£(90^-!)£()!

For the login names I will be testing, erroneous data such as data that goes over the 30 character limit

For the password, I will be testing it with erroneous data such as all letter passwords or all number passwords

## Patient Details

For the patient details, I will be testing erroneous data such as numbers for the names instead of just characters; extreme data such as a name 50 characters long as well as range checks for the date of birth etc

## Appointments

For the appointments, I will be testing if I can overwrite a date that has already be taken (which I should not be able to). Other tests include testing if checking both checkboxes on the appointment forms yields the result of 2 letters appearing or with it appearing an error message.

## Clinics

For the clinics, I will be testing if it allows me to have the same reference code at a time which would be erroneous. I will also be testing its functionality with adding new records to see if I can enter extreme or erroneous data into the fields.

## Integration Testing

Integration testing is the bringing together of the different units once they have been fully tested and then having them produce the desired results. This is what I will be doing once the unit tests are complete.

1. The tests will need to confirm that there is a link between the main form and all the individual forms.
2. If procedures carried out from the available modules are working correctly and produce the correct results.
3. Data which enters the system is correctly translated into meaningful information which can then produce reports or letters or any other output form.

## Acceptance Testing / Beta Testing

Once both unit testing and integration testing is complete, I will be using the data provided by the end user(s) and then checking if the desired results are produced. If they are not, I will go back and make the correct adjustments to the system such that it does produce the results expected of it.

The end user(s) will be able to make suggestions and/or requests if they wish to do so.