Design

Contents

[Output 2](#_Toc348346116)

[Input 3](#_Toc348346117)

[Files /Data structures and methods of access 4](#_Toc348346118)

[Data Structures 4](#_Toc348346119)

[Student table 4](#_Toc348346120)

[Teacher table 5](#_Toc348346121)

[Stud av table 5](#_Toc348346122)

[Staff av table 6](#_Toc348346123)

[Day table 7](#_Toc348346124)

[Appointments table 7](#_Toc348346125)

[Lesson table 8](#_Toc348346126)

[Methods of access 8](#_Toc348346127)

[Validation 9](#_Toc348346128)

[Processing stages 10](#_Toc348346129)

[Evaluation Criteria 11](#_Toc348346130)

# Output

the system will display to students blocks they have declared themselves to be available for on the input screen and up until the sorting algorithm has been run they can change and see their choices.

The system will display in much the same way for teachers who will be inputing their available times in the same way.

For admin it will also output the data of who has been offered what appointments, they will be able to, using the results and the see results form, see the appointments given out by teacher and student.

The system will also output the data of appointments to the user who it involves via the send results button on the admin form which will send out a mass mail merge involving the students.

# Input

Data will be collected from the school data bases as to the lessons and the student and staff data. It will be exported from the schools csv files into new correctly formatted csv files by the program.

Data on the availability of students and staff will be collected by them logging onto the program, it will not require a password as it will check who’s computer area the program Is being opened in and log them on automatically. Once they log onto the program it will send them to a list of all the available blocks of time where they will use checked lists to tick all of the blocks they are availale for and this will be input into their respective availability records.

# Form design

## frmStart

|  |  |
| --- | --- |
|  | btnEnter  btnAdmin |
| First form opened  when opened – checks the network profile and from that derives weather they are a teacher or a student and if they are a member of staff weather they are an admin the form then acts acordingly  btnEnter – sends the user to the student form if the user is a student and the staff av form if they are a member of staff.  btnAdmin – only visible if the memer staff is also admin, when clicked opens the Admin form. | |

## frmMain

|  |  |
| --- | --- |
|  | btnImport  btnSettings  btnAlgorithm  btnSeeResults  btnSendResults  btnBack |
| btnImport – a button that when clicked will import data from the colleges csv files that contain student data and teacher data into the csv’s that will be used for the consultation evening.  btnsettings – a button that when clicked will open up the settings form  btnAlgorithm – a button that will execute the algorithm that will organise the appointments  btnSeeResults – a button that will open up the results form  btnSendResults – a button that will seen the lists of appointmetns to each student and teacher via college e-mail  btnBack – a button that will open the Start form | |

## frmStudent

|  |  |
| --- | --- |
|  | lblName  chklstAvailable |
| lblName – a lable that will cotain the name of the user  chklstAvailable – a checked list box in which the student will check the boxes of the bocks of time he will be available. | |

## frmStaff

|  |  |
| --- | --- |
|  | lblName  chklstAvailabilty |
| lblName – a lable that will cotain the name of the user  chklstAvailable – a checked list box in which the member of staff will check the boxes of the bocks of time he will be available. | |

## frmResults

|  |  |
| --- | --- |
|  | lstStudents  lstStaff  btnBack |
| lstStudents – a list box that contains the names of all the students, when a students name is clicked on it will open up the see results form with the data on the students appointments on it.  lstStaff – a list box that contains the names of all the staff, when a staff member is clicked on it will open up the see results form with the data on the member of staff’s appointments on it  btnBack – closes the form and opens up the main form again. | |

## frmSeeResults

|  |  |
| --- | --- |
|  | lblTitle  lstAppointments  btnBack |
| lblTitle – a lable that contains the name of the student or member of staf that is having their appointments looked at  lstAppointments – a list box that contains the appointments that the student or the member of staff has including the teacher , the student and the time.  btnBack – a button that when clicked closes the see results form and opens the results form. | |

# Files /Data structures and methods of access

## Data Structures

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student table (StudNo, StudID, Forename, Surname, Year, Form, Email, PhoneNO) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | | coments | | Example data |
| Stud NO. | Short | 2 | 0000 |  | Primary key | |  | | 0011 |
| Stud ID. | Integer | 4 | 000000 |  | To store the id of the student for outputs | |  | | 112544 |
| Forename | String | 60 | xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx |  | To store the name of the student for outputs | |  | | Jason |
| Surname | String | 60 | xxxxxxxxxx xxxxxxxxxx xxxxxxxxxx |  | To store the name of the student for outputs | |  | | Scott |
| Year | byte | 1 | 00 |  | Purpose to help divide between the years which will have separate consultation evenings | |  | | 13 |
|  | | | | | | | | | |
| Teacher table (StaffNO, Forename, Surname, Email, Admin) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | coments | | Example data | |
| Staff NO | short | 2 | 000 |  | Primary key |  | | 032 | |
| Forename | string | 60 | xxxxxxxxxx  xxxxxxxxxx  xxxxxxxxxx |  | To store the teachers name for outputs |  | | Jane | |
| Surname | string | 60 | xxxxxxxxxx  xxxxxxxxxx  xxxxxxxxxx |  | To store the teachers name for outputs |  | | Law | |
| Admin | Boolean | 2 | True/false |  | To decide weather the teacher will have access to the admin area |  | | true | |
|  | | | | | | | | | |
| Stud av table (Appointment, *StudNO*, *Day*, Available) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | coments | | Example data | |
| appointment | byte | 1 | 000 | Must be in staff av table | Primary key |  | | 12 | |
| Stud NO. | Short | 2 | 0000 |  | Primary key / Foreign key |  | | 0011 | |
| Day | Byte | 1 | 00 |  | Primary key / Foreign key |  | | 1 | |
| Block | Byte | 1 | 0 |  | To store which block the student is available for |  | | 1 | |
| Available | Boolean | 2 | True / False |  | to store whether the student is available |  | | True | |
|  | | | | | | | | | |
| Staff av table (Appointment, *StaffNO*, *Day*, available) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | coments | | Example data | |
| Appointment | Byte | 1 | 000 | Must be in stud av table | Primary key |  | | 001 | |
| Staff no. | Short | 2 | 000 |  | Primary / Foreign key |  | | 0012 | |
| day | Byte | 1 | 00 |  | Primary / foreign key |  | | 01 | |
| Block | Byte | 1 | 0 |  | To store which block the staff member is available for |  | | 1 | |
| available | Boolean | 2 | True/False |  | Store weather or not they are able to do that appointment |  | | false | |
|  | | | | | | | | | |
| Day table (Day, Start, End, length) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | coments | | Example data | |
| Day | byte | 1 | 00 |  | Primary key |  | | 03 | |
| Start | Date |  | hh/mm |  | Records the start time |  | | 0600 | |
| End | Date |  | hh/mm | Must be atleast an hour and a half after start | Records the end time |  | | 0830 | |
| Length | byte | 1 | 5|10 | It will be from a selection of 5 to 10 mins | It is used to work out how many appointments. If 5 1 hour blocks and if 10 an hour and a half | There will be a drop down | | 10 | |
|  | | | | | | | | | |
| Appointments table (*Day*, *LessonNO*, appointment) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | coments | | Example data | |
| Day | Byte | 1 | 00 |  | Primary key / Foreign key |  | | 01 | |
| Lesson NO. | Short | 2 | 0000 |  | Primary key / Foreign key |  | | 0456 | |
| appointment | byte | 2 | 0000 |  | Stores the time as the slot |  | | 0023 | |
|  | | | | | | | | | |
| Lesson table (LessonNO, *StudNO, StaffNO*) | | | | | | | | | |
| Field name | Data type | Length (bytes) | Format | Validation | Purpose | comments | | Example data | |
| Lesson NO. | Short | 2 | 0000 |  | Primary key |  | | 0124 | |
| Stud NO. | Short | 2 | 0000 |  | Foreign key |  | | 0011 | |
| Staff NO. | Short | 2 | 0000 |  | Foreign key |  | | 024 | |
|  | | | | | | | | | |

## Methods of access

# Validation

When btnBack is clicked on the settings form it will check whether there has been a value for both the start and end days have been entered e.g.

For counter = 1 to NDays

If day(counter).start = -1 then

set = false

Else if day(counter).end = -1 then

set = false

end if

if set = false then

msgbox(“you have to enter start and end time for all the days”)

exit sub

end if

next

# Processing stages

# Evaluation Criteria

The project must

* Produce an easily printable list of appointments for each student and teacher
* Must score a unanimous vote as simple to use in the beta test.
* Must never generate two consecutive appointments for a student.
* Must handle multiple lessons with multiple teachers
* Must be multi user multi access
* Must allow users to log on just by their network area