Exemplar Candidate Work

Part 1 of 2

GCE in Applied ICT

OCR Advanced GCE in Applied ICT: H715

Unit G057: Database design

Contents

Introduction

This exemplar material serves as a general guide. It provides the following benefits to a teacher:

- Gives teachers an appreciation of the variety of work that can be produced for this unit
- Shows how the mark scheme has been applied by a senior assessor

It is important to make the point that the teacher support materials play a secondary role to the Specification itself. The Specification is the document on which assessment is based and specifies what content and skills need to be covered in delivering the course. At all times, therefore, this teacher support should be read in conjunction with the Specification. If clarification on a particular point is sought then that clarification should be found in the Specification itself.

Moderator's Commentary: G057 Database design

Total mark for portfolio: 46 (Max. 50)

This portfolio illustrates the work of a candidate who has analysed and designed a relational database with 4 tables. The banner requires at least three tables. This is for a booking system for the school's library. The candidate has provided evidence of normalisation to 3rd normal form as well as providing a user interface and user documentation to enable easy use of the database. The database contains forms for input of data and different reports used to inform the librarian of, for example, overdue books, as well as reports that can be used by the librarian to help library users to find, for example, books by a particular author. The candidate has also provided technical documentation allowing amendments or additions to be made to the database by a technically minded person. They have provided evidence of testing their database and finally evaluated their solution to the problem as well as their own performance in creating the database. The work is well structured, but the main tasks have not been clearly defined. The use of a contents page makes it easier to follow and assess. The pages have been numbered manually although not all of these numbers are visible on the scanned copy.

Task a

The candidate has included three pages outlining how and where databases are used. Whilst this might be useful for teaching and learning, it is not necessary, or advisable, to include these notes for the task or the unit as a whole.

The candidate goes on to analyse and explains, in detail, the current (manual) system used in the school library, explaining the problems that occur. The candidate then explains how the new database will allow multiple users to interact with the system at the same time, what data is to be input and that the output will mainly be in the form of reports that aid the user to identify, for example, who has books on loan and which books are on loan.

Also included in task a, is an explanation of the planned database structure, which can be seen on pages15 – 16 in the form of a data dictionary of the four tables. This includes the use of text, date/time, number, currency and logical field types. Also included in these tables are details of the formats required for the data and any validation required to ensure correct data entry. There are also details of which queries are to be run together with a reason why they need to be made available. The candidate also produced a similar table for the reports to be created.

The plans for forms, reports and the user's menu and sub-menu screens are clearly laid out and easy to understand and follow. These can be seen on pages 20 – 25 and 31 – 33.

The analysis and design section is substantial and very thorough. Technical language is used throughout. It includes all the necessary content and is a good example of the planning required for forms, menus and reports. Maximum marks can be fully justified for this task.

Mark Band 3 **Mark Awarded: 6 (Max. 6)**

Task b

The candidate has included task b in their analysis and design work. The work can be found on pages 10 – 14. Entity relationship diagrams (ERDs) are provided and explained in detail as well as the process of normalisation to third normal form (3NF). The explanation includes the use of primary and foreign keys and shows a clear understanding of the process by the candidate. The candidate has identified all the entities required, ensured that all attributes are atomic and defined relationships between entities. All many-to-many relationships have been resolved. There is also a relationship diagram explaining the 1-to-many relationships. Maximum marks can be justified for this task.

Mark Band 3 **Mark Awarded: 6 (Max. 6)**

Task c

The candidate has provided narrative evidence of the implementation of their database. This, whilst easy to follow and understand, is not a requirement of the unit; however, it does aid the assessor in understanding the screenshots. The screen shots, which should enable the moderator/assessor to see evidence of the way the tables and fields have been constructed, are too small to be of much assistance and further scanning has made some of them almost useless. It would be more beneficial if these screenshots had been cropped from the right and then enlarged to make them easily readable and able to show the validation and/or input masks defined for the fields. The candidate appears to have set all data types to the most appropriate for the task and also appears to have used validation in the date of loan field, where the due back date is +7 days after the date of loan. An input mask has been used for the ISBN of each book and validation rules for the year group of the student and for their form room. Although some of these validation rules do not stop errors totally, they should reduce the chances of errors occurring.

The problem with the readability of screenshots continues in the section on the production of forms and the user interface. The forms created are for adding/deleting/amending a student, or a book as well as a form that includes sub-forms for student loans. The forms all include buttons to add, save, delete or print the record seen as well as navigation buttons to move through the records. The field lengths for all data that could be input are identical lengths. This should not be the case. For example, on the book form, the price field should be a lot shorter than the book title or author fields. Only one form includes the use of sub-forms. This form is well laid out and includes all the relevant details. It too unfortunately, includes all fields of the same length and apart from the labelled buttons to add, save, delete or print the record; there are no other instructions/ user aids.

Because the portfolio is not split into tasks, the assessor must then find the evidence for setting up a menu system. This is on pages 59 – 63. A main menu includes buttons to all the forms as well as a button to link the main menu to a sub-menu for reports as well as an “About us” and an exit button. The candidate appears to have protected the database by disabling the navigation pane.

There is sufficient evidence for mark band 3 but, for maximum marks, the field widths on the forms would need to be varied to match the data to be entered and instructions given as to what data should be entered into some of the fields on the forms.

Mark Band 3 **Mark Awarded: 7 (Max. 9)**

Task d

The candidate has provided evidence of creating several different queries which include the use of Now(), parameter queries and complex queries using ‘and’. The queries are the sort you would expect to see a librarian using and follow those outlined in the analysis and design section of the portfolio. The queries are difficult to read due to the scanning of the work but the explanations given by the candidate make them easier to follow and understand. There are 8 queries in total. Evidence of the creation of the queries can be seen on pages 46-49 whilst evidence of the output from them can be seen on pages 70-87. These queries were then converted into reports to make them easier to understand and follow for the user. All reports include suitable headers and footers including the date and time of running/printing. Some fields are displayed in the report header, for example the genre as well as the start and end dates of a query for ‘Books of a particular genre’. One report, “All overdue books for a form”, includes the total number of overdue books as well as the total price of the books. These have used both count and sum in their calculation. However, there is no overt evidence that grouping has been used in any of the reports.

The reports are laid out in different ways, however, they are all easy to follow with the inclusion of separator lines and colour. The lack of evidence of grouping would suggest a mark in mark band 2 but the reports do make correct and effective use of queries, so the lower mark in mark band 3 is best fit.

Mark Band 3 **Mark Awarded: 5 (Max. 6)**

Task e

Task e requires the production of two sets of documentation: a user guide and technical documentation. They should not be one document. This candidate has only provided one document which includes all the required user guidance but also technical information regarding backing up of data and acceptable data types. All other technical documentation is found throughout the portfolio.

Most of the implementation section is regarded as technical documentation. It includes information about how to recreate and maintain the database on pages 35-63. The candidate has clearly defined the database structure and shown the data relationships on pages 13-14. Pages 107- 116 provide a printout of the data dictionary as provide by the software. This is not required and includes a great deal of unnecessary information. It is preferred that a candidate produces their own data dictionary with only the information required contained within it. However, these pages are annotated to explain a little of the content. The data dictionary provides evidence of validation routines and input masks. The candidate includes printouts of all queries and reports on pages 69-102. There is evidence that the database is locked to users by means of the menu but no help is given if any changes have to be made to the underlying database by a technical person.

The candidate’s user guide on pages 171- 194 includes clear screenshots to aid understanding by the user. Instructions are given as to how to open the database and use the menu screens. An explanation of the navigation buttons on each form is given as well as information on how to enter data on the student loans form. Information is also given on how to add or delete records as well as saving or printing the record.

Overall, a mark in mark band 3 is appropriate however, no information is given as to how to amend a record on any of the forms and some information is of a very technical nature – more appropriate

to the technical documentation preventing anything above the bottom mark. For a higher mark the candidate would need to ensure that the user guide included all required instructions with no technical detail. The technical documentation should also be presented as a separate stand-alone document, rather than throughout the portfolio.

Mark Band 3 **Mark Awarded: 6 (Max. 8)**

Task f

A test table and evidence of all tests have been provided. The test table tests validation, the running of queries and reports, forms, buttons on menus and forms as well as the opening and closing of the database. Abnormal data is tested, for example a number is entered instead of a name in test 13, and extreme data is tested when years 13 and 7 are entered where 7 &13 are the extremes allowed. A variety of other abnormal and extreme data is used to test all validation rules. Evidence includes screenshots and an explanation as to why that specific data was used. Some screenshots are difficult to read after scanning.

All reports and queries are both tested to ensure that they run correctly and provide the ‘correct’ or expected data. Some errors were found, but the candidate then goes on to explain how these were corrected and provides evidence of retesting to show that the problems have now been solved.

The candidate comments on how well each element tested has operated and identifies some aspects that meet the specification. These comments are made after each test and are thorough although there could also be a final comment made after all testing is complete to bring together all the comments. However, there is sufficient evidence for maximum marks to be awarded.

Mark Band 3 **Mark Awarded: 9 (Max. 9)**

Task g

The candidate has provided an evaluation that includes a full critical analysis of the final solution linking it to all parts of the initial user requirements. The candidate has acquired feedback from a potential user to ensure accurate evidence of meeting user requirements as well as using their own knowledge. Strengths and weaknesses are given for the database content and layout, user interface, available reports and user guide. The candidate outlines ways to improve the database in the future,

The candidate’s own performance is also analysed from the analysis and design stage through to the testing stage of the process. At each stage the candidate gives strengths and weaknesses and provides ways to improve the process in the future. A full critical analysis is given. Maximum marks can be awarded.

Mark Band 3 **Mark Awarded: 6 (Max. 6)**

Total mark for portfolio: 45 (Max. 50)

G057 Assessment Evidence Grid

Unit G057: Database design

What candidates need to do:

Candidates need to produce: a relational database to meet a given specification requiring at least **three** related tables supported by design and analysis notes, technical and user documentation and an evaluation of the database produced.

Evidence needs to include:

a: [AO3] analysis and design notes [**6**];

b: [AO3] normalisation of the data model to 3rd normal form (NF) with documentation [**6**]

c: [AO1] a user interface including data input forms and methods of obtaining output [**9**];

d: [AO1] a working relational database [**6**];

e: [AO2] user and technical documentation [**8**];

f: [AO4] test plans and the results of the testing of the database [**9**];

g: [AO4] evaluation of the effectiveness of their solution and performance [**6**].

How the candidate will be assessed:

Task	Assessment Objective	Mark Band 1	Mark Band 2	Mark Band 3	Mark Awarded
a	**AO3**	The candidate produces analysis and design notes that present the initial data model **[0 1 2]**	The candidate produces analysis and design notes that use technical language fluently, and which includes the design of the user interface, and screen and printed reports; **[3 4]**	The candidate produces full and complete analysis and design notes that use technical language fluently and correctly, and which includes the design of the user interface, screen and printed reports. **[5 6]**	**6/6**
b	**AO3**	The candidate shows, in a graphical form, the normalisation of the data model to 1st normal form; The candidate produces incomplete documentation that shows some of the entities, attributes, keys, relationships and internally-generated or processed data; **[0 1 2]**	The candidate shows the process of normalisation of the data model to 2nd normal form; The candidate produces an ERD in 2nd NF and documentation that shows all entities, attributes, keys, relationships and internally-generated or processed data; **[3 4]**	The candidate shows the process of the normalisation of the data model to 3rd normal form; The candidate produces an ERD in 3rd NF and produce complete and detailed documentation that shows all entities, attributes, keys, relationships and internally-generated or processed data. **[5 6]**	**6/6**
c	**AO1**	The candidate produces a user interface and suitable and correct data input forms and provide straightforward means of obtaining output; **[0 1 2 3]**	The candidate makes effective use of validation and produces a user-friendly, well laid out user interface, and data input forms with title labels, field names, set widths, pull down lists and instructions, as appropriate; **[4 5 6]**	The candidate produces a fully-customised user interface that hides the underlying database from the user and provides input forms that allow data entry into multiple tables. **[7 8 9]**	**7/9**
d	**AO1**	The candidate produces a working relational database that allows the user to append, delete and edit data, run queries and print reports; **[0 1 2]**	The candidate produces a working relational database; The candidate creates reports that make use of queries, using data from more than one related table, and include grouping or arithmetic formulae; **[3 4]**	The candidate produces a working relational database; The candidate creates reports that make correct and effective use of queries, using data from more than one related table, grouping and arithmetic formulae. **[5 6]**	**5/6**

Unit G057: Database design (continued)					
Task	**Assessment Objective**	**Mark Band 1**	**Mark Band 2**	**Mark Band 3**	**Mark Awarded**
e	**AO2**	The candidate produces a user guide that enables a novice user to make effective use of the database; The candidate produces technical documentation that defines clearly and accurately the database structure, data relationships and the range of acceptable data; The candidate includes example output from queries and reports; **[0 1 2]**	The candidate makes good use of graphic images and use annotated screen prints to create effective user instructions for the database. The candidate produces technical documentation that defines clearly and accurately the database structure and data relationships; The candidate includes a data dictionary including the range of acceptable data and associated validation routines; The candidate includes example output from queries and reports; **[3 4 5]**	The candidate makes good use of graphic images and use annotated screen prints to create effective user instructions for the database; The candidate creates high-quality technical documentation that would enable someone else to recreate or maintain the database; The candidate define clearly and accurately the database structure and data relationships in the technical documentation; The candidate includes a data dictionary including the range of acceptable data and associated validation routines; The candidate includes example output from queries and reports. **[6 7 8]**	**6/8**
f	**AO4**	The candidate plans and carries out basic test procedures to demonstrate that the database meets the specification, including ensuring that the user can append, delete and edit data successfully, and that the queries and reports generate the expected outcomes; Using the results of their testing the candidate comments on the operation of the database in relation to user needs; **[0 1 2 3]**	The candidate designs and carries out systematic test procedures, covering most aspects of the database; Using the results of their testing the candidate comments on the operation of the database, and how well it meets the specification; **[4 5 6]**	The candidate designs and carries out systematic and comprehensive test procedures covering all aspects of the database, including rejection of data outside the acceptable range; Using the results of their testing the candidate provide a critical analysis of the operation of their database solution and how well it meets the specification. **[7 8 9]**	**9/9**
g	**AO4**	The candidate comments on the effectiveness of the final solution identifying how well it meets the defined user requirements, with some identification of how the solution could be improved; The candidate evaluates aspects of their actions and role in solving the problem and identify areas for improvement; The candidate's report may contain errors in spelling, punctuation and grammar; **[0 1 2]**	The candidate provides an analysis of the final solution identifying how well it meets the defined user requirements and its strengths and weaknesses in order to identify how the solution could be improved; The candidate evaluates their own performance by identifying strengths and weaknesses, with some suggestions for improvement to the overall process; The candidate's report contains few errors in spelling, punctuation and grammar; **[3 4]**	The candidate provides a full critical analysis of the final solution identifying how well it meets the defined user requirements, taking into account user feedback in order to identify how the work could be improved in the future; The candidate evaluates their own performance by identifying strengths and weaknesses, and uses this analysis to show how they will address these issues to be more effective in the future; The candidate's report is consistently well-structured and there will be few, if any errors in spelling, punctuation and grammar. **[5 6]**	**6/6**
				Total mark awarded:	**45/50**

Candidate’s work

Contents

How and where are databases used?

I am now going to complete the below table to look at how databases could be used across a number of different scenarios, and analyse how the user will interact with the database, what tables will be present in the database, what data is inputted into the system and what data is output in the system in its relevant form.

Area	How does the user interact with the database? (what why how)	What data is inputted?	What data is outputted?	In what form, and how, is the output generated?	What tables are in the database and how are they linked together?
Health	In the health sector such as the NHS, databases will be used for things such as asset management and patient records. This database is held on central servers and accessed by client computers, likely to be in each department. It is also possible for similar systems using technologies such as interactive touch screens to be used for other purposes, e.g. the possibility of patients booking themselves in for an appointment by a computer. This is useful as it means that appointments can easily be made and patient details can easily be retrieved, without the need for an employee to be present, therefore saving time and money due to ensuring that all data can be retrieved meaning it never duplicated.	The data inputted will obviously depend on the purpose of the database, for example an asset management database will contain a primary key (for example, asset ID number) and a number of other fields such as department, type of asset, cost when new and who may be using it. Data put into a patient database is likely to be name, date of birth, home address, other personal details and records of any treatment they have had. Again each patient will have his/her own ID for a primary key in the table. To retrieve this information the data needed to be input into the system is likely to be an asset ID number, and a patient ID number or name.	Data that will be output will be all of the fields in that table and any data in other linked tables within that database. This means, if for example a patient is searched for by their ID number, all of there personal details will be shown on screen. Other information likely to be shown is any records of appointments and whether they attended them or not, (may be on a linked appointment table) as well as doctors or nurses they have seen, medial examinations or treatment given along with any other advice or prescriptions given to the patient. This is.	The data that will be generated is likely to be presented as a report on screen, this is likely to be for one persons details in the case of health records as a query will be run using their name or ID, this report will be shown on screen however it is also likely that any reports may be printed off the system to be used by any employee. Things such as prescriptions will also be kept on a database with records for example of what people have had, how many times they have had it and if they have paid for it. This again is useful as this data is going to be outputted on a prescription form that the patient may take in order to gain their medicine.	There is likely to be many tables used in a health database, for example a appointment database will be linked to tables that contain details of each doctor, a table that contains for details on each patient and their personal information, a table that contains information on the department or room where treatments will take place, a table containing prescription information and a table containing information on the times appointments are available or have been made. This means the main table can be linked to all of these tables by using foreign keys (Primary keys from another table) meaning information only has to be entered once, for example once the patient details are bought up on the database, details of the relevant department or doctor can also be seen.
Hotels	A hotel is likely to use a database on a computer system in order to manage their bookings, by doing so they will be able to keep information on current and past customers, staff and hotel rooms, meaning that they can ensure that the correct people are put in the correct rooms at the right time and that staff are also timetabled in correctly.	Data that is inputted will be details about the room, details about the customer and personal details of staff. These will then be used on another database in order to match customers with rooms and to ensure staff are working when needed, the data will be input when somebody makes a booking in order to show that the	Firstly the data outputted will be the room bookings and staff timetable shown on screen, other information that may be outputted may be automated lists of rooms that will be needed to be cleaned as they are occupied, in order for the cleaners to spend the least amount of time working as they will know which rooms need to be cleaned and which do not.	The main part of the database is likely to be output on the screen; however documents such as invoices and receipts are going to be printed out from the system in order to give customers a paper-based copy. Again, details of customers may also be printed out or details of occupied rooms or number of guests may be outputted in paper form in order to give to, for example, cleaning staff or catering staff, so that they know	Tables likely to be used in the database are ones that contain customer information, staff information and a table relating to rooms containing information such as size, number of beds and whether it is en suite, along with any other special facilities of the room. These will all be linked to a central database so that the hotel can organise a table showing what customer is in a particular room

	This will ensure that the hotel runs as smoothly as possible and staff are not working when not required and helps ensure that room bookings are correct, helping customer satisfaction.	room is then occupied, hopefully meaning that the hotel will not over-book its rooms. Payment details may also be entered onto the system	Other data outputted from the system may be price information or things such as invoices or receipts to be sent to customers. Customer details may also be outputted onto a list for mailing purposes as information about previous customers may help the company advertise in order to persuade customers to come back again	how many people they expect to be cooking food for, this will again increase the efficiency of the hotel.	along with what staff members may be associated with that booking, e.g. Different cleaners may work over different areas of the hotel.
Banks	Databases in banks are going to be used to control peoples personal information, control different bank account types and control how much money people have in each of these accounts. Data may be inputted in a number of different ways, for example if people go to one of the clerks, they will be using a computer to input data, however if they were to pay in, for example, a check, they can use a machine which will scan the check and automatically find its value and who it is coming from, and simply ask the user to confirm that the details are correct. Once the check has cleared, the funds will enter the database and the values will be added to the relevant account.	The data inputted onto the system will depend on what the user requires to do, for example, if the intention was to open a new account, the customers personal information will have to be inputted, along with the type of account required and the funds to be added to the account when opened.	The main output data from the system is either a receipt for payments or withdrawals and monthly or quarterly statements to be posted to people for each of their accounts. Other outputted data from machines such as ATMs will be things such as available balances and PIN number help.	Output data such as withdrawal or deposit receipts will be printed from the system and therefore will be paper-based, when machines are used for information or deposit/withdrawal purposes, account information may also be displayed onto the screen.	For a banking database system, their is likely to be a main account screen where each person's individual accounts can be retrieved, this is likely to be linked to other table in the database in order to not duplicate data, other tables are likely to be things such as Customer names and details and a table for account types and details E.G. Current interest rate.
Sales of goods	The sales of goods are likely to be linked to an EPOS (electronic point of sale) database. This database will be used to contain information on the products that are being sold such as price, possibly type or weight and other descriptive factors for the product, along with a stock count. The users of the database are going to be the sales staff for the shop who will use a computer system to access stock records. In order to sell products, a barcode	The data inputted into the system will be the barcode when scanned by a barcode reader, however sales staff may also key in a unique product code to retrieve details from the system. They may also search by make, model and price on the system in order to be given a list of products.	When this data is input into the system, the product name and price will be returned and totalled on the sales system, however if product details are searched for, a list will be returned from the database with the matching criteria after the query has been run.	The output generated is going to be on-screen for the sales staff in order to complete the sales process, if a product has been searched for the list of matching products returned is likely to be produced as a report on-screen however this may also be printed when necessary. Once the sales procedure is over, the system will also produce a receipt showing the products that have been purchased, again it is likely a unique ID is given to this receipt, such as a receipt number so that it can easily be found.	For the sales system there is again going to be a number of different tables that come together on the sales table. Firstly a table will be created for all of the products with assets such as product number and price. There is also likely to be a database that has the details of sales staff on and a table that contains customer names and addresses. These will be linked in the sales table when the sale is complete, showing the products that were sold to a person, who sold them and the customers details which may be used for

	scanner will also be used to retrieve the information about the product, therefore making it easier to sell products as they can be scanned in and the database will be accessed and the stock count updated correctly.				receipt purposes or advertising purposes.
Libraries	In a library, it is likely that interaction will again be by a computer and a barcode scanner. Most libraries will have their own barcodes on books that, similar to a retail shop, once the book is scanned in, one will be deducted from the stock count and the book is added as a loan to the relevant customers account.	The data input into the system when a book is taken out is likely to be the customers details (usually scanned in from a library card) and the barcode of the book(s) that wish to be taken by the customer.	The data output from the system will be the customer's account shown on screen to the librarian and there may also be an output form which contains details of the books which were taken and when they are due back to be given to the customer so that they remember to bring them back on time.	These outputs will be shown on screen to the librarian as any notes may be added to someone's account for example, to say that they are very often late returning. This along with other account information such as previous loans will be shown on – screen whilst a printed copy of the load and its dates may be given to the customer.	There is likely to be three main tables within the database, one of these will be the loans table that compiles all of the loans that are out/have been taken out. The two tables likely to be linked to this are going to be a table containing the information on all of the books available and their related quantities and a table detailing all of the customers and their library ID number (may be typed or scanned in from their card)
Police	The police are likely to have a number of different databases however the main one is likely to be a database that contains details on all recorded crimes and, if found, who has committed them. To interact with this system the user will usually use a computer to search for a crime or criminal, and the relevant result(s) will be returned back to them. This database is however also linked to other systems such as ANPR (automatic number plate recognition) cameras which automatically read number plates of cars, this will then check the database and alert the user, on screen if a car has driven past that, for example, has been linked to a crime by its registration plate. By doing this, the efficiency of the police service can be greatly improved as such systems have a very high accuracy and can help catch law-breakers.	Data inputted into the system is likely to be a criminal name or a crime number if somebody is searching through the database for a particular person or crime, however it would also be possible to search by other characteristics e.g. type of crime or area where it was committed. Again data may also be input into the system automatically by systems such ANPR cameras.	Different types of data are likely to be outputted in a number of different ways, firstly data from systems that are portable such as ANPR cameras are likely to alert a police officer by either making a sound and/or showing a notification on-screen. When searching the database, the user will also see information such as criminals or crimes committed on the screen of a computer system, however it is also possible that they may wish to print information or even images off this system to share with other people for informative or identification purposes, such data likely to be output is a criminals name, date of birth and address (if known) and crime details such as the type of crime, the date it was reported, who reported it and if there are any suspects.	The output generated can vary depending on what system is being used with the database. Firstly in an automatic system such as ANPR cameras, the police will be notified usually by an on-screen notification as part of the computer system they use, when using computer systems in order to search for data, the results will again be generated on screen, however it is likely that particular records, details or images will be able to be printed off in order to have a paper-based copy as well.	There is likely to be a variety of tables to be used within this database, with the main table likely to be of recorded crimes. Tables linking to this could include personal details of the police officers who are linked to the investigation, and a table of criminals who may be suspects or found guilty of the crime. Other tables that are related may include, for example, the types of crime or areas of where crimes occur.

Analysis

Problem identification

; currently has a logging system for the learning centre within the school. In order to keep records of who has taken and returned books, it uses a flat file database system in Microsoft Excel; they have one large spreadsheet which holds information such as student details, book information and the associated dates. (As shown below)

First Name	Last name	Year group	Form	Form Room	Form Tutor	ISBN No	Book name	Author	Publisher	Subject	Price	Date	Date due back	Returned?
						0-340-73805-7	Database Projects	Mott & Leeming	Hodder & Stoughton	ICT	£14.99	16/05/09	23/05/09	Yes
						0-654-55453-3	Business & Economics	Nuffield	Penguin	Business Studies	£12.49	20/05/09	27/05/09	No
						0-340-73805-7	Database Projects	Mott & Leeming	Hodder & Stoughton	ICT	£14.99	25/05/09	01/06/09	No

The current system means that every time a student wants to take a book out they ask one of the librarians who adds on an extra row onto the excel system, by asking the student for their details and copying the details of the book down, manually entering the data in each time. Once the book is returned, the librarian has to try and locate the record and change the 'returned?' field to 'yes' rather than no.

Using this system obviously causes a number of problems due to its lack of sophistication and accuracy, and is a good example of where a better quality database is really required. Firstly one of the main problems with the system is that data has to be manually typed in. Using an alternative database system it may be possible to bring up book details by scanning in a barcode meaning the books details do not have to be re-entered every time a book is taken out. Another disadvantage of the system is also that the personal details of the student taking out a book are provided by the student so that there is no proof required to show that the person is actually a student of that school.

Currently by storing details about users or books manually, no proofs are required and errors can be made. Currently when a new loan is recorded, a new row is added onto the spreadsheet, this will lead to duplicate records for people and books. The current system does not have a faster way of entering data (e.g. drop down box) or any way of ensuring that it is correct.

When a book is returned on the current system, one of the librarians will open the spreadsheet and locate the entry, once the

record has been located the field 'returned' will be changed to 'yes', however as there is a long list, it could be very hard to try and locate a current record, this could also lead to errors on the system if the wrong record is changed. Another problem with this system is although there is a 'Due date' for when the book should be returned by, there is no way to get a list of outstanding and overdue loans for the librarians to try and find. This is obviously a problem as the librarians will not be able to keep a track of what books they need to try and get back, an overdue record could easily go unnoticed for months.

By far one of the worst problems with this system is that you cannot use any filters, this means you cannot search for a particular person, book or record (no record number as a primary key), making it very hard to locate any particular loan which can then prove very time consuming.

When using the current system, only one librarian can edit records, which will obviously be a disadvantage of the current system during busy times, this is because by using standard spreadsheet software, one file will be created for the database and therefore it can be only be opened and accessed once, meaning it cannot be opened and edited on more than one PC as once the file is open on one PC, the file will be shown as 'Locked for editing' when opened on any more computers.

Finally the lack sophistication also means that because there is not one particular record for each person, there is no way to 'mark' an account or make a note on an account to show if any particular people are known to damage books or not return them, and therefore there is no way to identify people that they no longer wish to loan books out to.

All of the above inefficiencies mean that the current system will lead to many duplicate records; also meaning it is easy to lose data and make errors as all information is entered manually. Having no filters or search facilities also means that records are hard to find and this can be very time consuming. This also means that people that repeatedly damage, or do not return books, do not have any way to be traced and therefore cannot be stopped from getting loans of books.

End user requirements

I am now going to look at how a new system would affect the end users. Firstly the system will require details to be retrieved more easily so that every record does not have to be looked at in order to find the correct one. This could be done by searching by student

name or by providing each student with an ID number; this would allow the easiest method of recalling students as it would not matter if two people had the same name.

Another feature of the new system would be that you can easily see how many books a student has taken out and currently has, this could be done by searching for a student record and having a link from this account to all of the books that are currently on loan by this student, this could again be linked to each student ID as each loaned book will have a student ID assigned to it.

In order to keep track on people that have previously lost, damaged or failed to return books, a notes section could also be included on each students account, this would mean the librarian can add notes if they fail to return a book again or can refuse to loan a book because of stolen/lost books that have still not been paid for.

When the database is designed it should ensure that the database can be opened by multiple users, this means loans can be made at different terminals on the same database at the same time, this should help to reduce the waiting time for people wishing to take out books as more than one librarian will now be able to assign loans at the same time.

When the database is designed, it will also be a good idea to have a separate table in the database for student information; this means each student can be given a student ID number to avoid confusion in the case of any students having the same name. This will also mean that it is easier to update any information on students as once the record for a student in this table has been updated; any information about this student in the rest of the database will be updated. It would also be useful in this database if a 'books' table was also used, meaning a final 'loans' table could be used to link information about a book and the relevant student who is having the book as a loan. This would therefore create a relational database and ensure that data is not unnecessarily repeated.

By recording books in this way in their own table, each book can also be assigned its own book ID number, this could be written in the book or made up as a barcode so that the book can be scanned in or the book ID can be manually entered on the 'Loans' table when someone wants to take or give back a book.

By having a database designed in this way, queries can also be run on the database. For example, a permanent query could be saved that shows all outstanding books that are currently on loan, or

queries can be made when necessary, for example to find who has a book if somebody else would like it when it is back.

Once these queries have been made, they can then be produced in a report which may have a particular house style, in this case with the schools logo and colours. This could be useful as it is possible from the database to produce a query of overdue books from each form, a report could then be produced for each of these which could be sent out to the relevant form tutors in order for them to ask students to return books. This would work well as reports would display the relevant information in a clear and professional way.

Finally it must be considered how many people are going to be using the system and the training that they may need to be given, for example it may be that people need to have a few days training on how to use the new system. In this case it will be sufficient for a user guide to be created for the database that staff can read through before using the database, and then be able to refer back to the user guide should they have any problems.

How is this going to be achieved?

The database that I am going to produce for the library is going to be a relational database meaning that the one database will contain many different tables that are all going to be linked together. This will be by the use of primary keys of one table that will be an attribute of another table, therefore becoming foreign keys. The database that I will design will also be able to run queries to find any information required such as overdue books and reports in order for the results of any query to be presented clearly and in a professional manner.

In the relational database that I am going to create I will ensure that the database is as easy to use as possible. Firstly as the Book ID, Student ID and Loan ID are primary keys, they will be automatically generated therefore meaning that new loans, books or students can easily be added to the system. When students are being entered onto the system a drop-down box will also be added in order to select the relevant year group. Once the year group is selected the second drop-down box will then give a list of options for possible form tutors for that year group. I am also going to add a drop-down calendar so that it is easy to see how long a book can be allocated for, for example if a particular book is loaned for three months, it will be easy to select three months in advanced where the relevant date is highlighted, for example if the date is the 23rd, the 23rd of each month will be highlighted for ease of use. To also try and reduce any possible errors, each field will have a set data format, for example the price of a book when put onto the system

will be set to currency. This means that only a price can be entered and no other irrelevant name such as book author or the book title can be entered into the field as this will not be accepted. By linking my tables in this way when a loan is created, a Book ID and Student ID can easily be entered which will link all of the student details with the book details. By linking this on a loans table, other information can be added to this record such as date due back.

After creating my database, I am going to look at the types of queries that I could run from the database in order to help staff keep track of books and loans more easily. I will create queries such as ones to find outstanding loans, find all books that are out to a particular student and to find all books that have a certain Author and are of a particular genre.

Following this, I will then take selected queries that I have created to produce reports. These reports will all have a particular house style when I create them as this will make them easily identifiable for library, which can be achieved by using consistent styles, colours and logos. The reports I am going to create will be ones such as overdue loans linked to student names, these will also be printed for each form group as they can then be passed on to the relevant form tutor in order for the form tutor to try and get the books back. Other reports that I could create are ones such as how many books have been damaged/lost. This report may be useful as this can then be presented to the necessary people in the school who are in charge of ordering new books and they may also wish to know who has damaged or lost books in order to fine them or ban them from taking more books. Finally I will also create a form where this information can be easily entered as a form is a much more user-friendly way of entering data.

One problem that I may encounter when implementing this new system is that staff at the library will not be familiar with using it. To try to help staff get around this, I am going to create a user guide for the database that staff can read to learn how to use the system effectively. This user guide can also include a troubleshooting section so that if any errors occur on the database, a solution can be easily found. This user guide will be clear and contain a good amount of detail. But will however not contain any technical information or any technical vocabulary as this may confuse the staff that are trying to learn about the new system.

To create this system I am going to be using Microsoft Access, this will allow me to create a database that contains many tables and I will easily be able to show how the tables are linked together. I will also be able to use this database design program to create queries

that contain the required information and to turn these queries into any necessary reports, whilst also editing the visual side of the report to ensure that they look professional and that all reports have the same house style.

All of these features will satisfy my end user requirements as it will be possible to easily enter new details and retrieve existing details from within the database. Using the form I will create will also satisfy my user requirements. as any additional details or notes added onto the system by a form is an easier way to do so. This system will then meet any remaining user requirements as it will keep all loans recorded, well organised and will also be able to produce any information required by the use of the queries.

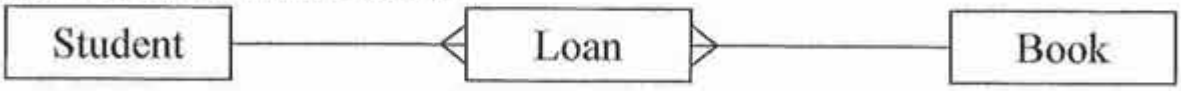
Database Design

An entity relationship diagram shows the relationships that entities have between each other. An entity is a subject of interest to a person or organisation, about which data is stored, for example for the library database, a book would be an example of an entity as object is of interest to the library and data is stored about it.

Each entity also has its own attribute; these attributes are a piece of information about a particular entity, for example attributes for 'Books' could be the book name, author and price.

I am now going to construct my first entity relationship diagram; this will show me the relationship between the entities that I will identify and therefore shows how the system will be structured. These relationships may be Many to Many, where many instances of one entity can be related to many instances of another, for example teachers and students could be the two entities as many students can have one teacher and many teachers will teach one student. Otherwise there could be a Many to One relationship, meaning many of one entity can be related to only one instance of another. For example one mother can have many children however children can only have one mother. Otherwise the relationship could also be a One to One relationship, for example a husband and wife can only be married to one person (each other) at any one time.

This is a very basic version of an Entity relationship diagram, however this is not very suitable for a database, as a Many to Many relationship means that there will be repeating data, which can make the database less efficient and can waste time as data will be entered more than once, to remove this problem, I have now decided to split the entity relationship up and insert an entity in the middle, which will be loans.



Above is my next Entity relationship diagram, here I have split it up using a 'Loans' entity in the middle, this means that the attributes of the entitles should now not be repeated.

In order to analyse the organisation of the entities and their related attributes, I am now going to complete the normalisation process. By completing each stage of this, I am going to be ensuring that there is no repeated data and that all attributes are directly related to their primary key. Firstly I will start by looking at the data in its un-normalised form. When the data is un-normalised, it is all in one

table and therefore has no organisation at all, therefore when un-normalised; the database will be like this:

Un-normalised Data
First Name
Last name
Year group
Additional student notes
Form
Form Room
Form Tutor
ISBN No
Book name
Author
Publisher
Genre
Price
Date
Date due back
Returned?

I am now going to change this data so that it is in 1st normal form, by doing this I am going to remove any key dependencies, this means I am going to create primary keys and ensure that the attribute is dependant on the key field of that entity. I am also going to create a link therefore my 1st normal form is going to look like this:

1st normal form
Student ID
First Name
Last name
Year group
Form
Form Room
Form Tutor
Additional Notes
Book ID

Book ID
ISBN No
Book name
Author
Publisher
Genre
Price
Date
Date due back
Returned?

After creating the 1st normal form, I am now going to remove any part key dependencies to create a 2nd normal form. To do this I have looked at the attributes and have decided to create a loan entity, this is because the date of the loan, date the book is due back and if the book has been returned or not, is not just dependant on the Student ID, it combines the student ID and the Book ID to create the actual loan, therefore this 'loan' entity will be the link between the student ID and Book ID, meaning that this is a part key dependency. This means it will require two foreign Keys which are the primary keys from the other tables; both Student ID and Book ID.

2nd Normal Form

Student ID
First Name
Last name
Year group
Additional Notes
Form
Form Room
Form Tutor

Book ID
ISBN No
Book name
Author
Publisher
Genre
Price

Loan ID
Date
Date due back
Returned?
Book ID
Student ID

I am now going to look at creating my 3rd Normal form. By doing so, I am going to be removing any non-key dependencies. This is where I am going to look if there are any attributes that are dependant on other attributes on any non-keys (another attribute that is not a primary key). After looking at my second normal form, I have decided that I can further split up the 'Student' entity. This is because the Year group, Form room and Form tutor attributes are dependant on another non-key, the 'Form' attribute. This is because the form that a student is in affects what year group they are part of, where their form room is and who the form tutor is. For example, form 11.6 is year 11 and is the 6th form, which will have a

specific form tutor and form room. Therefore by splitting this into its own entity, the data will also not be repeated as many students will be in the same form.

3rd normal form

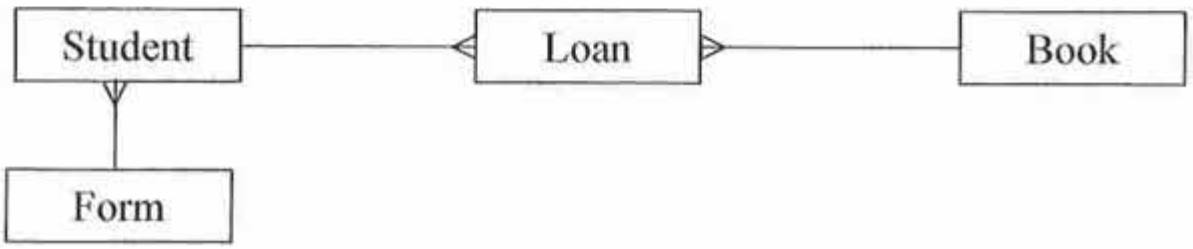
Student ID
First Name
Last name
Additional Notes
Form ID

Book ID
ISBN No
Book name
Author
Publisher
Genre
Price

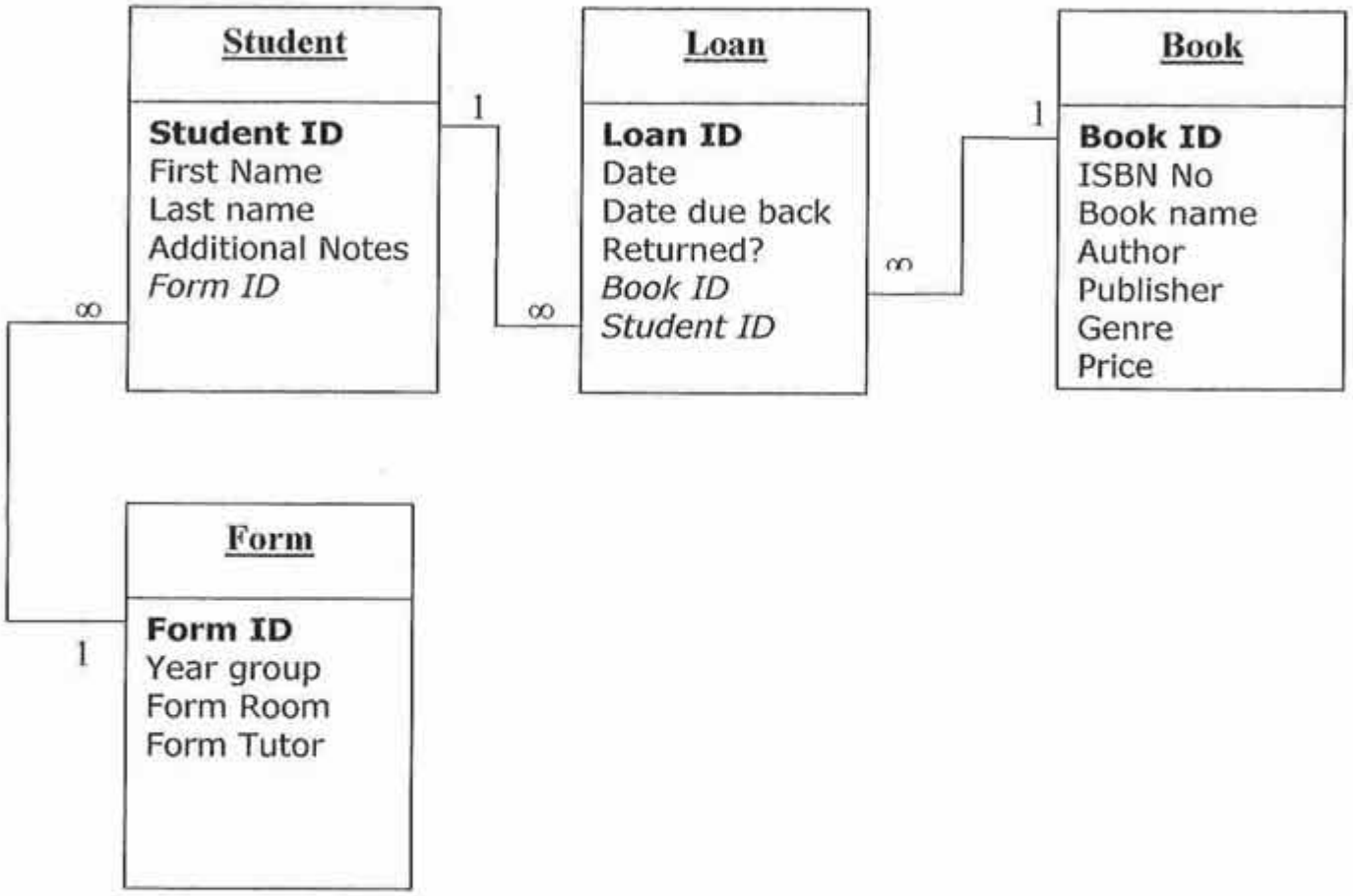
Loan ID
Date
Date due back
Returned?
Book ID
Student ID

Form ID
Year group
Form Room
Form Tutor

After creating this 3rd normal form, each attribute is now entirely dependant on its primary key, this means that there is no repeating data and ensures that the database is as effectively organised as possible. After creating this 3rd normal form, I am now going to create my final revised entity relationship diagram. I have made the relationship between student and form a many to one relationship as one form can have many students but a student can only have one from.



I am now going to create a relationship diagram; this is going to combine everything I have looked at so far by taking each entity and its type of relationship to other entities (e.g. one to many). However I am now also going to use my 3rd normal form to also show the attributes within each entity including any primary keys (Bold) and foreign keys (Italic text), this will then allow me to see exactly how each entity links together by using direct links from attribute to attribute.



This relationship diagram clearly shows the relationships between each entity by linking the attributes (primary and foreign keys) from within each entity. All of the relationships in this diagram are Many to One (represented by 1: ∞).

Data Dictionary

Name	Student			
Description	Table that contains all of the information required about every student to be entered into the database.			
Aliases	N/A			
Relationships		**Related to**	**Type**	**Which end?**
		Loan table	Many to One	One
		Form	Many to One	Many
Attributes	**Name**	**Type/Format**	**Length**	**Key**
	Student ID	Auto-number	4	Primary key
	First name	Text	20	
	Last Name	Text	30	
	Additional notes	Text	200	
	Form ID	Number	2	Foreign Key

Name	Loan			
Description	Table that links students to books with other loan information such as date due.			
Aliases	N/A			
Relationships		**Related to**	**Type**	**Which end?**
		Book	Many to One	Many
		Student	Many to One	Many
Attributes	**Name**	**Type/Format**	**Length**	**Key**
	Loan ID	Auto-number	5	Primary Key
	Date	Date/Time – format will be DD/MM/YYYY		
	Date Due back	Date/Time – Format will be DD/MM/YYYY		
	Returned?	Boolean check box – Yes/No		
	Book ID	Number	5	Foreign Key
	Student ID	Number	4	Foreign Key

Name	Book			
Description	Table that will contain all information related to each individual book			
Aliases	N/A			
Relationships		**Related to**	**Type**	**Which end?**
		Loan	One to many	One
Attributes	**Name**	**Type/Format**	**Length**	**Key**
	Book ID	Auto-number	5	Primary key
	ISBN Number	Number in format nnn-n-nnnnnn-nn-n	13	
	Book Name	Text	30	
	Author	Text	20	
	Publisher	Text	20	
	Subject	Text – Format as a drop-down list	20	
	Price	Currency in format £nnn.nn	5	

Name	Form			
Description	Table that contains the information on the form ID, the year group, tutor and the tutors form room.			
Aliases	N/A			
Relationships		**Related to**	**Type**	**Which end?**
		Student	One to many	One
Attributes	**Name**	**Type/Format**	**Length**	**Key**
	Form ID	Auto-Number	2	Primary Key
	Year Group	Number	2	
	Form Room	Text – E.G TA45.	4	
	Form Tutor	Text	20	

Query Designs

The purpose of queries

The purpose of Queries are to search for data that is stored in a database, this is useful as it is much faster than manually looking through each table of a database as once these are full of data this would take a very long time whereas by running a query the relevant results can be found much quicker.

There are therefore three main types of queries used within a database; firstly there are 'simple' queries. These are where there will be one specific criteria in one field of a database or table. Secondly there are as 'parameter' queries that can be run, this is where the user of the database can enter a specific or range of values to be searched for, such parameter queries will display an appropriate message on-screen asking the user to enter the criteria for that query. Finally there are also 'complex' queries, these are queries made up of criteria in 2 or more fields, or use parameters such as 'AND' or 'OR' to search for one thing 'And' another or to search for Data A 'or' Data B.

I am now going to design ten queries that I am going to use on my database once I have created it. I am firstly going to look at two simple queries and then design other parameter and complex queries that will meet my user requirements that I have already outlined.

List of Queries

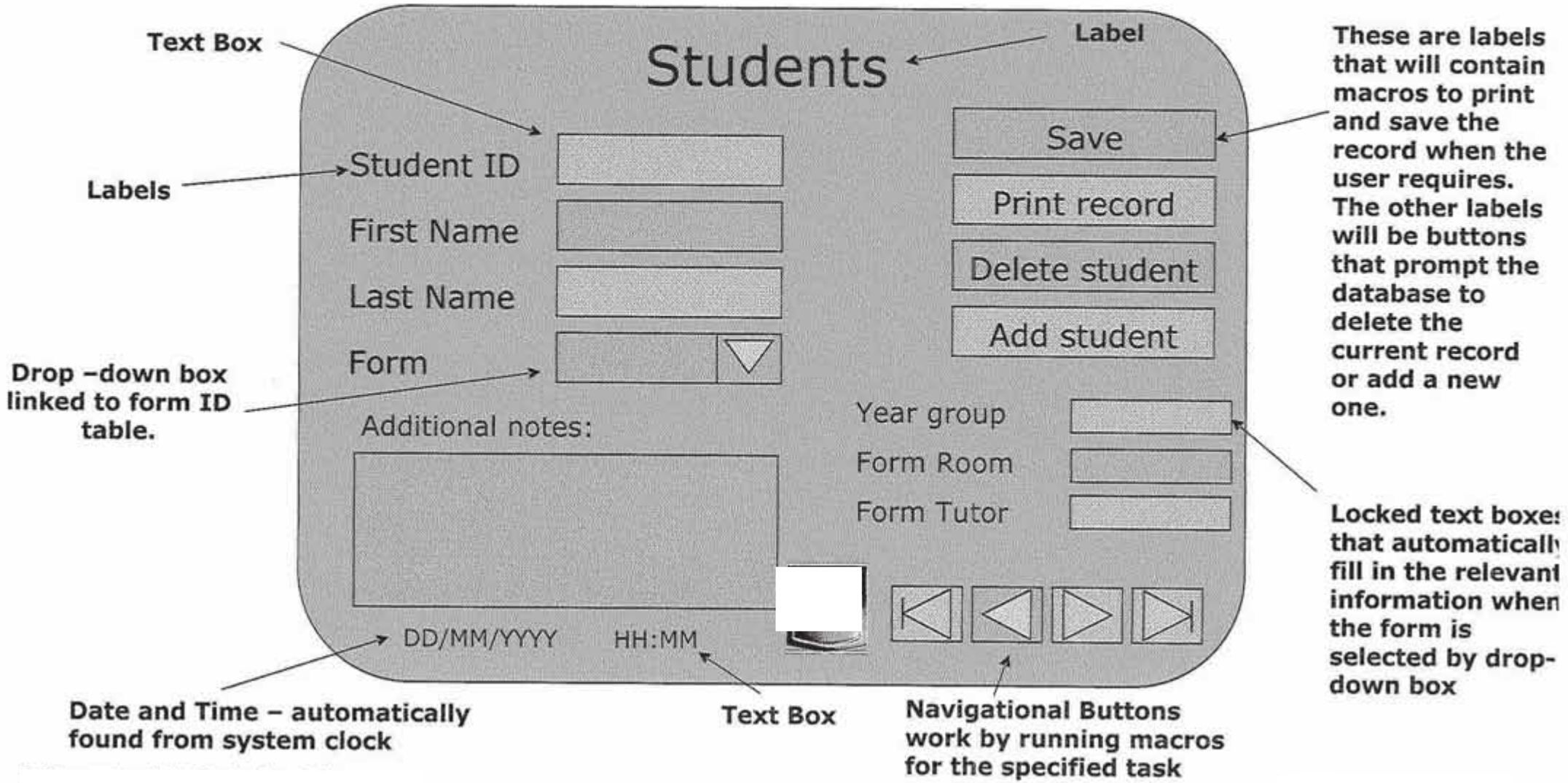
Query Name and type	Underlying table(s)	Criteria	Fields/table
Find all books due back today (Simple)	Student table, Loan table and book table	Date due back = Today	**Student ID** First Name Last name *Form ID* **Book ID** ISBN No Book name Author Publisher Genre Price
All books by [author]	Book Table	Author = [please enter author]	**Book ID** ISBN No Book name Author Publisher Genre

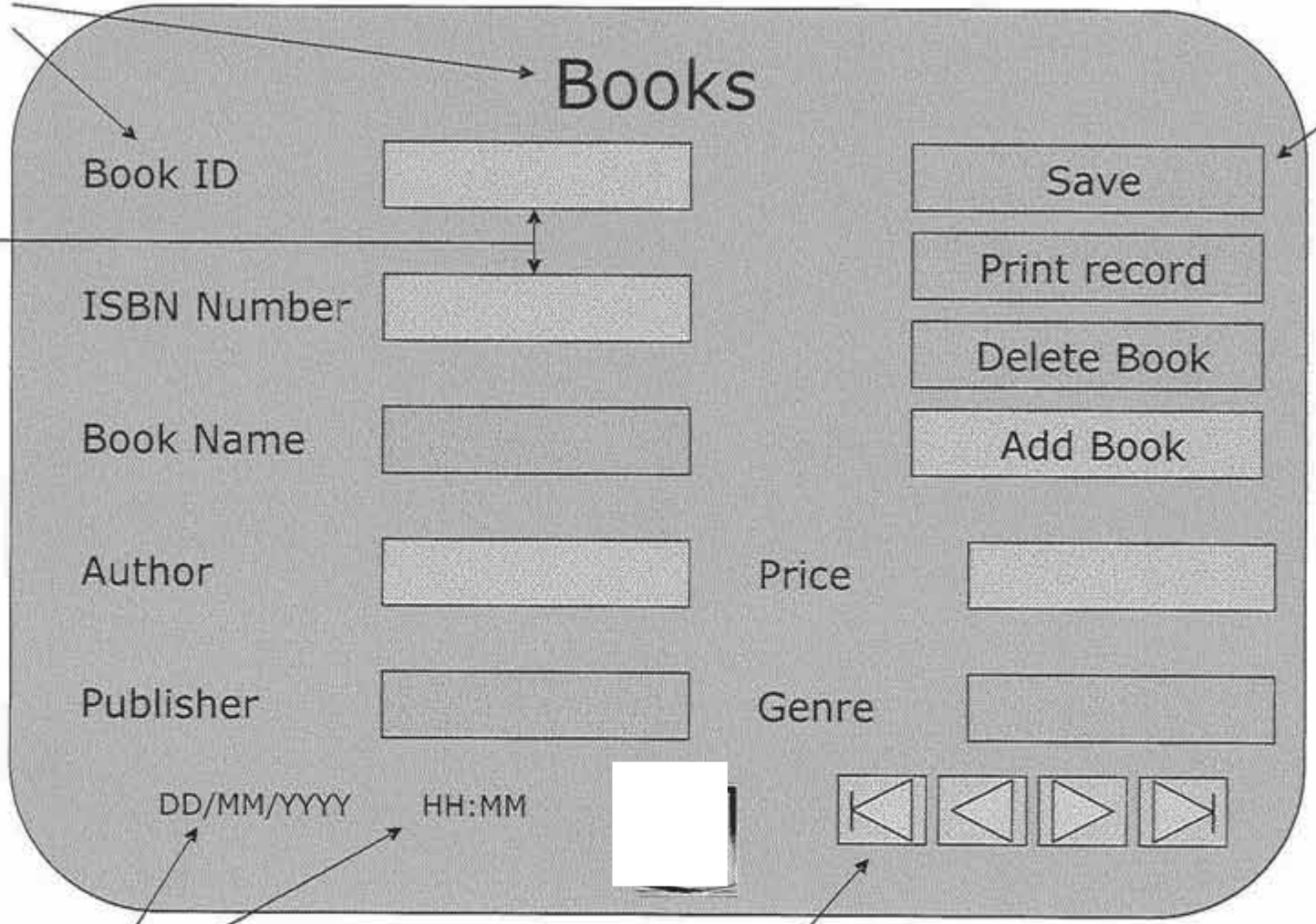
			Price
Find all books that are out to [student id number] (parameter)	Loan table, book table and student table	Student id= [please enter student id number]	**Book ID** ISBN No Book name Author Publisher Genre Price **Student ID** First Name Last name **Loan ID** Date Date due back Returned?
find all [genre] books (Parameter)	Book table	Genre = [please enter genre]	**Book ID** ISBN No Book name Author Publisher Genre Price
find all books [due back on date] (Parameter)	Book table and loan table	Date = [please enter date]	**Book ID** ISBN No Book name Author Publisher Genre Price
Find all books that have the [author] And are of [genre] (Complex)	Book table	Author = [please enter author] and genre = [please enter genre]	**Book ID** ISBN No Book name Author Publisher Genre Price

Find all overdue books from [form id] (Complex)	Loan table, form table, student table and book table.	Returned = false AND form id = [Please enter form id] And Due back date <today	**Book ID** ISBN No Book name Author Publisher Genre Price **Student ID** First Name Last name
Find all students with [First name] and [Last name]	Student table	First name = [please enter first name] and last name = [please enter last name]	**Student ID** First Name Last name *Form ID*
All students that have a overdue loan by at least 2 weeks (Complex)	Loan table and student table	Returned? = false Date due back = <today-14()	**Student ID** First Name Last name *Form ID* **Book ID** ISBN No Book name Author
Find all [genre] books that are due back between [start date] and [end date] (complex)	Book table and loan table	Genre = [Please enter genre] Due back date = between [start date] and [end date]	**Book ID** ISBN No Book name Author Publisher Genre Price **Student ID** First Name Last name **Loan ID** Date due back

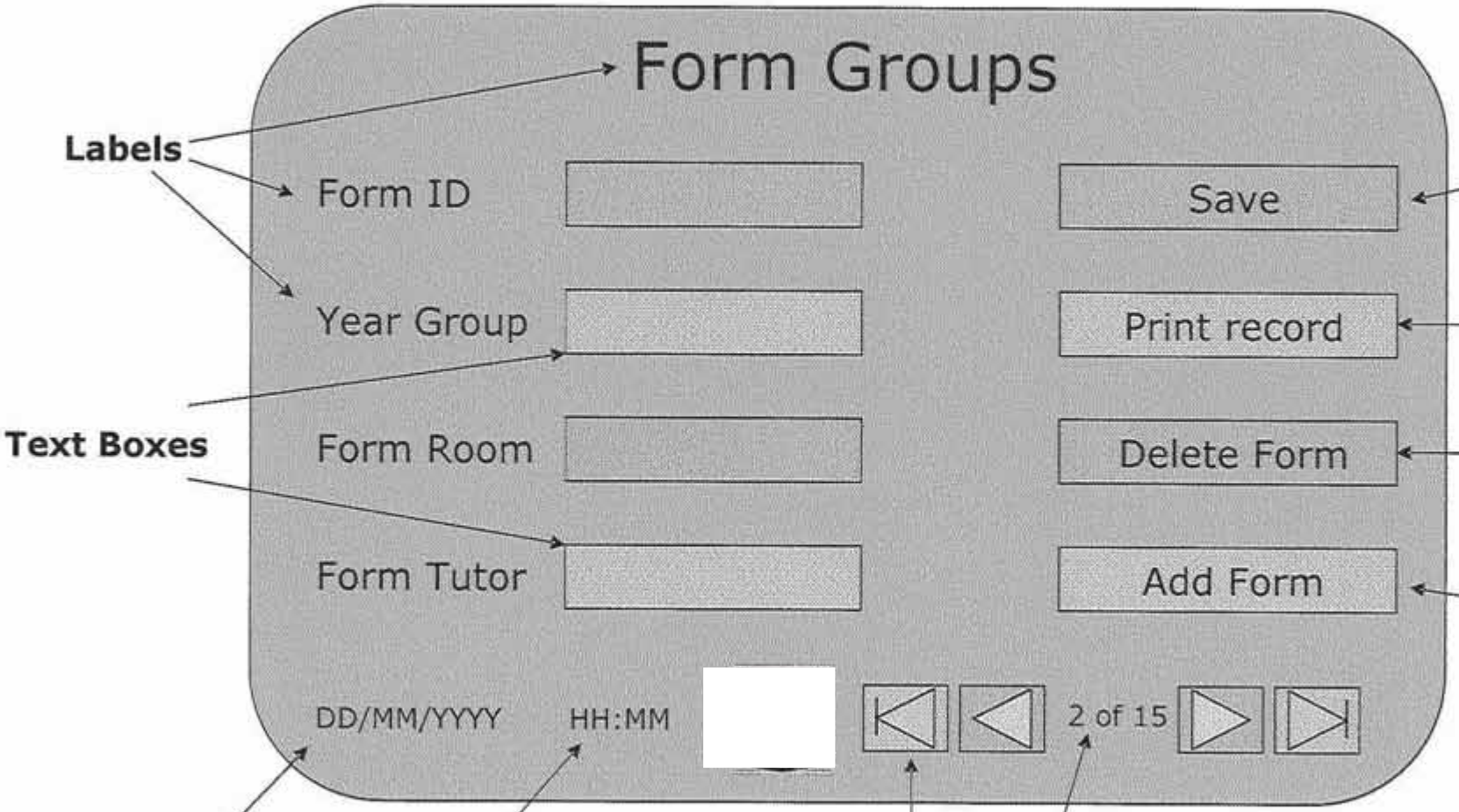
Form Design

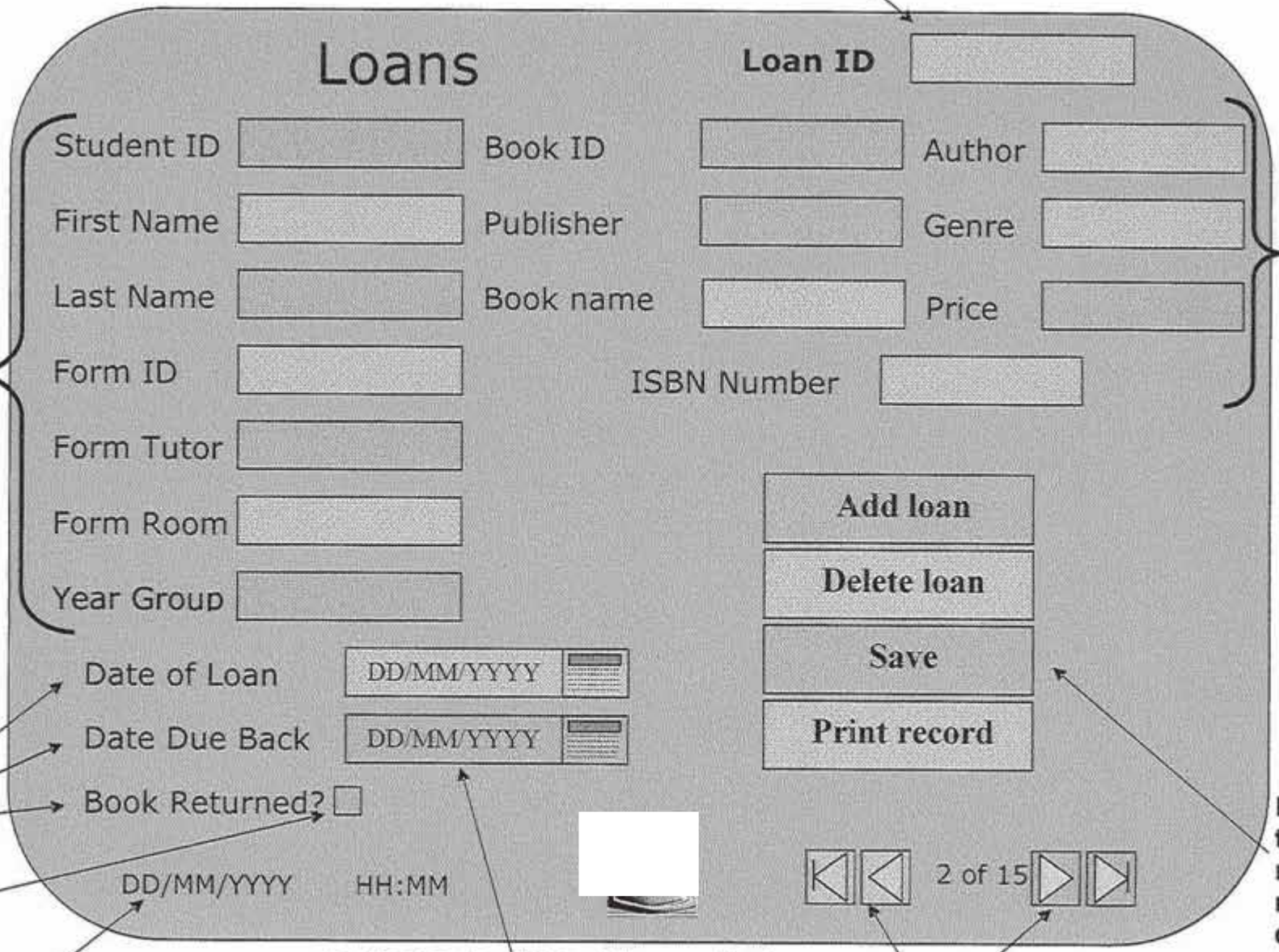
I am now going to design the data forms that I am going to create in order to both find data and add data to my database, by using data forms the database will be much more user friendly as it is much easier to understand and navigate, rather than using various tables within database. It will also ensure that only the correct information can be entered and will therefore reduce errors by the use of validation rules.




Labels
Books
Book ID
Text Boxes
ISBN Number
Book Name
Author
Publisher
Price
Genre
Save
Print record
Delete Book
Add Book
DD/MM/YYYY
HH:MM
These are labels that will act as buttons to perform specific tasks by the use of macros; they will save and print the record. The final two buttons will add a new book to the system and delete the current book that is showing on the form
Current date and time shown, automatically found from system clock
Navigational buttons to go to first record, last record, previous record and next record, these can be programmed using macros.

On all of the forms I am designing, I am ensuring that they are all going to have a consistent house style by using the same font and by having similar headers at the top of each form. Each form also has the same background, text box colours and Deferrers logo as well as having navigational features and the current date and time within the footer of the form.





Loan ID for primary key when a new record is added or to recall a previous loan

This set of labels and text boxes make up the data found from other tables, after entering the student ID, all other entries are automatically filled in and locked so they cannot be edited. Information for this part is found from the Student table and its linked table, the Form Table.

These fields are also automatically filled in once a Book ID is entered, this information is found from the Book ID table. By including information from the three other tables, the loan table clearly compiles the information of a student and a book to create a record every time a book is loaned out.

Labels

Check Box

Date and Time labels

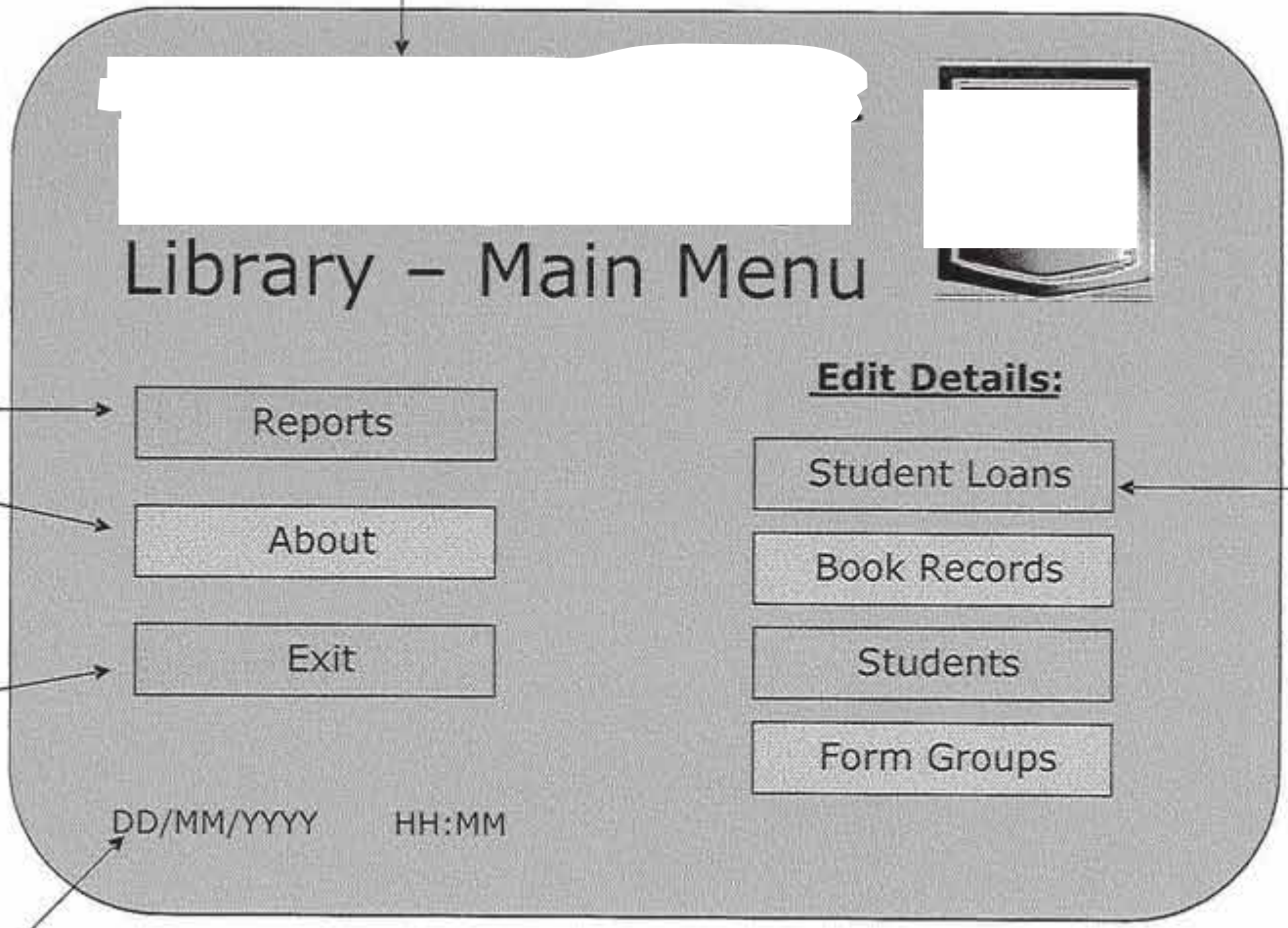
Locked auto-fill text boxes that automatically fill in when drop-down calendar is selected.

Labels to act as navigational options between records.

Large Buttons to run relevant macro on the database E.G to save the record or add a new record.

Label

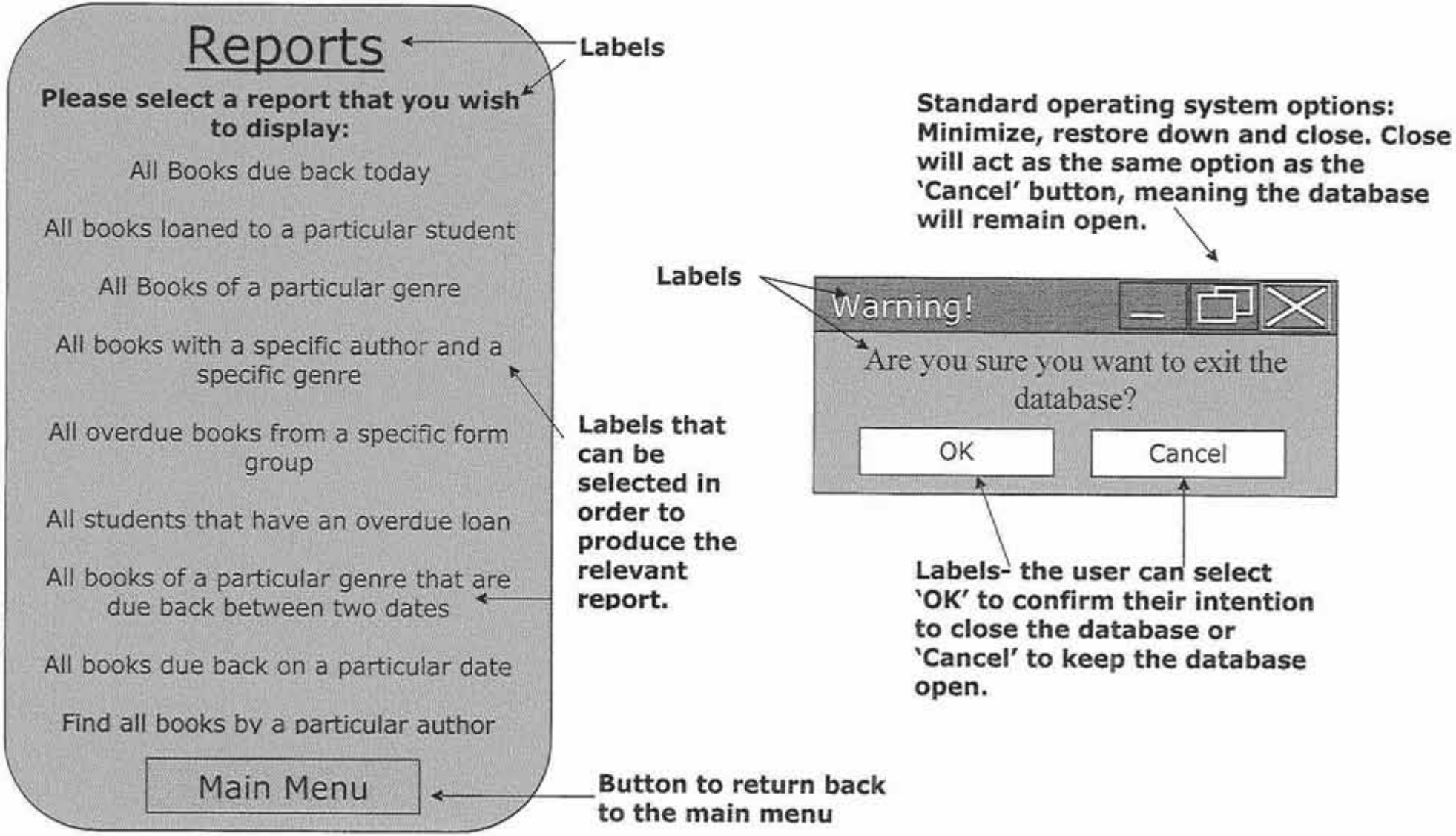
Labels will act as buttons, the reports button will navigate to a reports sub-menu listing all of the available reports that I have designed. The about menu will open a window stating information such as the designer, version number and date of the last update. The exit button will bring up a warning asking the user if they are sure they want to exit the database.



These labels will also act as buttons and will navigate the user to the relevant form within the database. This means that the user can easily navigate to different forms within the database. Once the user has navigated to a particular form they can navigate back to the main menu by selecting the logo on the bottom of each form.

Date and time labels from system clock

I am now going to design the sub-menu and warning message that will b e displayed when the user selects the 'reports' and 'exit' buttons from the main menu. By making my database work this way I can be sure that the user will not accidently exit the database and that all reports are easy to find.



Report designs

The purpose of reports

The purpose of a report is to present data in a clear, understandable and professional manner. These reports are usually in a printed format however you can also have on-screen reports that are basically a form that displays the result of a query.

Reports can be run from either a table or query, this means that all or selected information from a table can be printed in a professional manner. Any queries that I have run could also be made into a report that can be printed off and used for the relevant purpose. I am going to look at the 10 queries that I have designed and decide which of these would be appropriate in a report format and why I may want these results displayed as a report. For example, to meet my user requirements it will be necessary to produce a report for each form group showing overdue books that any students may have. These will look much more professional than simply printing the report as a query as it can be made into the appropriate house style and grouped or organised relevantly.

List of Reports -

Report name	**Purpose**	**Underlying query**	**Additional information**
All books due back today	To generate a clear printed list that shows all books that should be returned during the course of the day.	Find all books due back today (Simple)	The relevant results will be displayed in alphabetical order by book name.
All books loaned to a particular student.	To produce a printed report that can be given to a student upon request so that they can take it away to find the books at home.	Find all books that are out to [student id number] (parameter)	As all results on this report will be for one student, the report will also be sorted in alphabetical order by book name.
All books of a particular genre	To generate a printed report that can show all books of a	find all [genre] books (Parameter)	The books can be ordered into alphabetical order by book

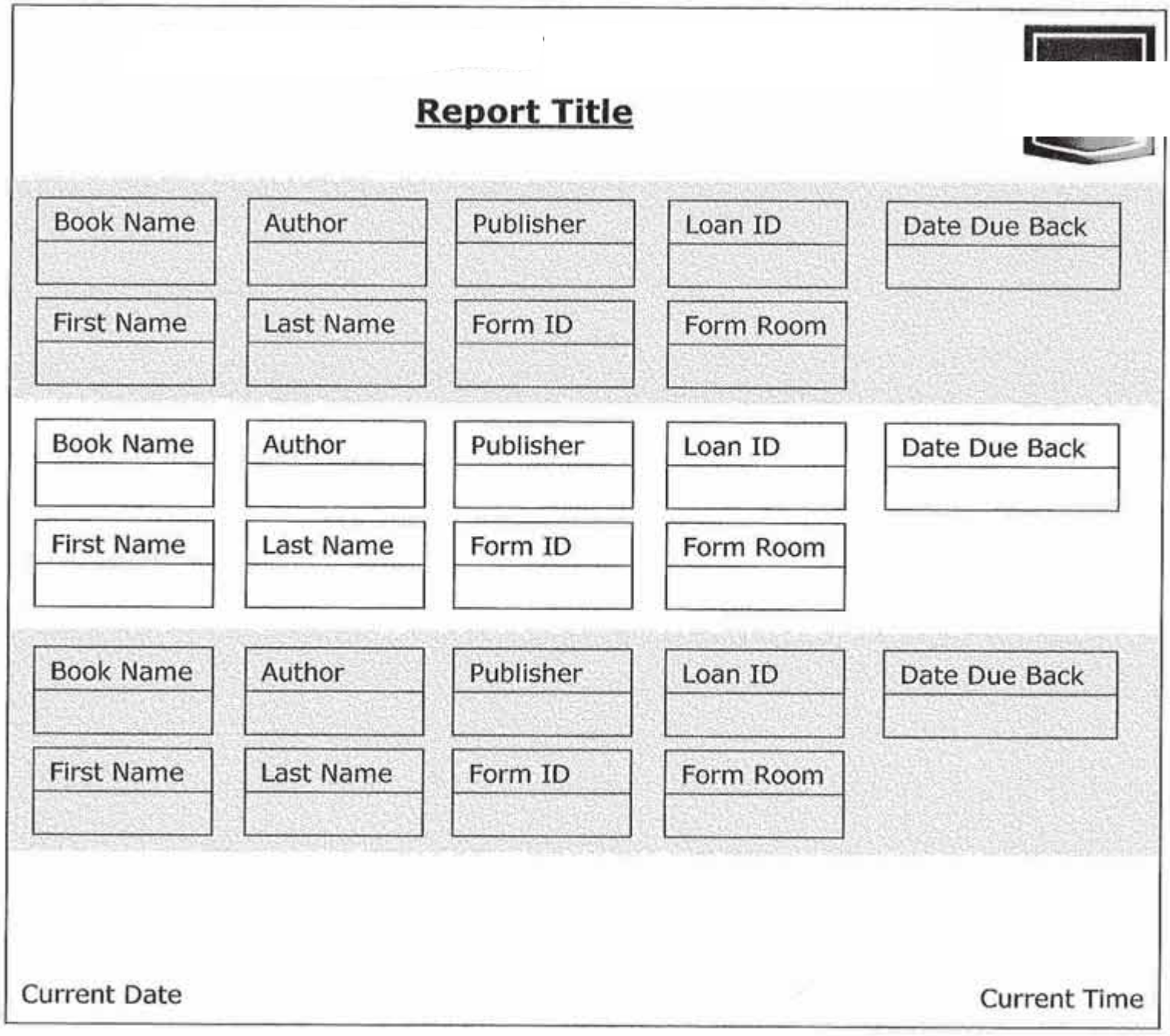
	particular genre, this is useful if a student asks for a book on a particular subject or genre as this can then be handed to the student to look around the library.		name, information such as price and genre can be omitted as they are not relevant and the genre will be the same for all.
All books with a specific author and a specific genre.	A particular student may like a particular author and want to research a subject that they like. By producing this report it can be handed to the student so they have a list of books that would be relevant to them	Find all books that have the [author] And are of [genre] (Complex)	These books can again be sorted into alphabetical order so that they are clear and presented professionally. As the title will state the author and subject of the book these do not need to be included.
All books overdue for a particular form.	This report will meet my user requirements as it will allow a list for each form to be generated so that form tutors can ask their students to return the relevant books	Find all outstanding books from [form id] (Complex)	This list of results will be grouped by student as otherwise the same student may be repeated many times. These student names will be in alphabetical order ensuring that students are easy to find and no data is repeated. This

			report will have a calculated element which will count how many books are overdue for that whole form and will also sum the price of all overdue books to show how much the whole form owes in books.
All students that have an overdue loan	This report may wish to be handed to senior members of staff to start to chase up students who have not returned books which are now more than 2 weeks overdue.	All students that have a overdue loan by at least 2 weeks (Complex)	These results will be grouped by student so that if one student has more than one book they both appear together. This could then also be organised into alphabetical order.
All books of a particular genre that are due back between [start date] And [End Date]	This report may be used to find when a particular book is back if a student needs it, for example a student may need a book in the next week for History. This means if a book is not available a librarian can tell them when it should be back.	Find all [genre] books that are due back between [start date] and [end date] (complex)	These books will be listed in order so that the ones that are due back first will be displayed first in the list. This report will not show other students names or details but simply the book name, author and due back date.

All books due back on a particular date	This report may be handed out to find all books that are due back, for example on the last day of a term or year, this means a reminder can be handed out early as handing one out on the due back date or afterwards will not be very effective as students will not be at school.	find all books [due back on date] (Parameter)	This report will be sorted alphabetically by student name followed by numerical order of the form ID in the case of two students having the same name.
All Students found by Name.	A report can be created from this that can be given to a student. This can be used to remind them what their ID number is so they do not forget it again	Find all students with [First name] and [Last name]	
Find all books by a particular author	People may want to find a book by a particular author which they like, in this case report can be given to them showing all books they may like	All books by [author]	The book table will be shown however the author attribute will be omitted from this list as it will be the same for all books.

Designs for reports

I am now going to design example reports for each of the reports I have listed above.

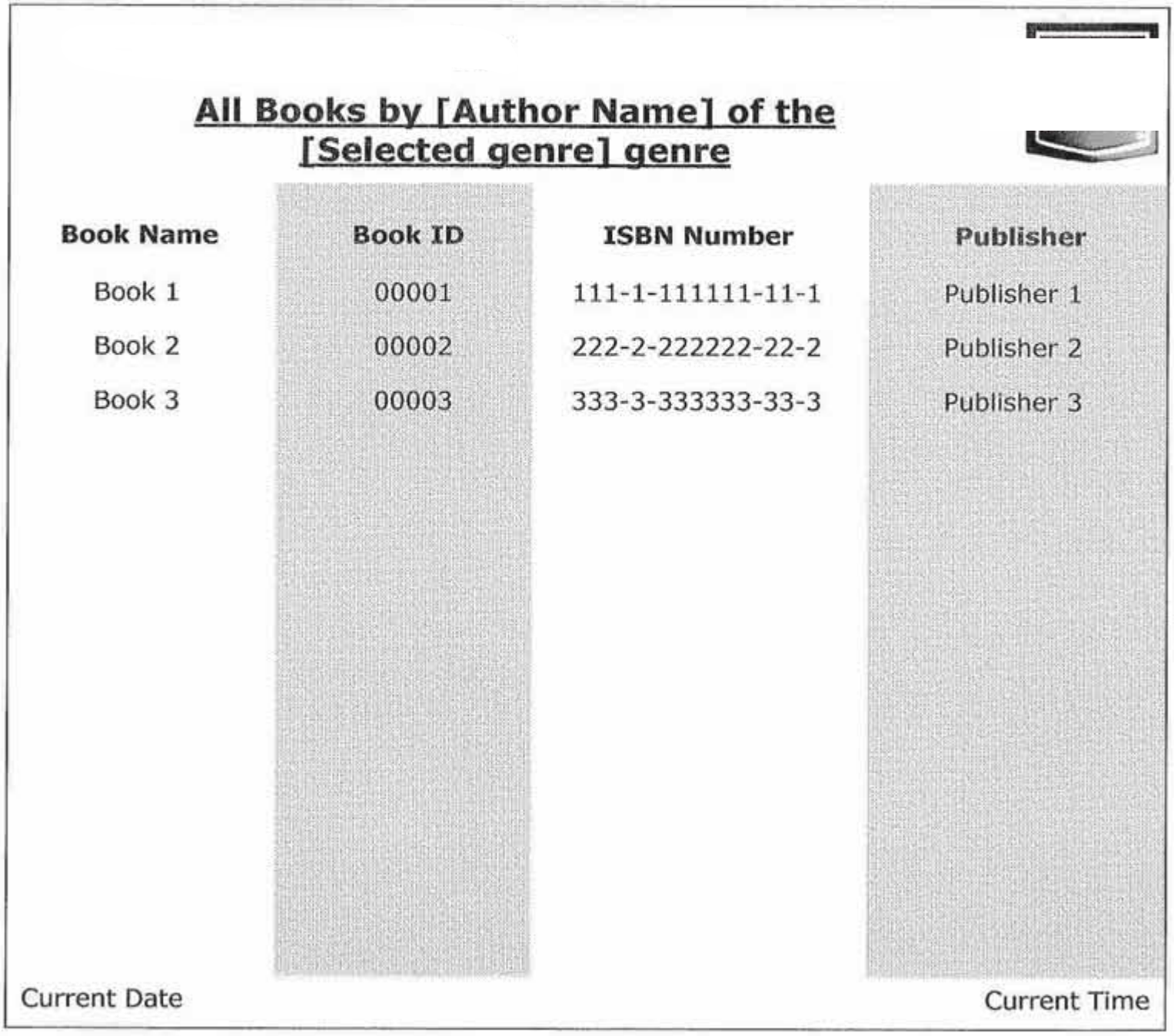


Above is the first design for my reports that I will create for my database. It will list all of the required fields from each table and clearly shows the house style from the colours as well as the logo. In the footer of my report there will also be a current time and date show from the system clock, this will ensure that any reports printed can easily be identified as being out of date

This report will be used for the following reports with various fields according to the relevant fields i selected when creating my query:

- All books due back today
- All students that have an overdue loan
- All books loaned to a particular student.
- All books due back on a particular date
- All books of a particular genre that are due back between [start date] And [End Date]

I am now going to create my second report design. This report will be used for the report 'All books with a specific author and a specific genre'.



All Books by [Author Name] of the [Selected genre] genre

Book Name	Book ID	ISBN Number	Publisher
Book 1	00001	111-1-111111-11-1	Publisher 1
Book 2	00002	222-2-222222-22-2	Publisher 2
Book 3	00003	333-3-333333-33-3	Publisher 3

Current Date

Current Time

This second report has the same house style as the first report that I have created however rather than being in a justified format as the first one is, this is presented as a tabular layout as this makes the information clear and easy to read.

I am now going to create my third report that will be used for the following queries:

- All Students found by Name.
- Find all books by a particular author
- All books of a particular genre

Report Title

First Name	
Last Name	
Student ID	0001
Form ID	02
Form Room	TB35

Book Name	A-Level History
ISBN Number	123-4-567891-01-1
Author name	Joe Bloggs
Publisher	Dolphin books
Genre	Educational

Current Date Current Time

For my third report I have again ensured that I am following my house style by using the same style and size text, by using the deferrers logo and the green stripe across alternating records meaning it is easy to recognise where one record starts and one ends. For this design I have created the report using two columns. I have also show on the design the two different sets of headings I shall use dependent on the report. This is because for the report to find all students after searching by their name, I will use the first set of headings to show their name, ID and other details. Alternately, the other two reports show details on books so I will use the other set of headings showing the book name, ISBN number, Author name, publisher and genre.

I am now going to design my final report, this is going to contain a paragraph to explain the situation to form tutors as it will be sent to each form tutor asking for any student's overdue books.

College

Form Tutor: [Form tutor entered here]

Form Room: [Form Room entered here]

Please find below a list of all students in your form that have overdue books from the College library. It is necessary that these are returned soon otherwise charges will be sent for the price of the book as listed below.

Students should be encouraged to return these books as soon as possible as otherwise they may not be allowed any new loans, unless the price of the book has been paid to the school. If any students have queries about this or have misplaced the book please allow them to come up to the library to bring this to our attention.

Student Name	Book name	Due Back Date	Price

Current Date

Current Time

This was my final report design that will be sent out to form tutors to show any overdue books, again this contains the same house style as all other reports. Once this report is run the necessary information such as Form Tutor and Form room will automatically be entered as well as information about students with overdue books. Finally the student name is show by placing both the student first name and student last name fields under this one heading so the students name appears as one entry rather than two.

Implementing my database

I am now I am now going to look at how I have set up my database in Microsoft Access 2007 including how I have set up each individual table including validation rules to ensure that Incorrect information cannot be entered by accident. Following this I am then going to look at how I have set up various queries for my database followed by Reports, Forms and menus that can be used in order to hide the background database from the users. This means that they then cannot see what the underlying database looks like as this may confuse them and therefore prevents them from accidently doing any damage to it.

After looking at my designs, I can see that I firstly started by creating 4 main tables. These were tables for each Loan, Student, Book and Form group, all of these which will be linked together. This screenshot clearly shows the first student table that I have set up. I started by creating the first field and setting this to be the 'Auto number' field which will make it the primary key, meaning each student will be assigned a unique number. I then continued to add the relevant fields that had already been planned.

As has been shown by the above screenshot, I have used validation rules on the First and Last name to ensure that numbers cannot be entered into the field, as the 'Text' attribute for that field usually allows both letters and numbers to be entered, however this clearly cannot be the case for a first and last name. I have also then entered 'Validation Text' to display the message 'Last Name much contain letters and no numbers' should anyone try entering numbers into this field. Using such validation rules should mean data entry into the database should be much more accurate.

Finally I have also decided to add an 'Additional notes' section as this will allow librarians to input specific notes about students should they wish E.G. should they have any special requirements or should they have a reputation of not returning books. This table will also have a foreign key (an attribute that is a primary key in another table) as it will link each student to a particular form. This has been left as a 'Text' field at the moment as I am going to add relationships between tables in once they have all been made.

The next table that I then decided to create was the table containing information on Loans. This table is one that will use two primary keys as it will bring lots of information together. This table is going to include a 'Loan ID' which will be the primary key of this table as each loan will therefore be provided with an individual load number. This table will then also include the dates of when loans were given and the date that such loans are due back. It will also include a Yes/No checkbox as to whether the book has been returned or not. This table will then also contain the foreign keys 'Book id' and 'Student id' as this will then link a specific book that has been taken out with any student in the school, meaning the loan table is the one that will link these two pieces of information together.

Following this I then changed some of the validation options for the 'Date of loan' field and the 'Date due back' fields. For the 'Date of loan' field I have simply selected the 'show date picker' option and selected the option 'for dates' meaning as this has been set to a 'short date' format box, a calendar will pop up next to the input field where the user can select the date from a calendar rather than have to type the date in – meaning they should not encounter any validation options. For the 'Date of loan' I then also included some other validation rules to ensure that the date of the book due back could not be entered incorrectly. Should anyone input data incorrectly into this field it is likely that they may enter the date due back before the date of the loan being made, meaning the database would become inaccurate. It is for this reason I have included a validation rule as >Now()+7 which means that the date due back must be 7 days after the day the loan is being made as the minimum loan time for our library is 7 days.

The next Table that I have created is a table that contains information on books that are kept within the library.

This design view shows the fields that I have designed for my 'Books' Table. As each book will need to be assigned an individual number, I have set the 'book Id' to be the primary key which will be set as an 'auto number' field meaning this number is automatically generated every time a book is added into the database. I have then also included other attributes within this table as planned in my designs for the database. I then also began to include validation rules on specific attributes to help ensure that people fill in the database correctly. Validation rules include the setting of the 'Price' field to a 'Currency' field type and setting the number of decimal places to 2. This means that there can be no mistakes in the format of the price, however as this will now default to £0.00, there is the possibility that this could be left blank at this price as the field will still have a value, however by setting the validation rule to '>0' I can ensure that the price has to be at least £0.01 and therefore this field now has to be filled in correctly and cannot be missed out.

The final validation rule that I included within this table is one for the 'ISBN Number' attribute. As ISBN numbers have a set format it would be fairly easy for anyone inputting data to record the ISBN numbers slightly inaccurately, this would then mean things such as queries or reports may not display or run correctly. In order to specify the format that the ISBN numbers need to be entered I entered an 'Input mask' rule so that the ISBN number can only be entered a certain way. By using '0's in the input mask I was specifying that the input characters can only be numbers rather than

letters, therefore all ISBN numbers have to be in the format '012 -3 -45678901 -23 -4' and therefore can only be input incorrectly if the wrong number is input by the user.

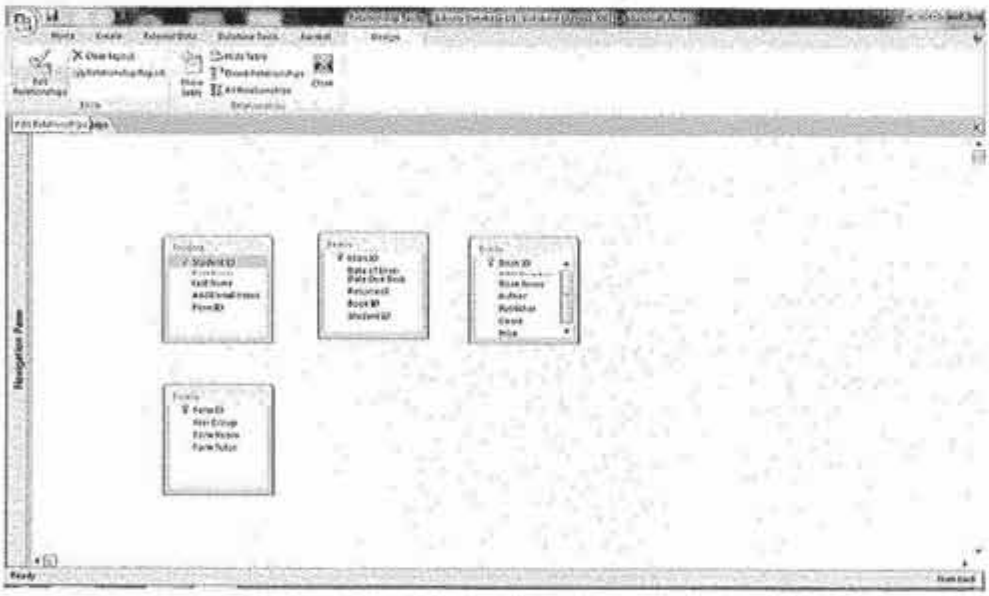
The final table that I am going to create using design view is the table containing information on Form Groups.

The first thing done on this Table was again to introduce my primary key; this was going to be my 'Form ID' where each form group is again assigned an individual number for reference within the database. The next attribute that I have used within the database is to specify what year group the form is from. As year groups within the school range from Years 7 to 13, I have used the validation rule 'Between 7 and 13' as this means only the years 7, 8, 9, 10, 11, 12 and 13 can be input into this field, all other numbers entered will produce the error message 'Year group must be between year 7 and year 13' therefore decreasing the chance of incorrect information being entered.

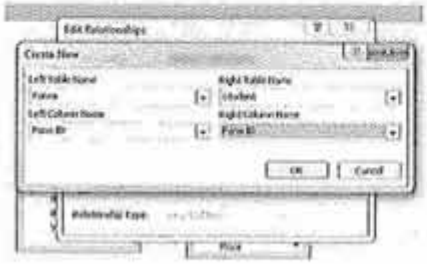
The next attribute within the Table is to specify the room number for a form group within the database; this attribute format has been set to 'text' as this will allow a mixture of letters and numbers. As the rooms within the school range between TA, TB and TC to specify which floor level they are on and are then followed by two numbers to specify which room on that floor E.G. TC09. In order to specify this in the database I have included a validation rule along with an Input mask to ensure this information must be entered correctly. To ensure that the room starts with TA, TB or TC, I have used the validation rule 'Like "TA*" or Like "TB*" or Like "TC*" with the * meaning further characters can be added afterwards. I have then used a Subnet mask of "??##". When the database reads this the ? stands for letters A-Z and the # stands for numbers 0-9 therefore the combination of

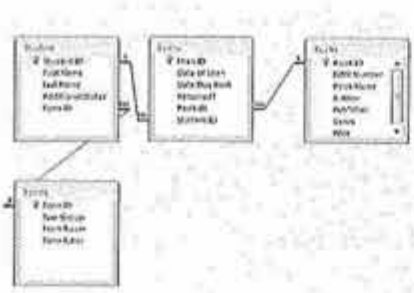
the subnet mask and validation rule will only allow for the combinations between TA00 and TC99, meaning that the chance of the user inputting incorrect information on this field is greatly reduced.

The final validation rule on this database I have used is to ensure that the Form tutors name is input correctly, however this can only be a very general validation rule as somebody's name could be anything, therefore I have simply specified that a form tutors name must begin with Mr, Mrs, Miss or Ms. I have done this using the command words 'Like' and 'or' within the validation rule e.g. Like "Mr *" OR like "Mrs *" Or like "Miss *" Or like "Ms *" Once again the * within this validation rule means that any text can be added after this beginning. I have also included the field size limit as 20 characters as per my plans as this should allow significant space to enter a teachers' title and surname.



The next part of designing my database was to add in the relationships that will link all four of my tables together so that data can be shared between them. To do this I first opened the relationships view and used the 'Show table' button to add in all tables that I have created onto the relationship view. Following this I then selected the 'Edit relationships' button to open up a dialogue box where you can create and edit the relationships across the database. Once I had done this I then selected the 'Create new' relationship selecting the fields within certain tables that I wished to link together – this had already been decided within my plans as it will be linking primary keys to foreign keys in other tables. I then repeated these steps until all three relationships for my database had been added. When creating the database I also selected the option 'Enforce Referential integrity' which means that if something is entered or deleted from a table, other tables will be checked. For

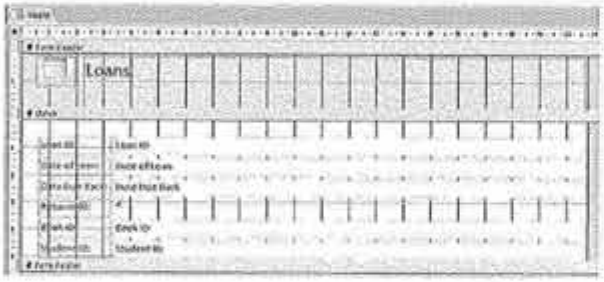




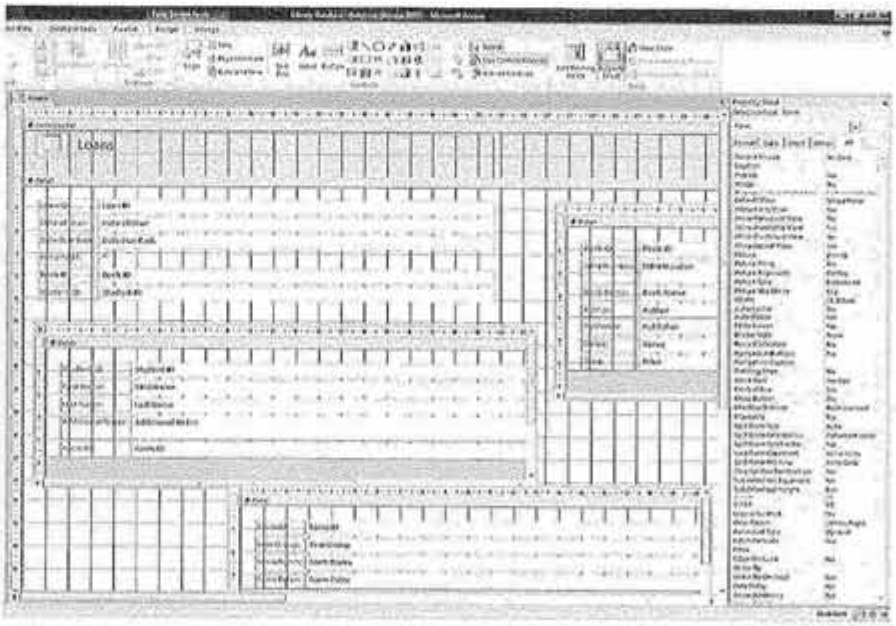
example it will not let you delete a student if they have a current loan in the loans table; this therefore leaves this relationship diagram.

The next part of my database creation was to make my data input forms that I had previously designed. These will be used to hide the background database so that users do not get confused and/or make changes that they do not intend to make, therefore by doing this I will be increasing both the security and accuracy of the database, as tables will be hidden and the users will be less likely to make mistakes.

I am now going to look at how to make the form for my 'loans' table. This will be used for the librarians to input data about new loans. I am going to do this by also using sub-forms within this main form. This means that when the Student ID and Book ID are entered, the corresponding data will be called from the other tables and be displayed within this main form.

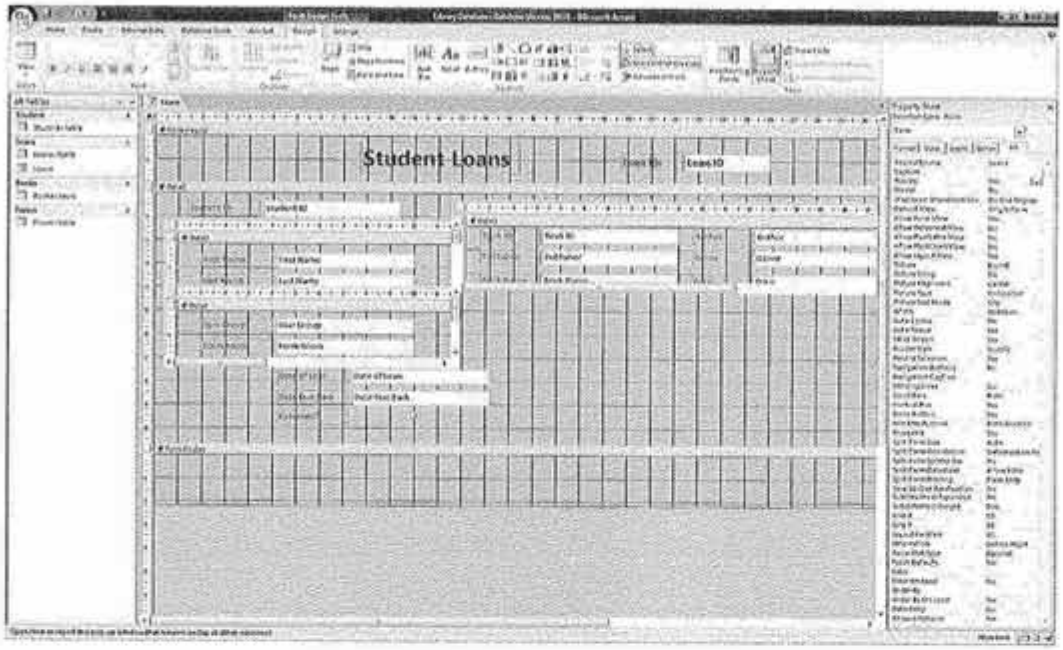


This screenshot to the left shows the form in its basic pre-made state before I have edited it. To begin creating this form I have started by selecting the 'create' tab followed by the 'Form' button. Once this form has been created I then opened it within design view where I could then begin editing the content within the form.

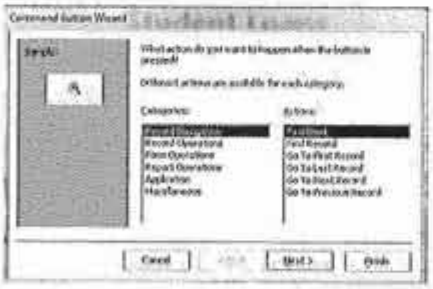


This next screenshot now shows how I have edited the form within design view to create the desired effect and information to be shown. To do this I increased the size of the 'detail' section and then selected each one of the tables I have created (Student table, Form table, book table) and dragged these onto the form. This then automatically input these as sub-forms that will be linked to the loans table. I then selected each one of these sub-forms and edited the properties by right-selecting the forms and selecting 'properties' which gave the property editing toolbar on the right side of the page. I then changed the settings 'Border style' to 'None' so

that no border was put around each sub-form as this would not look very professional. I then also changed the option 'Record selectors' to 'No' as I only wanted the linked record within the sub-form to be displayed, therefore did not want visible record selectors. I then also changed the settings 'Navigation buttons' to 'No' and 'Scroll bars' to 'neither'. This meant that all options within the sub-form were removed and therefore it simply displayed the fields that I wished it to of the one linked record.



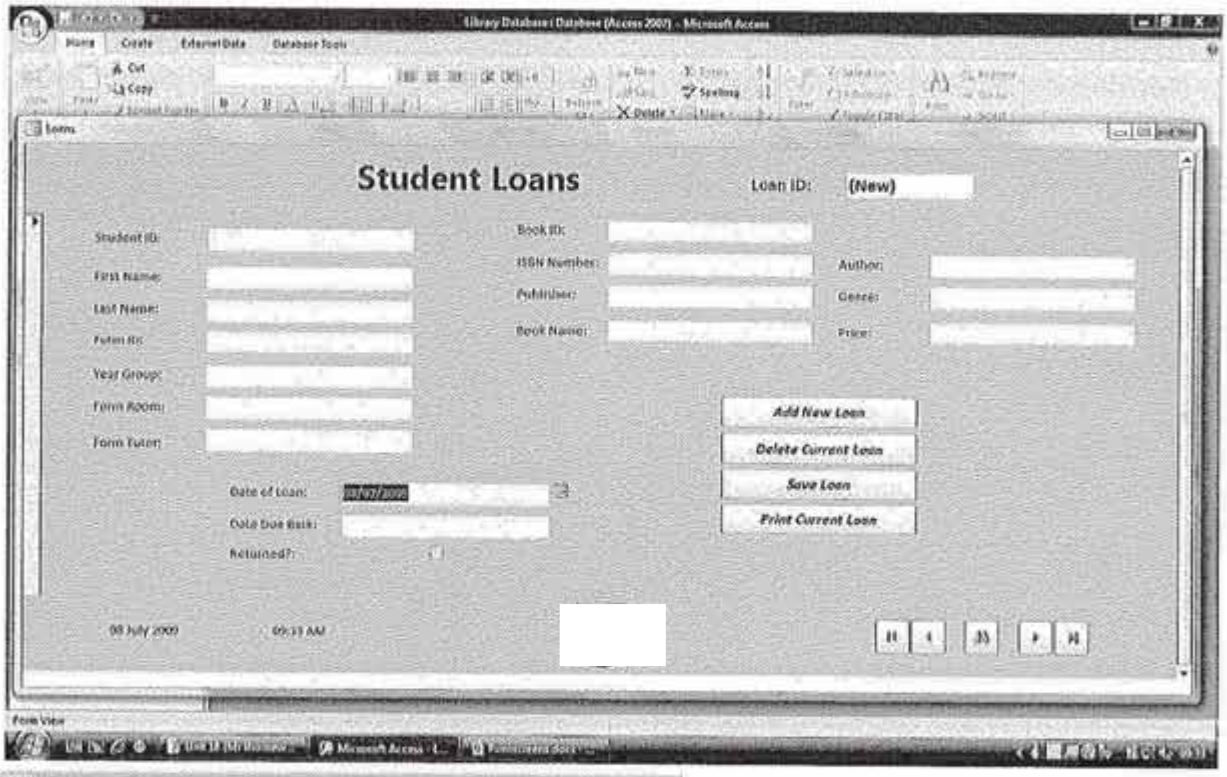
The next part of this form I edited was that I changed the background colour to green. This was done by right-selecting the page and selecting the 'fill/back colour' option. I then also edited the text at the top of the page and re-arranged the forms layout by dragging the sub forms around the page in order to follow the designs that I had made for my form. Finally I then also changed some of the properties for the entire form. To do this I changed the 'Border size' to 'Sizeable' and the option 'Popup' to 'Yes' meaning the windows-style border is placed around the form and that the form will pop up when it is selected from the main menu. This means it can then be closed and below this the main menu should be accessible.

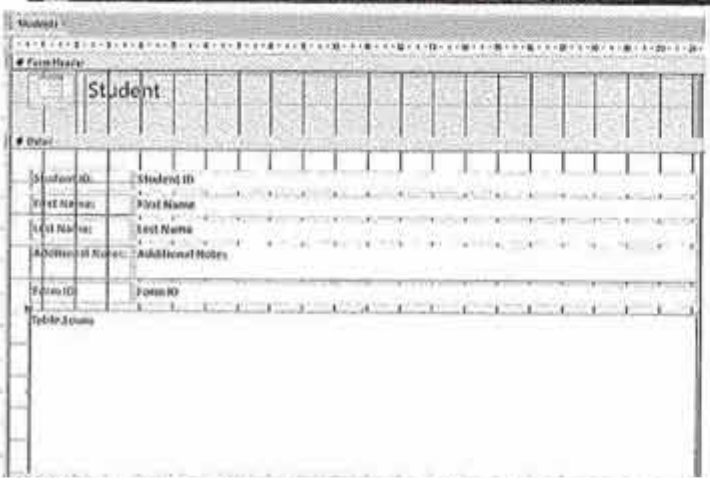


Following this I then decided to add a number of buttons into the form in order to help accessibility around the forms. Firstly within the form 'insert' tab I then selected insert button. Once I had done this I was then given the dialogue box shown to the left. For the first button I selected the 'Record Operations' option followed by the option to 'Add a new record', however for other buttons I used a variety of options such as ones to print and save records. Once this had been done I was asked if the button wanted to be an image or text. In this case I decided to make the buttons a simple button with a text description as this way it will be very clear and therefore keep the database easy to use. Following this I then also repeated these steps in order to add navigation

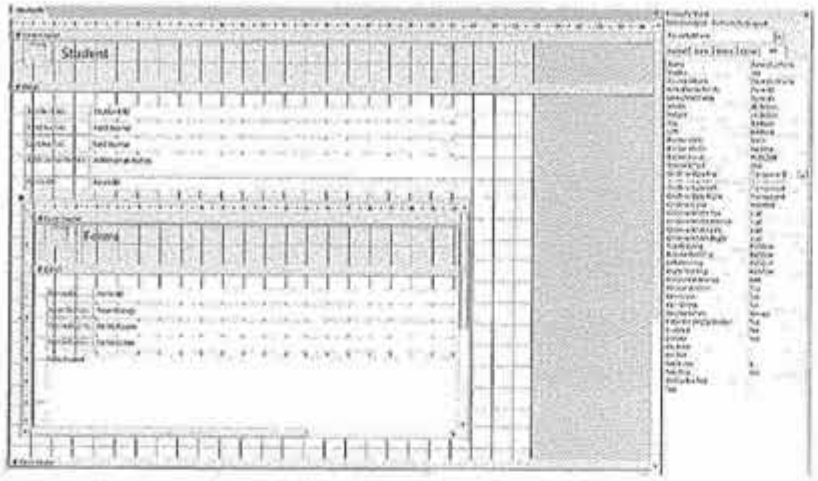
options onto my form, however this time selecting the 'Image' rather than text to be place on as the images used for record selectors should be fairly small yet clear and simple.

Another option that I added within the footer of the form within my database was a time and date option as this application may be open full screen on the librarians computer most of the time, therefore it would be useful for them to have the time and date displayed somewhere on the screen. To do this again under the 'Insert' tab I selected the 'Time and date' option which produced this dialogue box. I then selected the options that I wanted to include both the Time and Date and selected the format that I wished these to be displayed in. I then pressed OK and dragged the created time and date commands into the footer of the form where I wanted them. Finally I then added an Image of the into the footer of the Form so that it was clearly identifiable as a database made for the school. The below screenshot therefore shows the final design of the form that I created after I had finished editing it:

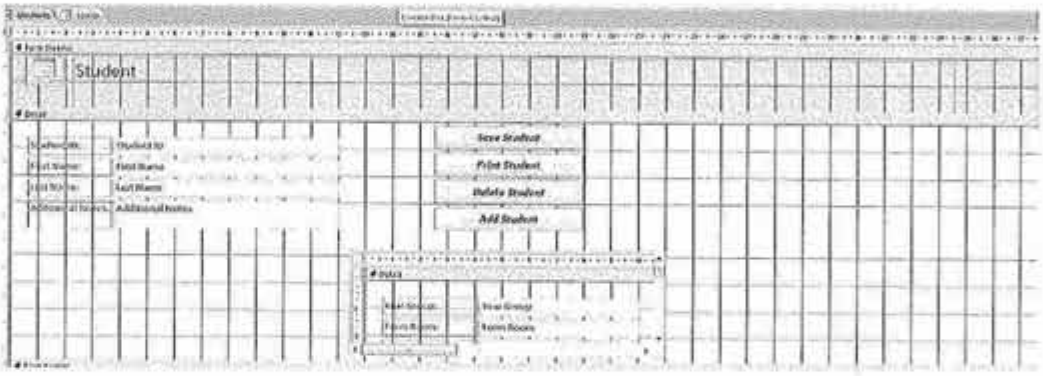


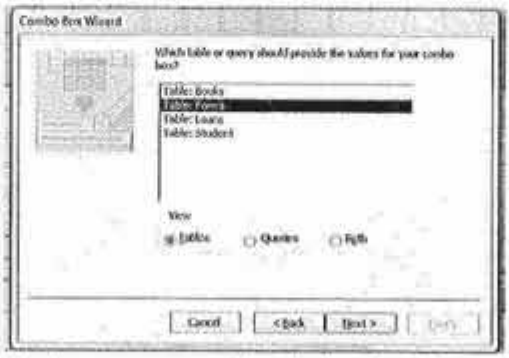


I am now going to begin designing my data input form for my 'Students' Table. This screenshot once again shows the Form as it has been made before I have began to edit it. I have then opened this again in design view in order to manipulate it according to my database form designs.

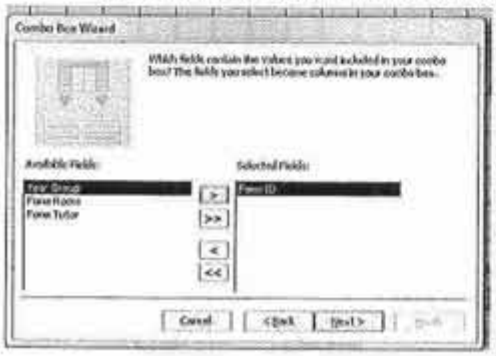


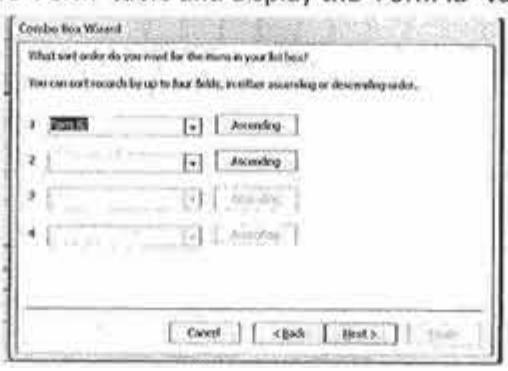
This screenshot shows the first steps that I have made when editing my database. To start, I deleted the pre-made table for 'Loans' which was automatically copied into the form which would show what loans the student had linked to, however this was not how I wished the database to work. After doing this I then selected the 'Forms' table from the list of tables within the application and dragged this into the form, therefore creating or Sub-form. Similar to the form for 'Loans' I then set the properties of this sub-form so that there was no border, that fields within the sub-form were locked and that there were also no scroll bars, record selectors nor navigational buttons shown. Once I had done this I then removed the 'form id' primary key that is used on the form as I wished to change this to a drop-down box in order to make the selection of 'Student form' much easier. I then also repeated the above steps from the 'loans' form in order to add buttons containing links to add, delete, save and print a student record.



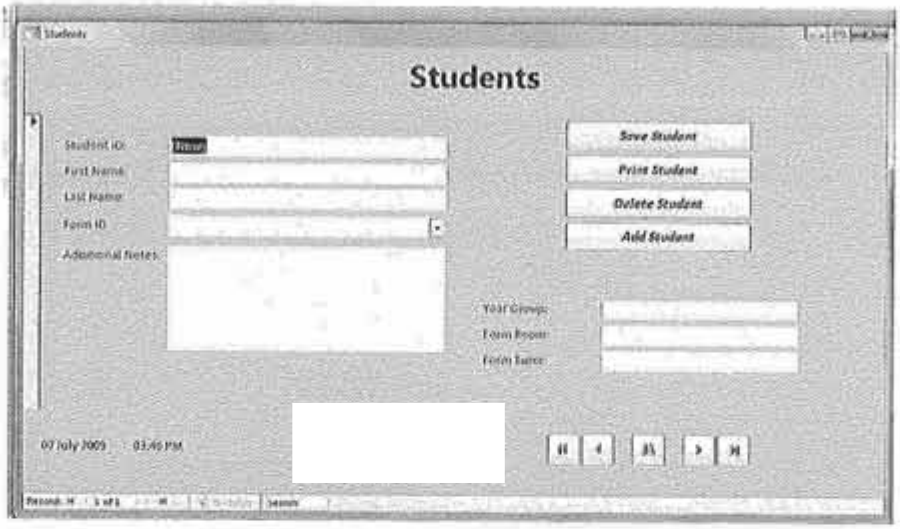


The next part of the form design was to add in a 'combo' box which is going to produce a drop-down list. I therefore selected insert combo box as was produced the wizard in order to set up the combo box. As I wanted this to link to the 'Form' table and display the 'Form ID' values, the first option I selected was that I wanted it to 'Look up the values in a table' Once I had done this I then selected the 'Form' Table. After doing this I then

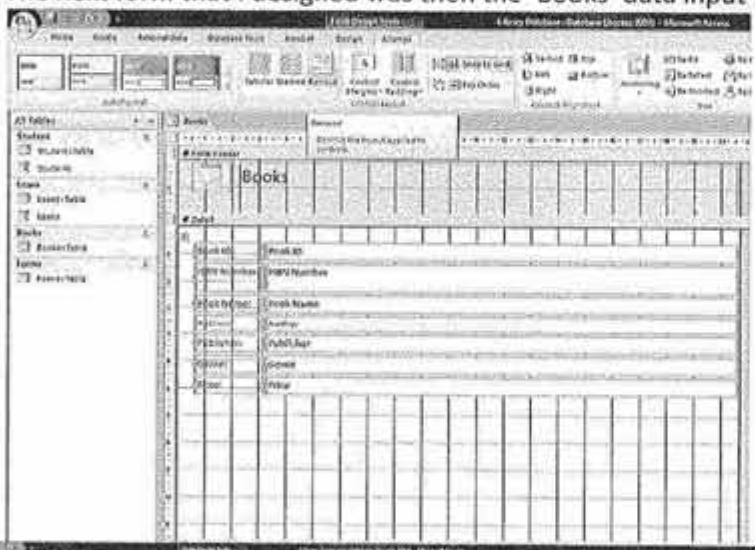


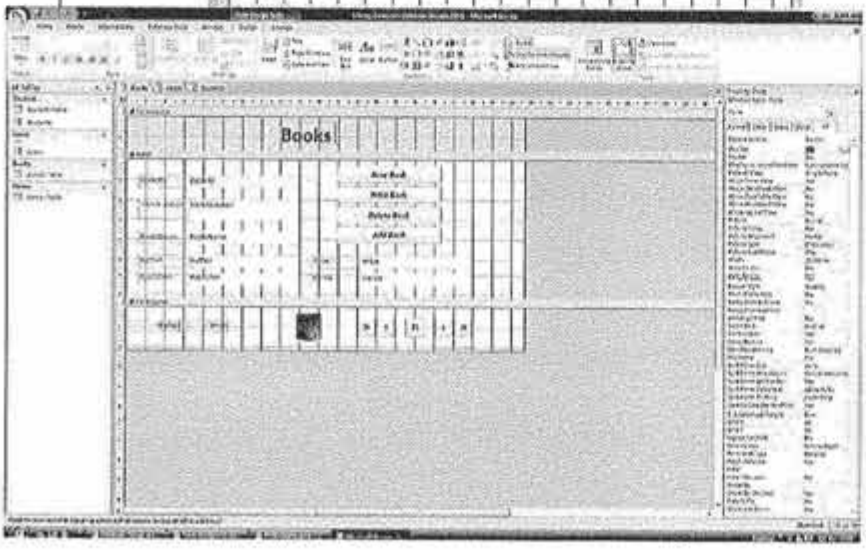


selected the 'Form ID' attribute out of the 'Forms' table, before finally selecting that I wanted values to be displayed in Ascending order, this then added the combo box into my data input form. Once I had done this I then repeated the steps used above in order to add navigational options on my database as well as the relevant background colour, time, date and Deferrers logo, producing the following final data input form:



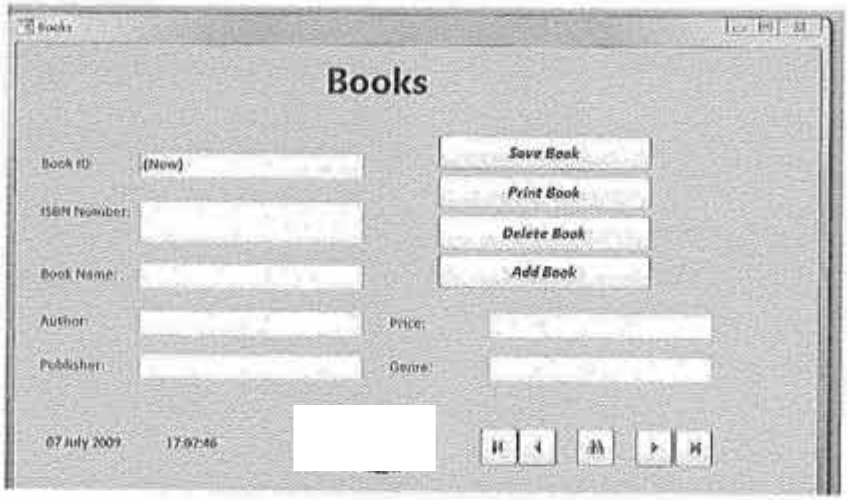
The next form that I designed was then the 'Books' data input form. Again starting from the pre-made form I opened this is design view to being editing it. Once I had done this I firstly when to the option within the ribbon to 'remove' all layout options applied to the table, as this meant that I could not move or re-arrange the attributes within the form on their own- they had to be as a set, therefore by removing the layout options I was able to move each individual part of the form.



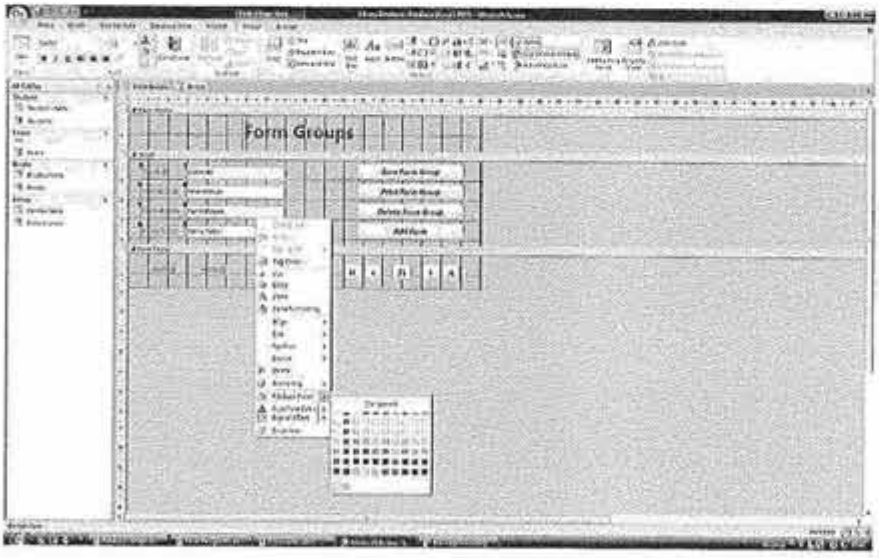


The next part of editing this form was to again use the steps that I have followed above to both add buttons to perform specific tasks such as print or delete the current record as well as navigation buttons. I then also added the time, date and logo into the footer of the form as well. Once again to set the options for the form I also changed the form properties so

that the option 'Popup' was set to 'yes', meaning that the form would be opened in a new window, the border size was also set to sizeable so that the size of this popup window could be changed. Finally I also set the record selectors and navigation options to 'no' and the scroll bars to 'neither' so that the user did not have to scroll around records or have multiple navigation options, as they only need to use the ones that I have created. After then also adding the background colour in order to match my other forms I was left with the following final form that I had created:

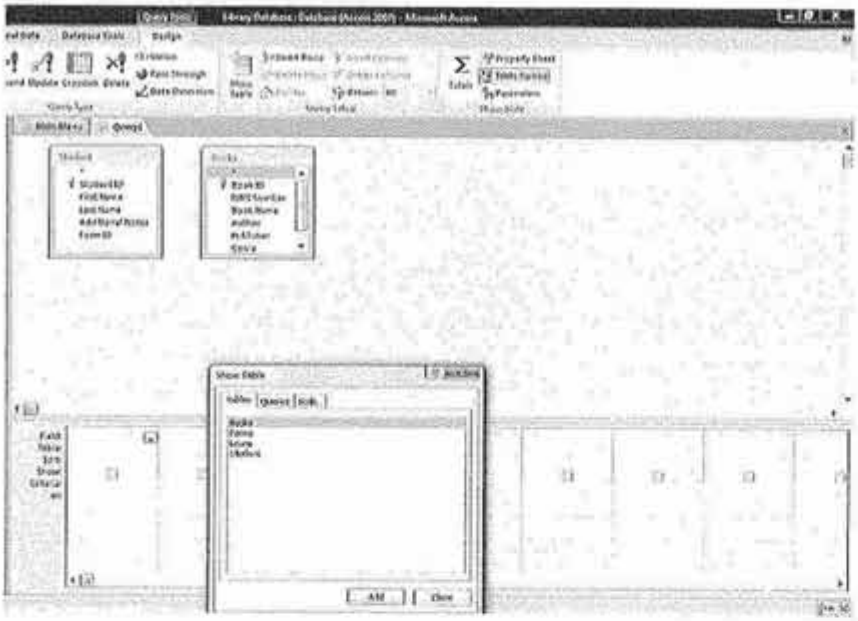


Finally by using the techniques that I have described above in the three previous forms I then designed the data input form for the table 'form groups'. After removing the



formatting options and again adding the record and navigation buttons, time, date and logo, I then also decided on my forms that in order to give it a more professional appearance I could also fill the background colour of the text input files to a light grey/silver rather than leaving them white as this looks much better. After also then adding the background colour and once again changing the properties the same as the other forms in order to remove the pre-made record selectors, scroll bars and to produce the form as a pop up, I had then created my final data input form for the loans table:

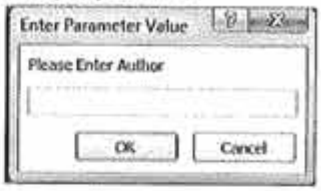
Now that I have completed making my data input forms I am now going to begin creating the queries that I will wish to run within my database. In order to start creating the queries for my database I firstly selected the option to create a new query and to create it within design view as this will allow me to make much more precise and complex queries. To start, I firstly added all of the tables to the query design view. From here I could then select attributes within the table that I wish to be included within the query as these can be used to specify parameters for the query or to simply display attribute results dependant on the query when it is run.



The first query that I decided to do was a simple one that would 'Find all books due back today'. After importing all of the required fields I wished the query to display. The date due back was added to this list so that I could make the query however the check box for 'show' was not selected meaning this would not be shown on the output query. For the criteria of this simple query I input the formula 'Now()'. This command will take the system date and therefore perform the required option to produce the query correctly. This query may be useful for the librarians as they can produce a list of books they can expect back today.

The next query that I decided to create was a parameter query. This query will allow the librarians to find a book by a particular author, this means should someone say that they like a particular author but do not know the name of a book the database can be quickly searched. In order to complete a parameter query I have input the

text 'Please enter author' in the criteria for the 'author' field within square brackets. This means that when this query is run the message 'Please enter author' is displayed in a dialogue box. When the response is entered, this is then used to run the query, therefore allowing the librarians to search for any author in the database rather than pre-specified ones.



Following this I then decided to produce a more complex query to run. This query should produce a list of all overdue books for a particular form. The librarians may wish to use this query regularly as they could print these results in order to send a form teacher a list of all their students who have overdue books. In order to run this

query correctly I therefore had to use three different criteria to be run at the same time in order to find the relevant results. Firstly I set the field 'Returned' to false meaning the books must have not been returned back to the library. I then set the 'date due back' to '<now()' meaning results displayed must now have been due back before the query is run and must not have been returned. I then also entered a parameter query within the 'Form ID' field with the criteria 'Please enter form ID' so that the ID number of any form can be searched for overdue books.

The next query I run was in order to find all students by both first and last name. I decided to add this query to the database as people will need their student ID number in order to take out a book when using the loans form. It is therefore inevitable that students are at

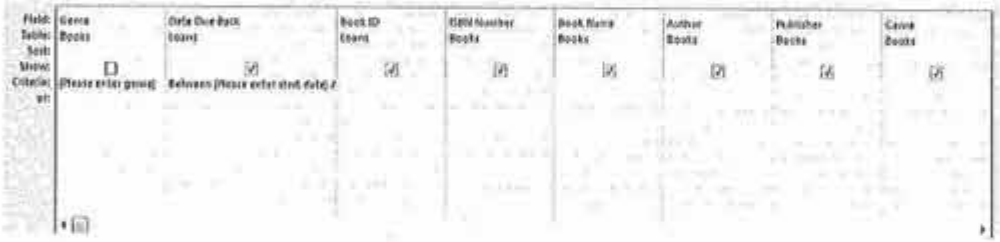
some point going to have forgotten their own ID number when wishing to take a book out. Therefore by running this query on the Student table the librarians will be able to find their information including their student ID number. To run this query I again did two parameter criteria within on query meaning only students with the entered first AND last name would be displayed, otherwise first name may produce a large list of common names and last name may produce results for brothers/sisters within the school as this is also not going to be unique.

Following this I then decided to produce another query which would show students who have overdue books by at least two weeks. This type of query may be useful to the librarians as this many now be classed as severely late as they are already likely to have had reminders

for example using the 'Overdue books for form' query before. This query is therefore important should any larger overdue cases need to be passed on to a higher level member

of staff such as head of the year. In order to produce this query I again needed to use two search criteria. The first of these simply stated that the book had not been returned by the value being 'False' as otherwise books returned will still show on the results. The other criteria for the query was that the date due back was '<=now()-14' meaning that the date the book was due back must be 'less than or equal to now minus 14 days', therefore at least 2 weeks ago.

After this query was complete I then decided to do a more complex parameter query in order to find books of a particular genre that were due back within two dates. This may be



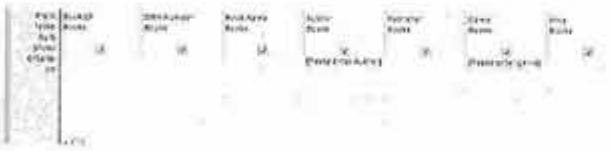
useful for the librarians for example if they do not have any horror books left and a student needs one between two dates for a project, as they will be able to check if any books due back in within that time or between now and when the project starts. This therefore required two specific query criteria within the design view for the query. The first of these was that the 'Genre' criteria was '[Please enter genre] so that the user can search for a specific genre of their choice. The second criteria used was 'between [please enter start date] and [please enter end date]'. This second criteria therefore used parameter tools so that the user can enter their own dates and the 'between' command to ensure the query searched between the two input values.

A simpler query that I decided may be useful for the Librarians to use was one to find all books that were on loan to a particular student. They may wish to do this if a student simply forgets what books that they have on loan or may wish to use it to check how many a student has got if they decide to impose limits on how many can be taken at once. In order to do this I again needed to use two different criteria at the same time to produce the required results. After adding in the necessary fields I set the criteria to the 'Returned' field to 'False' to ensure that the book had not been returned. I

then set the student ID criteria to [Please enter ID] so that by using a parameter query the librarian can enter any students ID to check what books they have on loan.

Another query I then decided to produce was a parameter query in order to find books that were of a particular genre. This query will be useful to the librarians as they are likely at some point to be asked to recommend a book on a particular subject, therefore they may wish use this query in order to look at what they have got. In order to do this I have again used a parameter query by entering in the criteria section [Please enter Genre} so that a dialogue box is displayed to which they can enter the Genre of the book.

The next query that I produced was again another parameter query as this is designed to find all books that are due back on a particular date. The librarians may wish to do this if they wish to have an idea of what books are due back on any given date; therefore they can then run this query.

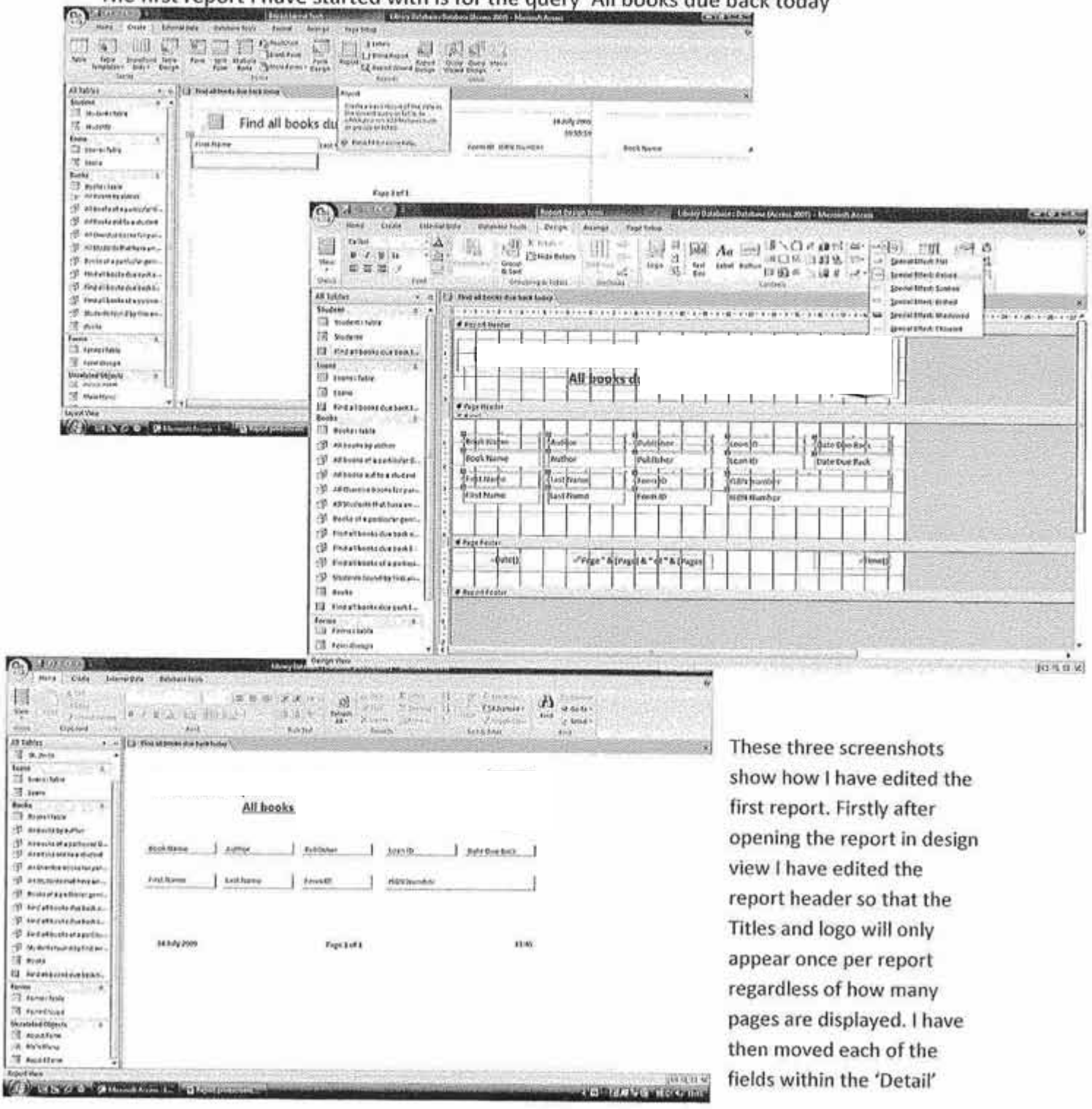


For my Final query I have then produced another query in order to find books of a particular genre by a particular author. This may be used by the librarians as they may also be told that someone likes comedy books by a particular author, and asked if they have any other books that are similar to a book they have had that is therefore by the same author and of the same genre. As this is going to be a variable and will change from person to person I have therefore set up a query which again has two parameters which will display the messages 'Please enter Author' and 'Please enter Genre'

I am now going to look at the reports that I have created which will display the results of a query in a user-friendly way rather than the end user having to look at a table, this means that functionality can be increased and therefore the database will be more effective.

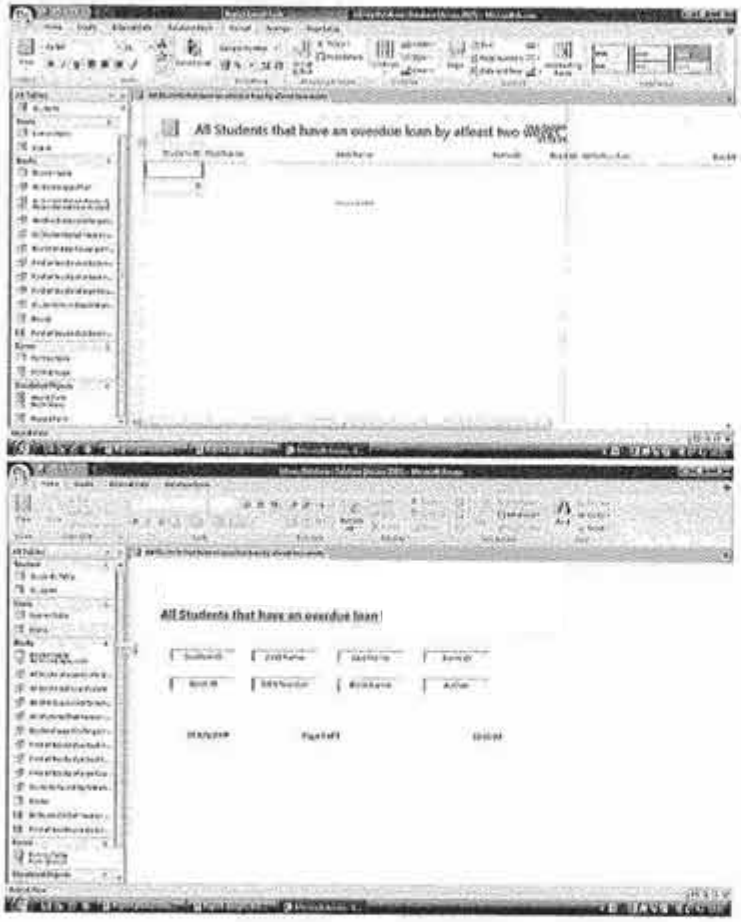
To do this I have firstly started by opening the report in a computer-generated format and then opened this is design view where I will then edit the content in order to manipulate the way that it works.

The first report I have started with is for the query 'All books due back today'



These three screenshots show how I have edited the first report. Firstly after opening the report in design view I have edited the report header so that the Titles and logo will only appear once per report regardless of how many pages are displayed. I have then moved each of the fields within the 'Detail'

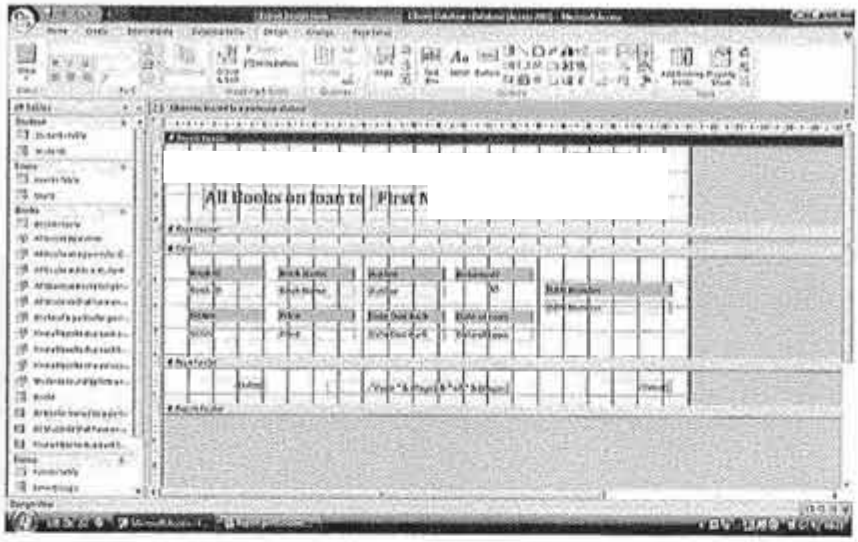
section meaning the title for each field and its data will appear once for each individual record. I have then entered within the page footer content that will be printed at the bottom of every page of the report, again regardless of how many pages there are. This formula's call the system date on the left side of the report and the system time on the right side of the report. In the centre I have also inserted the formula required in order to display what page number of the report each page is and how many pages this is out of. As the second screenshot shows I have also changed the 'special' effects to 'raised' which means the titles for each field appear to be raised off the page, therefore differentiating each record and making clear the difference between the headers and the actual data.



I then followed a very similar process in order to produce the report for the query 'All students with an overdue loan by at least two weeks'. This report will be very useful to the librarians as they may wish to print this and pass it directly on to higher members of staff so that they know who owes the school books back. I have again removed all layout options and re-arranged the page and included the same footer for each page of the report. To again differentiate between the headers for each field I have again used the 'special effects' option however this time I have selected the option 'Sunken' as constantly using the same special effect would make all reports look the same which would not be particularly interesting.

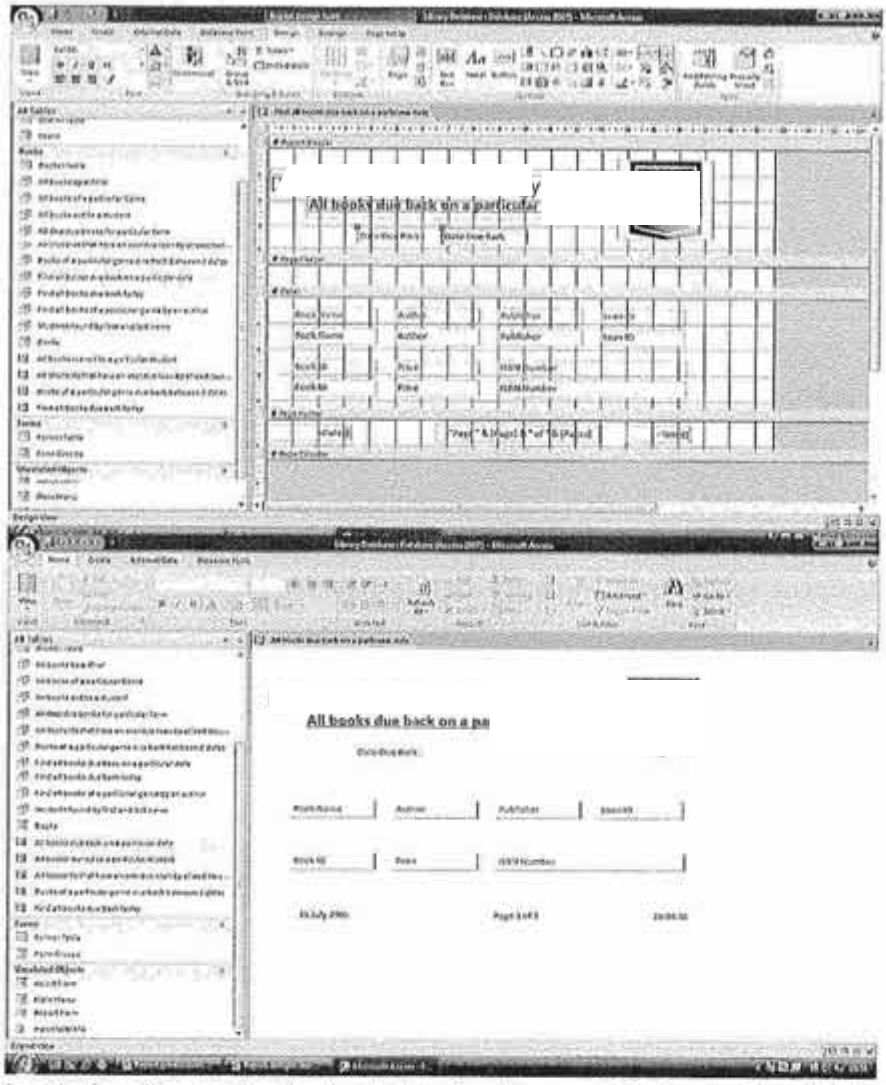
The next report I created was to show 'All books on loan to a particular student'. After looking at the automatically generated report there were many problems with it. Firstly I changed the report header to again make it consistent with other reports that are being produced off this system. I then also saw that the report was displaying lots of repeated data as it was going to display the 'first name' and 'last name' fields on the report for each entry, however this report is for one particular

student which has been entered to the system. Therefore I then deleted these fields out of the 'detail section' but however left the headers in place and moved these to the 'report header' section



as this meant the First and last name of the student entered would be shown at the top of the report, therefore making the report more clearly identifiable. I then once again used the same footer as I have done for the two previous reports as this provides useful information for the user.

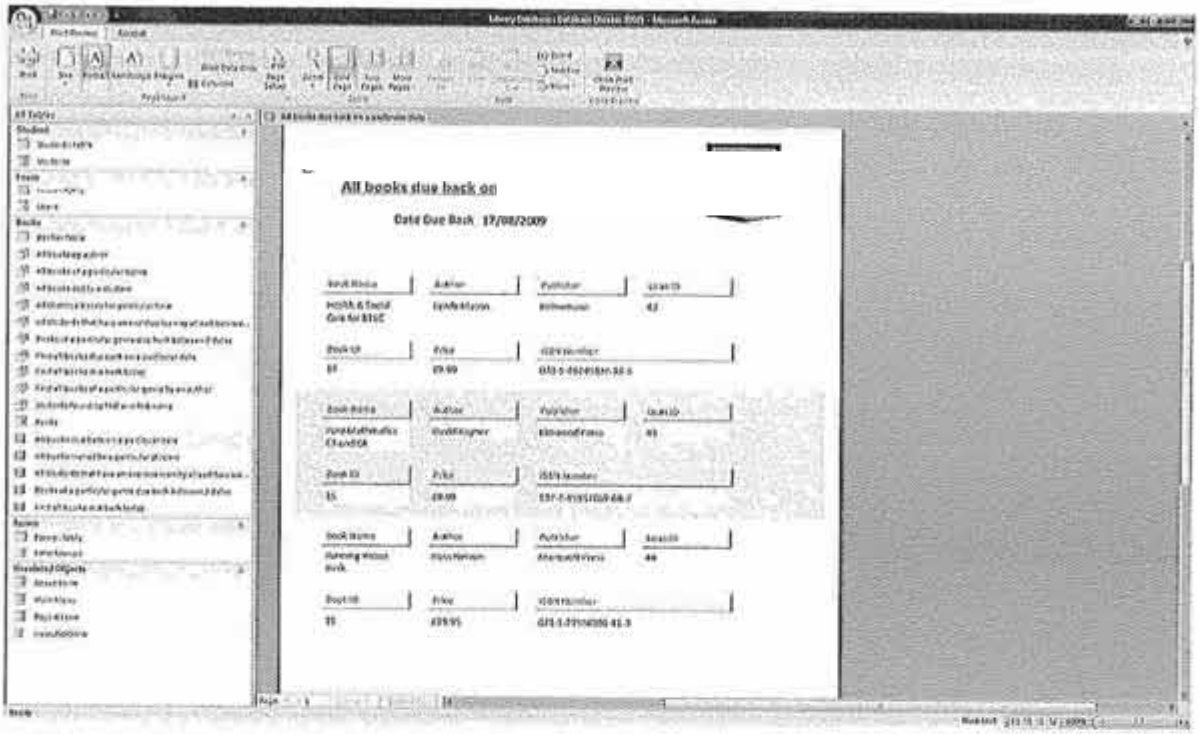
After producing this report I then also considered the fact that there may also be a case when a report is run that the database may not have any results to be displayed, for example if a student did not have any loans to display, therefore I created a macro which would produce a message to inform the user that there were no results to display for the query/report that was run. To do this in the design tab I created a new macro and entered the action as a 'MsgBox' in order to produce a message box. Under the 'Arguments' section I then set the properties that I wished the message box to have, therefore giving it a title of '[ıse system' and selection the option to beep as 'yes' so that an audible alert is given and the message to be shown is 'There are no results to display'. To add this into every report that I made I then opened the 'properties' sheet for each report and under the option 'On No Data' selected the macro that I had created.



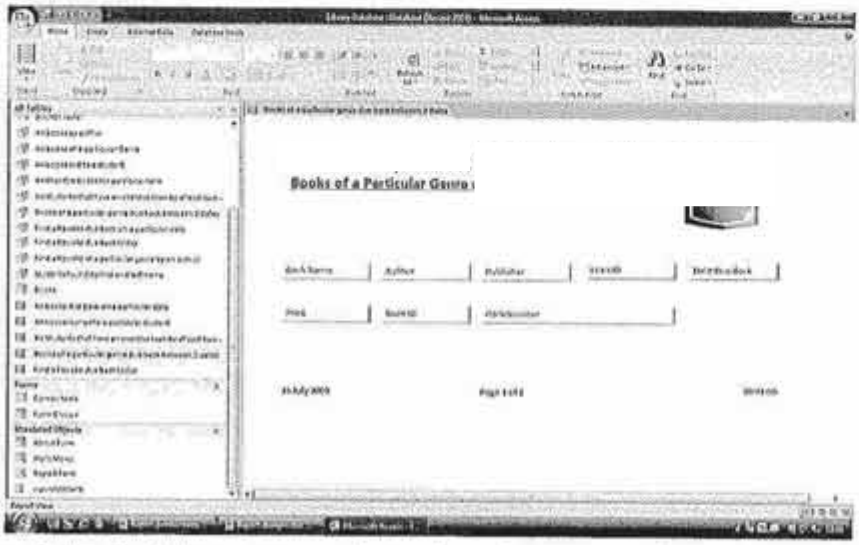
Following this I then began designing the reports for the query 'All books due back on a particular date'. After opening this in design view I once again linked the 'No data' macro and input the same footer and logo as my other reports. Following this I then re-arranged the fields shown within the 'Detail' section in order to display the report exactly how I had designed. Once again, as this report was for all books due on a date, the date field is going to be the same for every entry, meaning there would be repeated data across the report. Therefore I again decided to move this option to the top of my report into the 'report header' section so the due back date for the report is clearly identifiable at the top of the report.

I am now going to look at how to produce horizontal stripes across my reports in order to differentiate the records on a report. This is because I can see from the reports above that as each record has its own header and

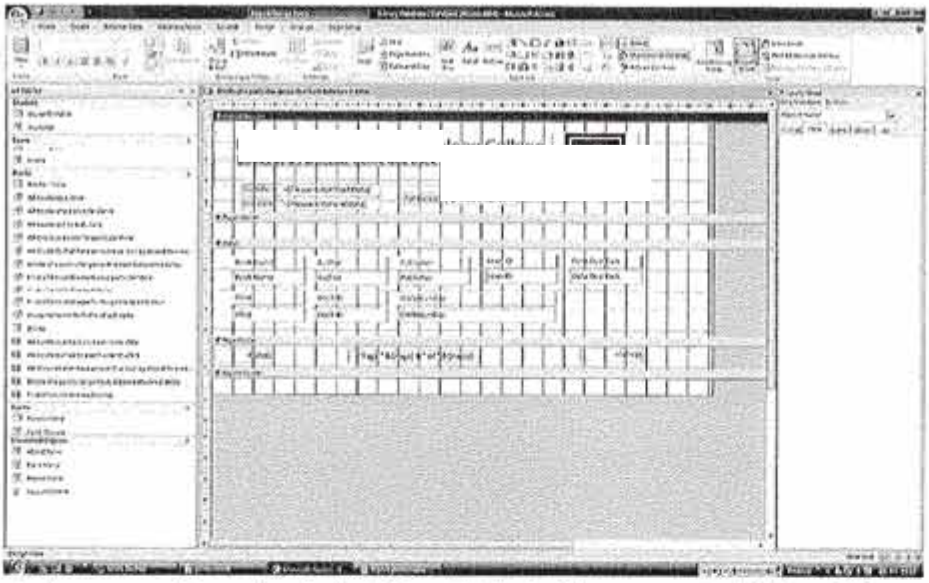
value, the start and end of each record could easily get confused, whereas if each record appeared to alternate in colour it would be much easier to see which record was which. To do this I am going to use a short Visual basic script defining the two colours I wish to use and then link this to the report. This Visual-Basic created macro can then be linked into the Report again by using the properties sheet and selecting the macro under the 'On format' option within the 'Event' Tab.



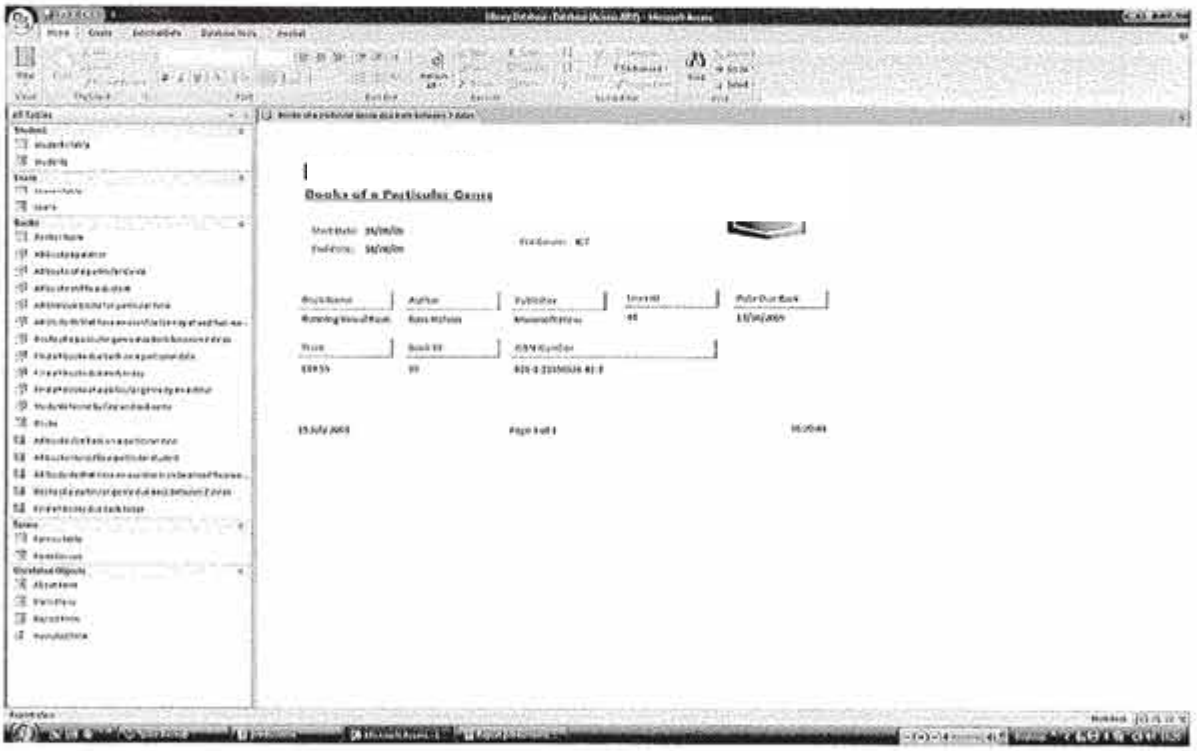
This screenshot therefore shows the final report with my macro working as each record now displayed will now alternate in colour making it much easier to identify each different record shown within the report.

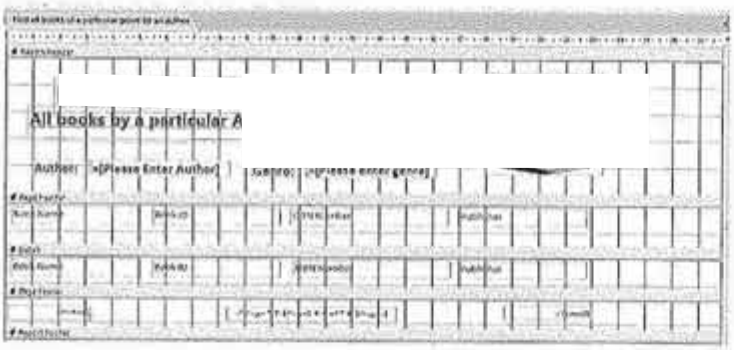


I then once again produced another report of a similar format in order to show the query 'Books of a particular gentre due back between two dates'. This was again done by using the same report footer and header and simply adapting the titles in order to fit this report. This again had the 'raised' special effects used for the headers of each field and also had the macro running to show the message for no data and to alternate the colour of each entry. Looking at this design I then decided that once printed you may not know what two dates were chosen and what genre of book was chosen as I had completely removed these from the report. Therefore I then selected the option to open this report in design view and then re-edited it

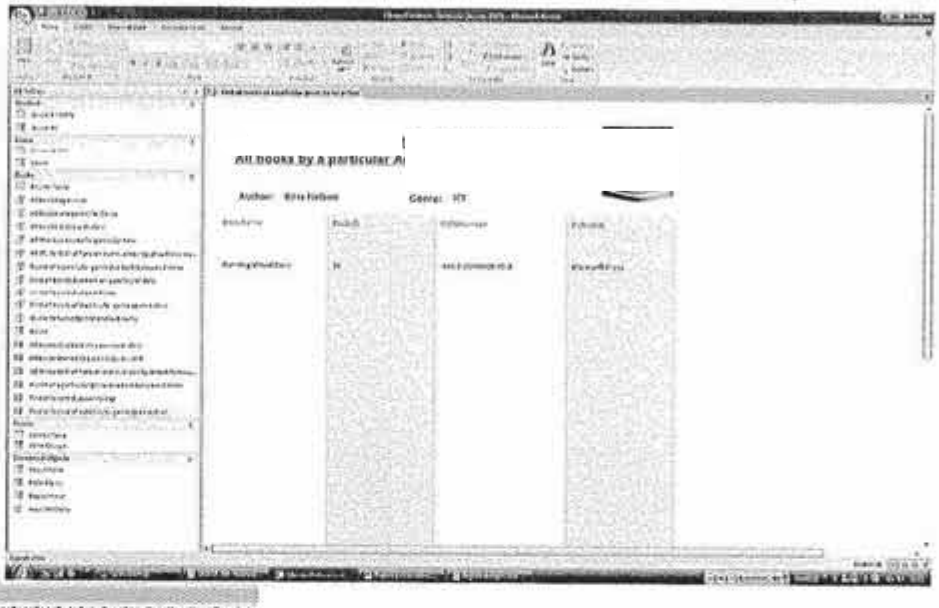


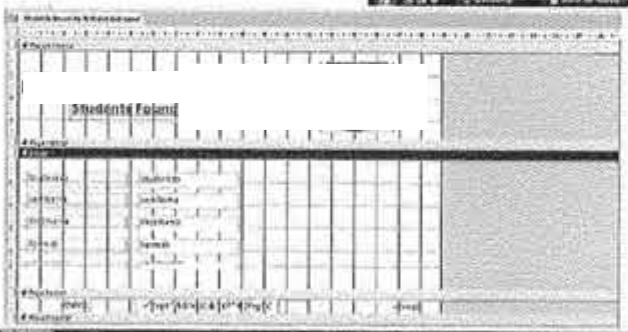
in order to add these details In. I decided to input these fields into the 'report' header as this way they are only ever displayed once at the top of the first printed page. This makes the report much more easilyidentifyable. I therefore had to use the code '=[Please enter Genre]' and similar parameter codes to link these fields to the parameter fields within my query. I was then left with the below final report once parameter values were entered:



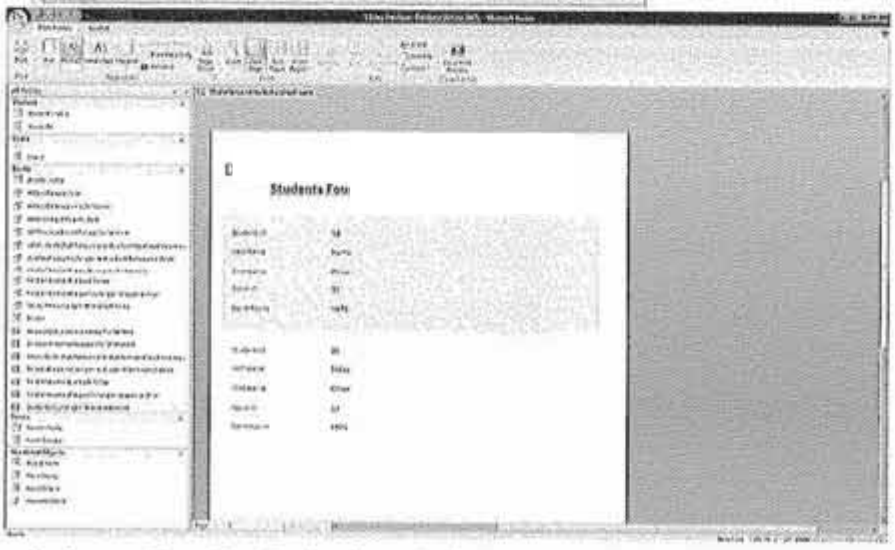


Following this I then moved on to produce a report for the query 'All books by a particular author and of a particular genre'. This report was similar to the previous report that I had created, however I devided to change the layout so that the titles for each field only appeared once and that the colour banding across records was now vertical rather than horizontal to separate each column. In order to do this type of banding I then simply added a colour block behind the fields on my design view, I did not have to use any code as this is only needed to alternate records which I am no longer doing, therefore this macro was not used on this design. I then added the two fields again into the header of the report to show the Genre and Author that had been selected as part of the parameter values.

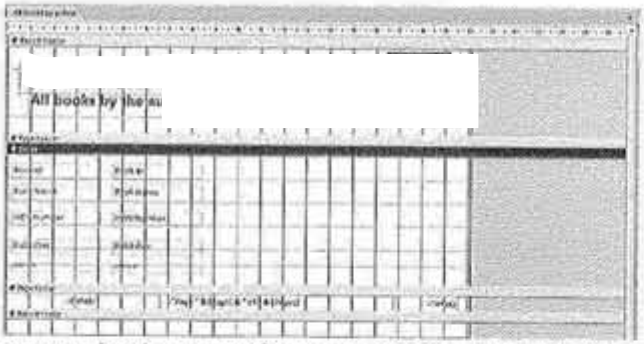




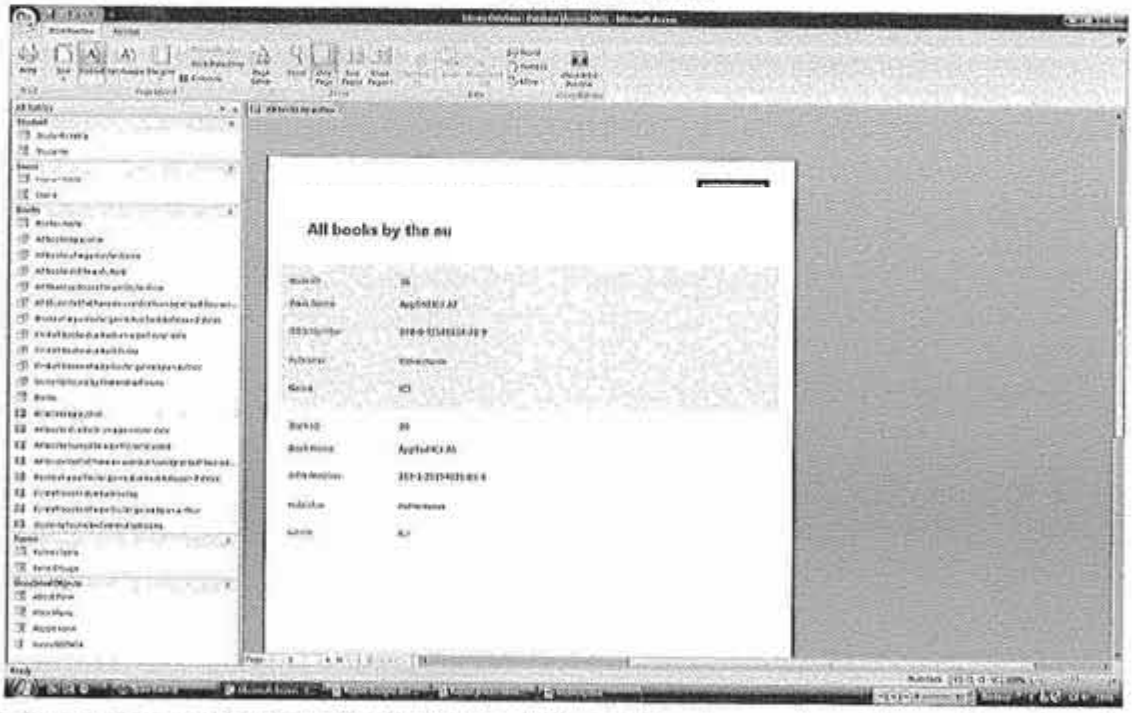
The next report that I have designed is one that will display the results for students found by name, in order to do this I have again included the time, date and page count within the footer of the page. I have then created the report header with the logo and the name of the report. I then simply spread out the fields that I wished to display within thet 'deatil' section of the header and once again included the macro that I have used within the 'on format' property of the page so that the colour of each record displayed is alternated. This may be

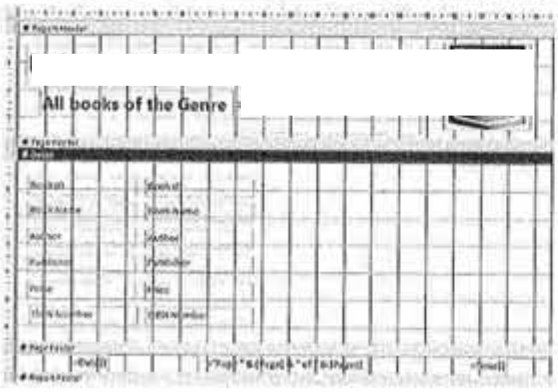


because the librarians may with to seach for someone by first name or last name and therefore the report may produce more than one result, therefore improving the efficeincy of the reports as each record can be more easily identified.

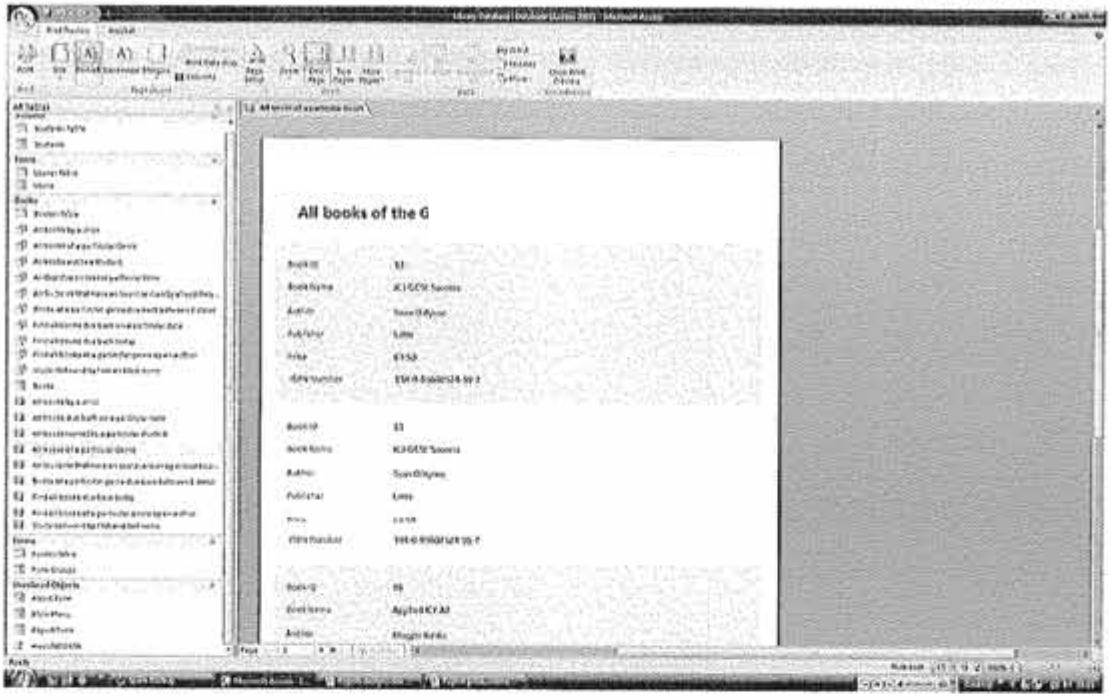


The next report that I then designed was for the query "All books by a particular author". This report was again deisnged in a similar way with the report footer being deisngned the same as all other reports and the header clearly showing the title and I therefore showing consistency throughout all of the reports meaning they can clearly be seen as having come from the same system. I have also once again included the use of the macro I have found in order to alternate the record colours to make it easier to read. Finally in order to avoid repeating data on the form I removed the 'Author name' field as this was going to be the same and set this parameter as part of the title, meaning when someone entered an Author name in the parameter query, this name is shown on the associated report.

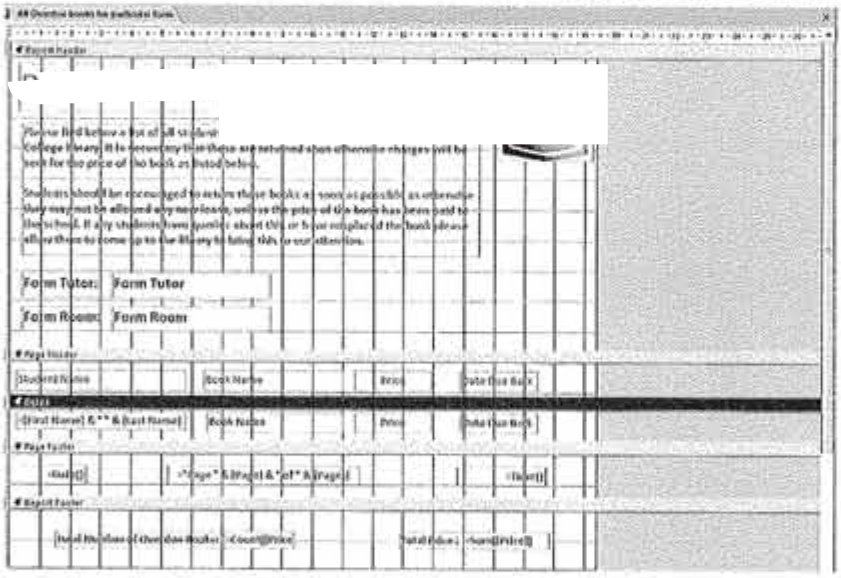




This penultimate report was the produced in order to show the query "all books of a particular genre". To do this once again similarly to the above report the macro was addded along with the headers and footers on the page. I then also added the parameter "=[please enter genre] as when this is generated in order for the query to run, the input value will display here rather than on every result for the query as this would mean lots of repeated data. The below screenshot therefore shows the final report that I designed and after running the query on 'ICT' the results that were displayed:

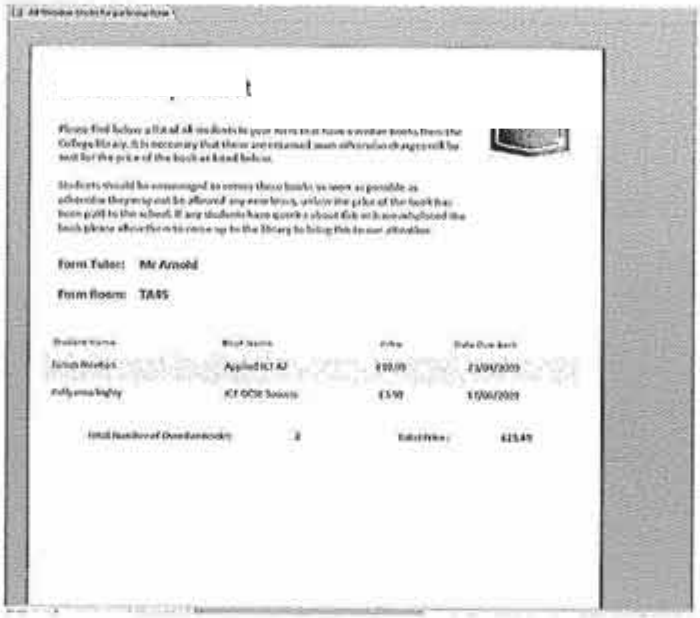


I am now going to look at my very final report design for my database. This report is linked to the query "All overdue books for a form". This is likely to be one of the most commonly used reports as

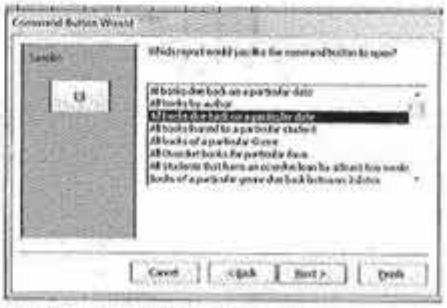


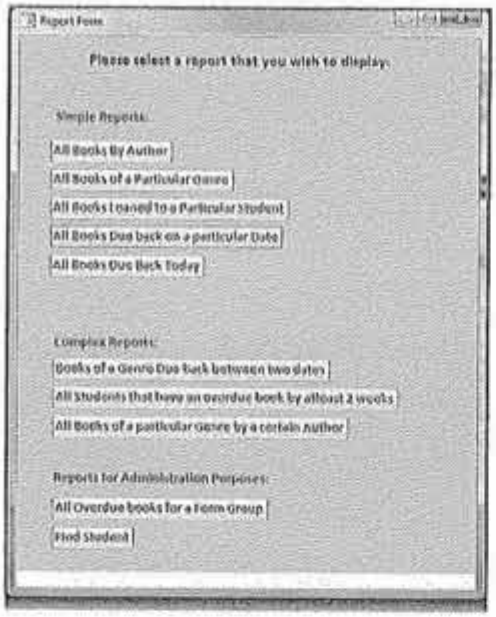
this will be run periodically E.G. Every week and passed along to form tutors so that they can individually remind students of the books they must return. Thisreport is one of the most complex that I have created for my database system in order to show the reuslts of a query. Firstly I have entered into the report header the name of the school and logo in order to create consistancy across each of the report designs that I have done, following this I have then created text to be put at the top of the report which is for the attention of form tutors, informning them of what this report is for and what they should do with it. I have then included the Form Tutor and Form room fields once within the report header as these only need to be displayed one – they will be the same for all results. By placing this information in the header it is only ever given once rather than at the top of every page. Content to be placed at the top of every page were however the titles for each field, as records will be displayed in rows lined up under these coloumns. Once again the 'On format' event was linked to my macro so that each record was alternated in colour. Within the detail section I then included the fields where results will be displayed which may be very little or multiple pages long. In the even of it being more than one page, the report header will not be shown but the titles will be shown at the top of every page in order to make it easier for the user to understand. Moving on from this I once again entered the time, date and page number within the "page footer" section as

this means it will appear once at the bottom of evey page, therefore clearly showing when it was produced and helping to keep the document in order. Finally within the report footer I created two sum formulas which will be displayed once at the very end of the report. The first one was a "count" formula which was "=Count([Price])" which simply therefore counted the number of records to show the total amount of books the entire form has overdue. The second one of these was "=Sum([Price])" which totaled the price of all records that were shown in order to give a total price of all the overdue books within a form. This may be useful as form tutors will be able to have an idea how bad their form is returning books and this may make form tutors complete slightly to get the numbers as low as possible, therefore encouraging people to return overdue books.



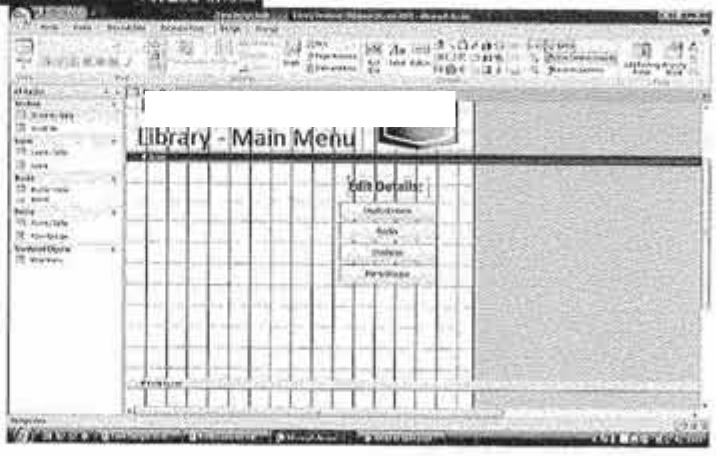
Now that I had created my tables, Queries and reports in order for my database to run, I then began to create the menus that I was going to use to link the content together. By using these menus and linking them to forms I had created I will be completely hiding the background database to the user and providing them with an easy-to-use, efficient way to enter data and search for data within the database. The first menu that I began to create was therefore the 'Reports' menu where there would be numerous buttons linking to the reports that I had just created. To begin I started by opening a blank form and inserting a text title and changed the background colour to green. Following this I then selected the "Button" button in order to start inserting buttons onto the form. This option opened up a wizard which allowed me to select that I wished the button to open a report, after selecting next I then continued by selecting the report that I wished the button to display. After doing this several times in order to create buttons for adding every report, I then created some text sub-titles and had therefore created the following 'reports' menu:

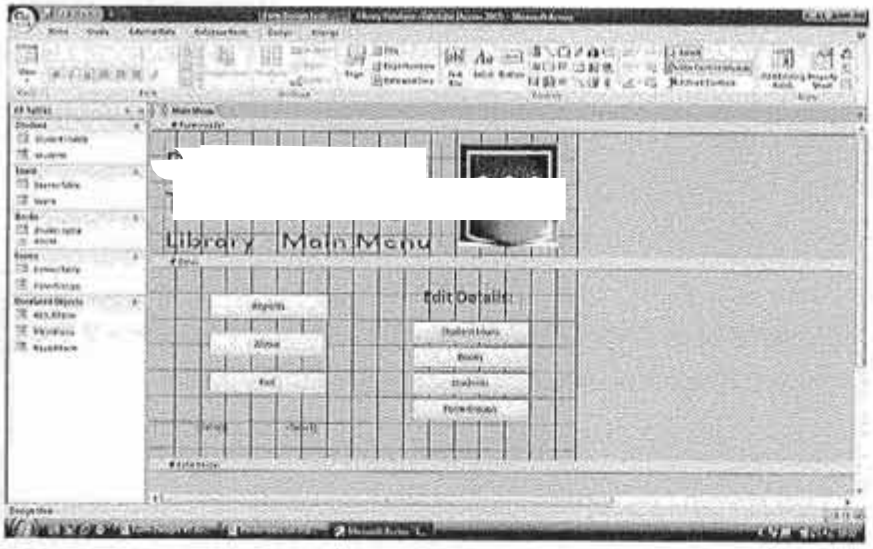




Once I had completed making this "reports" menu I then also began to create an "About" menu where I could provide basic information about the database such as when it was created/updated and who had designed and made it. In order to do this similarly to the "reports" design I created another blank form and simply entered text into text boxes on it, again producing this menu:

Now that I have produced the two sub-forms that I wish to come off my main menu, I am going to create the main menu for my database. This main menu will tie in all elements of my database and is where all the features of my database will be accessible from for the user, therefore making it simple and easy to use. After starting with a blank form I started by entering the title that I wished to display across the top of the main menu. Following this I then inserted a button onto the form and selected that I wished the command of the button this time to be to open a form. I repeated these steps in order to add the buttons for the forms for "Students", "Books", "Loans" and "Form Groups".



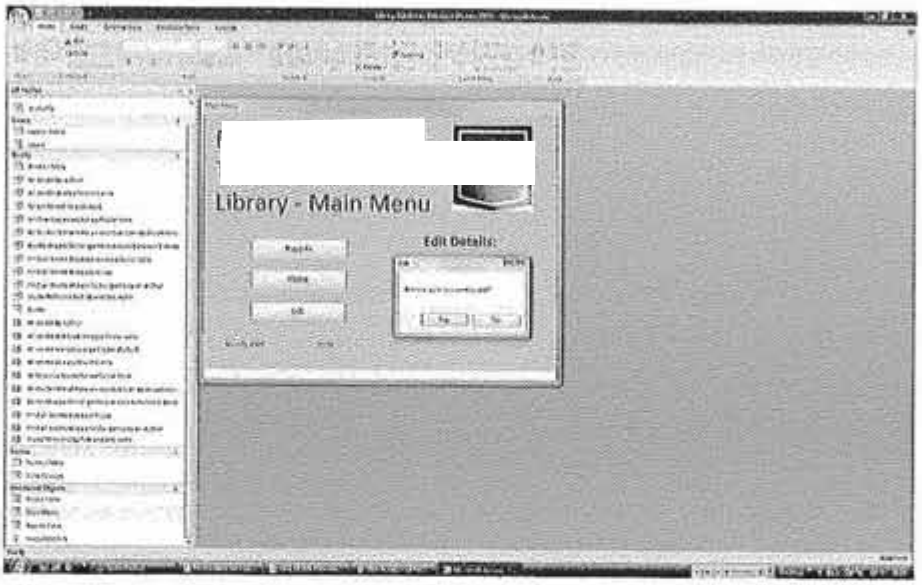


Once I had added these buttons I then began to add the other buttons for my form which linked to my other forms "reports" and "about" along with a command button "exit" to close the database. I then also added into the bottom of this form a Footer with the formulas "=Date()" and "=Time()" to show the system date and time to the librarians whilst they are using the system as this is something they may find useful. Finally I then also added the green background colour to make it consistent with other reports and forms along with the other menus that I had created.

In order to improve the efficiency in which the database worked I then edited the way the timer

worked as it did not update – in its current format it simply showed the time at which the database was loaded which will not be useful as it may be loaded and kept open all day. Therefore In order to change this I replaced the time formula with a visual Basic macro code which also called the time from the system clock. Once I had done this I then put a text box in place of where the timer previously was and set its "On timer" value to be this macro, which was to display the time. By doing this I could then set the timer interval to 1000 as this is measured in milliseconds which meant that the code would then refresh every second, meaning the time should constantly be being updated.

Finally another feature that I decided to add was a warning message when the exit button was selected should it have been selected by mistake, as currently this would then be too late and the database would close. I therefore then used another visual basic macro code in order to produce a dialogue box with the message "Are you sure you want to exit?" with the No button then returning back to the main menu and the Yes button allowing the action to continue to close the database. In order to link this to the Exit button I simply set the macro up and then under the 'properties' sheet for the exit button selected the macro under the even "On click", meaning when the button is clicked the macro is run. This therefore left the following message on my final Main Menu:

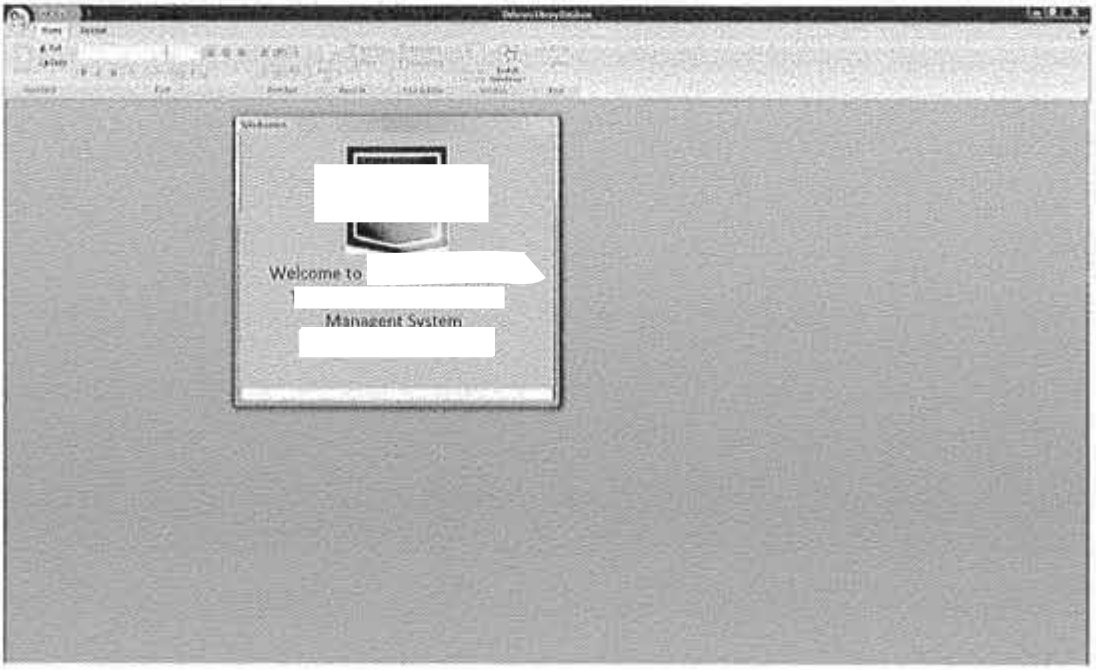


The final part of my database creation was to create a 'Splash screen' which will display for a few seconds before automatically closing which is simply used to welcome people to the database. The first part of this "Splash screen" was to simply start with a blank form. I then sized this to the size that I wanted the menu to be ad began adding content and a background colour to the form. Once I had done this I then used created another macro within access. In order to create the macro I entered the "arguments"

section of the macro firstly to 'Close' this form. I then set the second sequence within the macro to 'Open' the main menu form. Once I had done this I named the macro as "macrosplash" so that it was easily identifiable.

In order to then enable the macro within the properties page I entered the macro to be run be run under the event "on timer" and set the time interval to 2500 milliseconds (2.5 seconds),meaning the Splash screen will be shown for 2.5 seconds then close and open the main menu. Finally in order to further protect the database I decided to lock the database down as this will prevent the user from making changes that they do not wish to make.

In order to do this I opened the options for my database software and set restrictions on the database so that once opened only limited things could be changed. Firstly I disabled the option 'Display navigation pane' as this meant that the user was forced to use the menus and navigational systems that I had created rather than the normal pane where they would have been to access the hidden tables. I then also removed the option "enable design changes for tables on datasheet view" as I did not want the tables to be editable. After doing this, saving the database and re-opening it (as an end user would) I was therefore presented with the following screen and splash screen for 2.5 seconds. The changed I have made clearly show how the database can now only be changed by using the options within menus that I have created e.g. Data input forms.



Tables

Student ID	First Name	Last Name	Additional Notes	Form ID
16				15
17				15
18				15
19				16
20				16
21				17
22				18
23				18
24				19
25				19
26				20
27			Damaged Last Book	20
28				22
29				23
30				23
31				24
32				24
33				24
34				25
35				25
36				25
37				25
38				25
39				25
40			Fails to return books regularly	26
41				26
42				26
43				27
44				27
45				27
46				27

Form ID	Year Group	Form Room	Form Tutor
15	10		Mr
16	10		Mrs
17	10		Mr
18	11		Mrs
19	11		Mrs
20	11		Mr
22	12		Mis
23	12		Mrs
24	12		Mr
25	13		Mrs
26	13		Mr
27	13		Mrs

Book ID	ISBN Number	Book Name	Author	Publisher	Genre	Price
13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	Letts	ICT	£3.50
14	043-5-46245837-36-5	Health & Social Care for BTEC	Lynda Mason	Heinemann	Health & Social care	£9.99
15	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner	Elmwood Press	Maths	£9.99
16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks	Heinemann	ICT	£19.99
17	978-0-00728816-82-5	Apache	Ed Macy	Harper Press	Non-Fiction	£19.99
18	978-0-74879846-95-4	As Business Studies	Steven Foden	Nelson Thornes	Business Studies	£14.99
19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson	Microsoft Press	ICT	£19.95
20	212-1-25154121-65-4	Applied ICT AS	Maggie Banks	Heinemann	ICT	£9.99
21	894-2-21415041-51-5	Advanced Databases	Ross Nelson	Microsoft Press	ICT	£24.95
23	837-4-92749568-73-9	A Level Maths	David Rayner	Heinemann	Maths	£35.00
24	565-3-43565654-36-5	ICT for AQA AS level	Julian Mott	Hodder Murray	ICT	£24.99
25	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
26	435-8-56794838-38-3	Truth And Spirituality	Michael Keene	Folens	CPHSE	£13.99
27	968-4-93748377-28-4	ICT for A Level 2nd Edition	Julian Mott	Hodder Murray	ICT	£29.99
28	288-4-03820189-40-3	Pure Mathematics C1	David Rayner	Elmwood Press	Maths	£19.99
29	495-9-38958349-48-3	Pure mathematics C2	David Rayner	Elmwood Press	Maths	£19.99
30	843-8-58437847-83-4	Pure mathematics C4	David Rayner	Elmwood Press	Maths	£19.99
31	484-9-48392384-93-8	Statistics 1	David Rayner	Elmwood Press	Maths	£19.99
32	495-8-39493849-38-9	Statistics 2	David Rayner	Elmwood Press	Maths	£19.99
33	488-9-84058034-83-7	Mechanics 1	David Rayner	Elmwood Press	Maths	£19.99
34	895-8-69485493-84-9	Mechanics 2	David Rayner	Elmwood Press	Maths	£19.99
35	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	Letts	ICT	£3.50
36	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	Letts	ICT	£3.50
37	978-0-00728816-82-5	Apache	Ed Macy	Harper Press	Non-Fiction	£19.99
38	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
39	488-9-84058034-83-7	Mechanics 1	David Rayner	Elmwood Press	Maths	£19.99
40	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
41	837-4-92749568-73-9	A Level Maths	David Rayner	Heinemann	Maths	£35.00
42	968-4-93748377-28-4	ICT for A Level 2nd Edition	Julian Mott	Hodder Murray	ICT	£29.99
43	043-5-46245837-36-5	Health & Social Care for BTEC	Lynda Mason	Heinemann	Health & Social care	£9.99

Queries

Book ID	ISBN Number	Book Name	Author
15	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner
23	837-4-92749568-73-9	A Level Maths	David Rayner
28	288-4-03820189-40-3	Pure Mathematics C1	David Rayner
29	495-9-38958349-48-3	Pure mathematics C2	David Rayner
30	843-8-58437847-83-4	Pure mathematics C4	David Rayner
31	484-9-48392384-93-8	Statistics 1	David Rayner
32	495-8-39493849-38-9	Statistics 2	David Rayner
33	488-9-84058034-83-7	Mechanics 1	David Rayner
34	895-8-69485493-84-9	Mechanics 2	David Rayner
39	488-9-84058034-83-7	Mechanics 1	David Rayner
41	837-4-92749568-73-9	A Level Maths	David Rayner

Publisher	Genre	Price
Elmwood Press	Maths	£9.99
Heinemann	Maths	£35.00
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Elmwood Press	Maths	£19.99
Heinemann	Maths	£35.00

Book ID	ISBN Number	Book Name	Author
13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne
16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks
19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson
35	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne
42	968-4-93748377-28-4	ICT for A Level 2nd Edition	Julian Mott

Publisher	Genre	Price
Letts	ICT	£3.50
Heinemann	ICT	£19.99
Microsoft Press	ICT	£19.95
Letts	ICT	£3.50
Hodder Murray	ICT	£29.99

All books out to a student

18/[illegible]/2010

First Name	Last Name	Book ID	ISBN Number	Book Name	Author	Returned?	Genre	Price
		13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	☐	ICT	£3.50
		16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks	☐	ICT	£19.99

Date of Loan	Date Due Back
13/04/2009	17/06/2009
15/07/2009	06/08/2009

All Overdue books for particular form

18/03/2010

First Name	Last Name	Book ID	Book Name	Author	Publisher	Genre	Price
			17 Apache	Ed Macy	Harper Press	Non-Fiction	£19.99
			16 Applied ICT A2	Maggie Banks	Heinemann	ICT	£19.99
			13 ICT GCSE Success	Sean O'Byrne	Letts	ICT	£3.50

Date Due Back	Form Room	Form Tutor
01/08/2009	TA45	
06/08/2009	TA45	
17/06/2009	TA45	

All Students that have an overdue loan by atleast two weeks

18/03/2010

Student ID	First Name	Last Name	Form ID	Book ID	ISBN Number	Book Name	Author
			15	17	978-0-00728816-82-5	Apache	Ed Macy
		s	16	18	978-0-74879846-95-4	As Business Studies	Steven Foden
			15	16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks
			16	14	043-5-46245837-36-5	Health & Social Care for BTEC	Lynda Mason
			19	15	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner
			17	19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson
			15	13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne

All Students that have an overdue loan by atleast two weeks

18/03/2010

Student ID	First Name	Last Name	Form ID	Book ID	ISBN Number	Book Name	Author
			15	17	978-0-00728816-82-5	Apache	Ed Macy
			16	18	978-0-74879846-95-4	As Business Studies	Steven Foden
			15	16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks
			16	14	043-5-46245837-36-5	Health & Social Care for BTEC	Lynda Mason
			19	15	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner
			17	19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson
			15	13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne

Books of a particular genre due back between 2 dates 18/03/2010

Date Due Back	Book ID	ISBN Number	Book Name	Author	Publisher	Genre
16/07/2009	13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	Letts	ICT
29/07/2009	19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson	Microsoft Press	ICT
06/08/2009	16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks	Heinemann	ICT

Price	First Name	Last Name	Loan ID
£3.50 C			39
£19.95 C			40
£19.99 F			41

Find all books due back on a particular date

18/03/2010

Book ID	Date Due Back	ISBN Number	Book Name	Author	Publisher	Genre
35	17/03/2010	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne	Letts	ICT
15	17/03/2010	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner	Elmwood Press	Maths

Price	Loan ID
£3.50	50
£9.99	52

Find all books due back today

18/03/2010

First Name	Last Name	Form ID	ISBN Number	Book Name	Author	Publisher
		27	843-8-58437847-83-4	Pure mathematics C4	David Rayner	Elmwood Press

Genre	Price	Date Due Back	Loan ID
Maths	£19.99	18/03/2010	53

Find all books of a particular genre by an author

18/03/2010

Book ID	ISBN Number	Book Name	Author	Publisher	Genre	Price
24	565-3-43565654-36-5	ICT for AQA AS level	Julian Mott	Hodder Murray	ICT	£24.99
25	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
27	968-4-93748377-28-4	ICT for A Level 2nd Edition	Julian Mott	Hodder Murray	ICT	£29.99
38	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
40	542-3-45355643-23-4	Spreadsheet Projects in Excel	Julian Mott	Hodder Murray	ICT	£19.99
42	968-4-93748377-28-4	ICT for A Level 2nd Edition	Julian Mott	Hodder Murray	ICT	£29.99

Student ID	First Name	Last Name	Form ID	Form Room
			15	TA45
			17	TB25

Reports

College

All books due back on a particular date

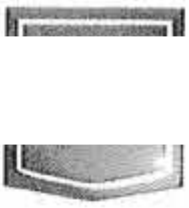
Date Due Back 17/03/2010

Book Name	Author	Publisher	Loan ID
ICT GCSE Success	Sean O'Byrne	Letts	50

Book ID	Price	ISBN Number
35	£3.50	184-0-85602524-59-7

Book Name	Author	Publisher	Loan ID
Pure Mathmatics C3 and C4	David Rayner	Elmwood Press	52

Book ID	Price	ISBN Number
15	£9.99	197-7-45857659-68-7



College

All books due back today

Book Name	Author	Publisher	Loan ID	Date Due Back
Pure mathematics C4	David Rayner	Elmwood Press	53	18/03/2010

First Name	Last Name	Form ID	ISBN Number
[redacted]	[redacted]	27	843-8-58437847-83-4

All books by a particular Author and of a Specific Genre

Author: Julian Mott **Genre:** ICT

Book Name	Book ID	ISBN Number	Publisher
ICT for AQA AS level	24	565-3-43565654-36-5	Hodder Murray
Spreadsheet Projects in Excel	25	542-3-45355643-23-4	Hodder Murray
ICT for A Level 2nd Edition	27	968-4-93748377-28-4	Hodder Murray
Spreadsheet Projects in Excel	38	542-3-45355643-23-4	Hodder Murray
Spreadsheet Projects in Excel	40	542-3-45355643-23-4	Hodder Murray
ICT for A Level 2nd Edition	42	968-4-93748377-28-4	Hodder Murray

College

Students Found By Name Search

Student ID	16
Last Name	
First Name	
Form ID	15
Form Room	

Student ID	21
Last Name	
First Name	
Form ID	17
Form Room	

College

All Students that have an overdue loan by atleast two weeks

Student ID	First Name	Last Name	Form ID
16			15

Book ID	ISBN Number	Book Name	Author
17	978-0-00728816-82-5	Apache	Ed Macy

Student ID	First Name	Last Name	Form ID
20			16

Book ID	ISBN Number	Book Name	Author
18	978-0-74879846-95-4	As Business Studies	Steven Foden

Student ID	First Name	Last Name	Form ID
17			15

Book ID	ISBN Number	Book Name	Author
16	978-0-43546214-76-9	Applied ICT A2	Maggie Banks

Student ID	First Name	Last Name	Form ID
19			16

Book ID	ISBN Number	Book Name	Author
14	043-5-46245837-36-5	Health & Social Care for BTEC	Lynda Mason

Student ID	First Name	Last Name	Form ID
24			19

Book ID	ISBN Number	Book Name	Author
15	197-7-45857659-68-7	Pure Mathmatics C3 and C4	David Rayner

Student ID	First Name	Last Name	Form ID
21			17

Book ID	ISBN Number	Book Name	Author
19	021-1-21556926-41-3	Running Visual Basic	Ross Nelson

Student ID	First Name	Last Name	Form ID
17			15

Book ID	ISBN Number	Book Name	Author
13	184-0-85602524-59-7	ICT GCSE Success	Sean O'Byrne

College

Please find below a list of all students in your form that have overdue books from the College library. It is necessary that these are returned soon otherwise charges will be sent for the price of the book as listed below.

Students should be encouraged to return these books as soon as possible as otherwise they may not be allowed any new loans, unless the price of the book has been paid to the school. If any students have queries about this or have misplaced the book please allow them to come up to the library to bring this to our attention.

Form Tutor: [redacted]

Form Room: **TA45**

Student Name	Book Name	Price	Date Due Back
[redacted]s	Apache	£19.99	01/08/2009
[redacted]	Applied ICT A2	£19.99	06/08/2009
[redacted]	ICT GCSE Success	£3.50	17/06/2009

Total Number of Overdue Books: **3** **Total Price :** **£43.48**

All books of the Genre ICT

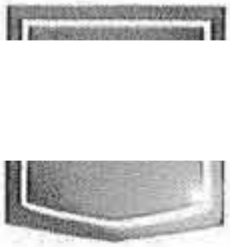
Book ID	13
Book Name	ICT GCSE Success
Author	Sean O'Byrne
Publisher	Letts
Price	£3.50
ISBN Number	184-0-85602524-59-7

Book ID	16
Book Name	Applied ICT A2
Author	Maggie Banks
Publisher	Heinemann
Price	£19.99
ISBN Number	978-0-43546214-76-9

Book ID	19
Book Name	Running Visual Basic
Author	Ross Nelson
Publisher	Microsoft Press
Price	£19.95
ISBN Number	021-1-21556926-41-3

Book ID	35
Book Name	ICT GCSE Success
Author	Sean O'Byrne
Publisher	Letts
Price	£3.50
ISBN Number	184-0-85602524-59-7

Book ID	42
Book Name	ICT for A Level 2nd Edition
Author	Julian Mott
Publisher	Hodder Murray
Price	£29.99
ISBN Number	968-4-93748377-28-4



All books by the author David Rayner

Book ID	15
Book Name	Pure Mathmatics C3 and C4
ISBN Number	197-7-45857659-68-7
Publisher	Elmwood Press
Genre	Maths

Book ID	23
Book Name	A Level Maths
ISBN Number	837-4-92749568-73-9
Publisher	Heinemann
Genre	Maths

Book ID	28
Book Name	Pure Mathematics C1
ISBN Number	288-4-03820189-40-3
Publisher	Elmwood Press
Genre	Maths

Book ID	29
Book Name	Pure mathematics C2
ISBN Number	495-9-38958349-48-3
Publisher	Elmwood Press
Genre	Maths

Book ID	30
Book Name	Pure mathematics C4
ISBN Number	843-8-58437847-83-4
Publisher	Elmwood Press
Genre	Maths

Book ID	31
Book Name	Statistics 1
ISBN Number	484-9-48392384-93-8
Publisher	Elmwood Press
Genre	Maths

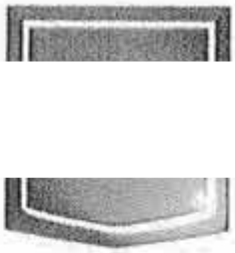
Book ID	32
Book Name	Statistics 2
ISBN Number	495-8-39493849-38-9
Publisher	Elmwood Press
Genre	Maths

Book ID	33
Book Name	Mechanics 1
ISBN Number	488-9-84058034-83-7
Publisher	Elmwood Press
Genre	Maths

Book ID	34
Book Name	Mechanics 2
ISBN Number	895-8-69485493-84-9
Publisher	Elmwood Press
Genre	Maths

Book ID	39
Book Name	Mechanics 1
ISBN Number	488-9-84058034-83-7
Publisher	Elmwood Press
Genre	Maths

Book ID	41
Book Name	A Level Maths
ISBN Number	837-4-92749568-73-9
Publisher	Heinemann
Genre	Maths



Books of a Particular Genre due back between two dates

Start Date: 15/07/2009
End Date: 10/08/2009

For Genre: ICT

Book Name	Author	Publisher	Loan ID	Date Due Back
ICT GCSE Success	Sean O'Byrne	Letts	39	16/07/2009

Price	Book ID	ISBN Number
£3.50	13	184-0-85602524-59-7

Book Name	Author	Publisher	Loan ID	Date Due Back
Running Visual Basic	Ross Nelson	Microsoft Press	40	29/07/2009

Price	Book ID	ISBN Number
£19.95	19	021-1-21556926-41-3

Book Name	Author	Publisher	Loan ID	Date Due Back
Applied ICT A2	Maggie Banks	Heinemann	41	06/08/2009

Price	Book ID	ISBN Number
£19.99	16	978-0-43546214-76-9

College

All Books on loan to

Book ID	Book Name	Author	Returned?	
13	ICT GCSE Success	Sean O'Byrne	☐	ISBN Number
				184-0-85602524-59-7
Genre	**Price**	**Date Due Back**	**Date of Loan**	
ICT	£3.50	17/06/2009	13/04/2009	

Book ID	Book Name	Author	Returned?	
16	Applied ICT A2	Maggie Banks	☐	ISBN Number
				978-0-43546214-76-9
Genre	**Price**	**Date Due Back**	**Date of Loan**	
ICT	£19.99	06/08/2009	15/07/2009	