

**Tackling the
COMPUTER STUDIES
Coursework
Project**

**A guide to successfully completing the coursework element of
the computer studies examination
(CIE 0420 Computer Studies).**

Revised for the Summer 2012 examination session

This project is probably the largest single piece of work you will have to produce. This guide will take you step by step from planning, through design to producing, testing to evaluation.

The important word here is **guide**. This is not a substitute for the real thing, nor will it tell you specifically what to write. All the imagination and creative work has to come from **you**. What this guide will give you is the questions that you need to ask yourself at every stage: it will tell you how to set out your work, what sections to concentrate on, where marks can be gained (and lost), give you the confidence you will need when you begin to wonder and doubt if you are on the right lines or going away from your objectives. Read it well and become thoroughly familiar with it – it might be the only way you will survive and come through with the mark you deserve.

CONTENTS

	Page
1 Introduction	2
2 Project report contents	3
3 Defining the problem	5
4 Analysing/ describing the present system	6
5 Evaluation of the present system	8
6 Alternative solutions	9
7 Planning	10
8 Solution analysis	11
9 Hardware and software	12
10 The solution	13
11 Documentation	15
12 Evaluation & future development	16
13 Other considerations	17
14 Finally	18
15 Appendix 1: Mrs Bouchette's Dog Grooming Salon	
🐾 <i>Situation outline</i>	19
🐾 <i>Problem statement, Outline</i>	20
🐾 <i>Objectives</i>	
🐾 <i>Defining the problem</i>	21
🐾 <i>Evaluation of present system</i>	23
🐾 <i>Alternative solutions</i>	24
🐾 <i>Planning</i>	26
🐾 <i>Describing the system</i>	27
🐾 <i>Hardware & software</i>	30
🐾 <i>Testing</i>	31
🐾 <i>User guide</i>	32
🐾 <i>Technical guide</i>	35
🐾 <i>Evaluation & future developments</i>	38

INTRODUCTION

The project report that you are going to produce for the IGCSE is probably going to be the largest piece of work that you have so far attempted – in any subject. It is vital that you listen to our advice, and follow the timescale that we suggest. If you do not do this you may find yourself in a real mess.

The purpose of the project is to “allow candidates to demonstrate their ability to undertake a complex piece of work, which is the computer-based solution to a significant problem, to complete the solution and present their results” [0420 syllabus pg 5]. The project must contain substantial practical work... and must be thoroughly documente. Your project will be assessed under the following headings:

- Analysis
- Design
- Implementation
- Testing
- Documentation
- System evaluation and development

The project is marked out of 50 and then divided by 2 to give the actual % that will be carried towards your final IGCSE grade. The maximum contribution that the project can make is 25%. There is no reason why you cannot achieve 20+%.

It does not matter whether you choose to work with spreadsheets, databases, a programming language or a combination of 2 of these.... The project will suggest the best software ‘tool’ to develop your solution. Do not think (or worry) that one will be better received than another – it is a working and well documented solution that counts.

9. Outline plan for developing the solution
How do you propose to develop the solution. What stages are going to be necessary. Describe the steps involved in moving from the 'idea', through design, through building the prototype. It is also necessary to indicate the time frame that is going to be required for each stage.
10. Outline of the proposed system
Present the initial ideas: spreadsheet, database... with any necessary descriptions. For example, a map showing how various forms and tables are connected is a good idea for Access solutions.
11. Detailed proposed solution
Spreadsheet with formulas, database record structure, form designs, report layouts.... and accompanying explanation.
12. System requirements
What hardware/ software do you need to run the project. This must not just be a shopping list. You should explain and justify the necessity of the items you are listing and how they fit into the design/ plan you have produced.
13. User guide
Explanations as to what the user will see on the screen, how to get into the spreadsheet/ database, how to enter data, change data, print... Identify a number of tasks the user will wish to perform and write the instructions.
14. Technical guide
Instructions and explanations which will allow another person to make modifications to your solution.
15. Testing plan and results
An outline of the way you have systematically tested that your solution works as you want. It must not crash when unexpected data is presented, nor should it give silly errors when data of the wrong type or size is entered. Selecting command buttons on an Access form should link you to the correct objects.
16. Completed project evaluation
This is a critical look at how well your project has addressed the initial project that you set yourself. Your solution should have addressed the objectives you set for yourself. You will be marked down for not being honest or for giving a fanciful view of your work. Be honest and realistic.

And that is all there is to it!! Don't worry. You will be doing things in stages and the project report will gradually be put together. This is important because it is easy to work with ideas/ designs/ solutions without writing down what you are doing/ thinking – wrong. You need to remember to carefully write-up a stage before moving on to the next. As you go through the various stages make sure that you are keeping the pieces together in a safe place. **It is your responsibility to keep everything - not your teacher's.**

ANALYSING / DESCRIBING THE PRESENT SYSTEM

When you were defining the problem, you outlined the way in which data was collected, saved, processed and why there was a problem to be solved. What is needed now involves two parts: a narrative description of the way the current (manual) system is working and a pictorial description in the form of a set of system flowcharts. The idea is to explain in accurate detail what actually happens now.

To consider:

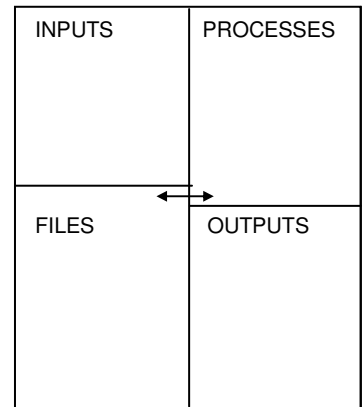
- In what order are any processes carried out
- What data enters the system
- At what points in the system is data entered
- Is the data verified or validated
- What kind of files (master or transaction) are there
- What operations are carried out on the files
- When are reports produced
- What are the purposes of the reports

- and so on

To begin with, try dividing a sheet of A4 paper into 4 sections like this:

This is called an IPFO chart (for obvious reasons).

Try and think of the separate inputs, files... that occur in your system. For example: inputs might be invoices, marks from a test, request for information; files might be tutor register, marking sheet, seating plan... and so on. Take time with this because the more you can think of, the easier will become the task of identifying where exactly any problems occur and what you might do to solve them.



Now comes another tricky task – building a systems flowchart. There are different ways of producing these. Any layout will be acceptable **IF** it correctly tells the full, accurate story of what is going on now. Whichever layout you choose there should be a flowchart for each process you have listed in the IPFO chart.

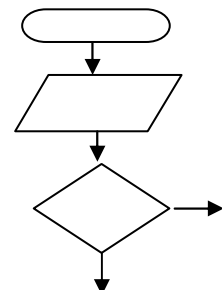
Try this method. Split a sheet of paper into 5 columns:



Put the items you have collected in the four areas under the right headings **but** in the **order** that they actually happen. For example you have to input before validating, you need to have a file to update but does it need to be put in any order first. Do not forget the standard box shapes and to link across the columns correctly. You need a small chart for EACH process.

Alternatively you may wish to stick to the usual flowchart style:

You may have a number of small flowcharts, each representing a separate 'operation' and containing only 3 or 4 'boxes'. Do not worry, this is perfectly alright. Your systems are, after all, relatively simple – they do not involve many users or files. For the most part you might be the only user and an A4 folder might be the only file.



The main point to remember is that you are practising the skills of analysis and presenting the findings in a recognised standard way.

Having described the system diagrammatically, you can now successfully describe the system in narrative form by writing some narrative description to accompany each flowchart. It is impossible to say how many words to write since each process is so different. It might be quite simple (find and update a record) or complex (use a sales figure to update a stock record and adjust a reorder schedule).

On the marking scheme...

- | | | |
|----|---|---|
| 3. | How well, overall, is the existing system described | 2 |
|----|---|---|

EVALUATION OF THE PRESENT SYSTEM

The evaluation of the present system is concerned with your opinion and thoughts about how well the system operates. Do not be completely negative. There will be some good things happening in the present system. Almost certainly the system is quite simple – a reminder that your system should be too. Perhaps the present system has grown beyond what the people involved originally intended, or maybe the volume of data is now too large to be handled effectively. Look for the following:

- data volume too large for the system to cope effectively
- paper files becoming messy – especially if updated manually
- paper transactions getting lost
- searching for data items or records too time consuming
- data cannot be sorted into a useful order very easily – or not at all
- records can only be accessed in one order
- presentation of information is inadequate for what the user wants
- recalculation of values is time consuming and prone to error
- a number of files with similar information exists so files are often out of sequence with each other, much duplicated data

In addition you should also highlight anything that works, that is well liked:

- the simplicity of the system
- easy data entry
- an easily remembered coding system

Look at any ways that this system could be improved. Simple things could be done:

- maybe redesign input forms
- streamline reporting
- take a fresh look at how calculations are made

This is an important element of the report because in here you should be able to identify specific problems that make the system inefficient or impractical. You should end this system with a list indicating these points – good and bad.

Look back at the list of objectives formed when you were defining the problem – these are what you said the system should do or be capable of.

If possible, try and get hold of some documentation that exists in the present system. This makes the task of evaluation easier - there is something to refer to and it is easier for the reader of the report to see what is actually being considered.

On the marking scheme...

4. How well is the existing system evaluated

2

ALTERNATIVE SOLUTIONS

This might seem only a small part of the project but it must be well done. Once you have identified the problems with the present system you can begin to consider other solutions.

You must not decide straight away that there is only one solution worth considering and that this is what you are going to develop. There will be other methods available for developing a solution. I suggest that you should outline at least 3 alternative solutions. For example:

- a better, more organised manual system – better forms design, file design
- retrain the staff or employ someone with specific skills: librarian, sales staff...
- some computerised solution:
 - word processed
 - spreadsheet
 - database
 - specially written program
 - prewritten program...

For each alternative describe how, as a solution, it could fulfil the objectives you have already listed.

Write about 300 – 400 words for each alternative. Once you have done this for each alternative, choose one and justify your choice. This is done very easily if you compare the solutions against your objectives.

On the marking scheme...

5. Description of other possible solutions

2

PLANNING

Putting together an action plan

This section concerns the practicalities of developing your ideal solution into a working prototype or model. You should consider the stages necessary. For example your list might look something like this:

- Identification of data that is going to be collected together.
List the items of data you are going to collect together.
Create a suitable record structure for them.
Creation of database tables and relationships.
Creation of formulas that are required for a datasheet.
- Decide on a suitable human computer interface (HCI)
Design input forms, reports...
- Describe validation rules or checks and build them into the database or spreadsheet.
- Build the database or spreadsheet.
- Design a testing strategy.
- Produce user and technical documentation.

Produce a list such as this which is expanded a little for you to describe the stages. For each step you need to decide on the amount of the time required to complete that step. Remember that some stages may overlap or be linked so you need to work out which steps can be carried out independently of others and which steps will be dependant on others. This schedule gives a working time frame. You are collecting marks for your ability to show clearly how you intend proceeding.

Once you have this list, the various steps need to be put into a formal structure such as a Gantt chart or some other critical path analysis tool such as a PERT chart.

On the marking scheme...

6. How clear is the candidate's overall plan for the solution of the problem? 3

SOLUTION ANALYSIS

Describing the solution

Your solution will contain a number of separate modules which, together, will be the working solution. For example,

- Data entry, which may have different parts:
 - enable new stock to be added
 - enable sales to take place
- Processing of orders, which may involve:
 - update stock file
 - create out of stock report
- Produce reports – a variety depending on who needs information.

It does not matter whether you are using a spreadsheet or a database, this basic outline consisting of separate modules still applies. You must clearly indicate what modules you are going to have, together with a brief description of what each module is actually for.

There are various ways that this section might be put together but you need to ensure that it is very clear how all the various parts connect together. Block diagrams of some kind, data flow diagrams (DFDs), systems flowcharts..... Do not be put off by these phrases/ideas, you should have plenty of examples from your teacher, and they all do the same job – illustrate how the various modules (blocks) are connected (linked). Some will show how data flows and is stored, some might show who does what, with what data and when.

If one word is important here, it is **structure**. Somehow you have to let the reader know how the different parts fit together to make the solution work. There is a technique called top down analysis that you should be familiar with. Starting with an overall view, keep on breaking down processes until you can describe a whole process in detail. For example, you might start off by talking about stock control. This has 2 distinct processes: selling and purchasing. Selling involves deleting stock, creating an invoice, sorting out reordering. Purchasing involves increasing stock, dealing with invoices or purchase orders, updating the reorder lists.....

Some of these processes or tasks might need to be done in a particular order so you need to think about sequence as well.

Take each of the modules that you have described in outline and construct an algorithm for each one. **Do not** be put off by the word *algorithm*. What is required, for each stage/module, is a flowchart/diagram which shows clearly the steps necessary to make the module work. It might require 10 steps (updating a stock file and checking the stock level has not dropped too low), it might require only 3 or 4 (choose a report from a menu and print it). However, each module **must** be described or represented in this way.

Added to this comes the question of whether the stages that you are describing are actually appropriate to the solution that you are going to develop. Does the system need the stages that you suggest in order to make it work in way you want? Each module must be a fundamental part of the system, not just something that you have decided to include in order to smarten it up, or make it appear more 'sophisticated'. You must show that your analysis requires these modules, and that they are necessary components of the completed system.

On the marking scheme...

- | | | |
|----|--|---|
| 7. | How well is each stage represented by charting methods | 2 |
| 8. | How well is the method of solution described | 3 |

HARDWARE AND SOFTWARE

What hardware is needed to make your solution work?

The biggest temptation here is to just produce a shopping list of items. This is **not** what is needed. Yes you must list the requirements but you must also justify the items you suggest. This means that you need to **explain** – with 2 reasons for each - why the piece of hardware is needed. This is quite straightforward. Keep it simple.

For example:

Mouse (or trackerball):	To enable successful use of the system, pointing to data items on the screen, selecting options on screen... a necessary part of the graphical user interface.
Printer:	A colour printer may not be necessary, especially if you are just printing stock reports, lists or orders. Explain the difference between laser and inkjet printers (cost, benefits of colour at a later stage...)
Monitor:	What size? This could be an interesting point to debate. A nice 19" screen? Touch sensitive?
Backups:	How will you backup the data/records? Archive to CD, use a tape streamer?
Scanner:	Need to input documents or images?
Camera:	Digital photographs might be needed to enhance database records (car sales, personnel records...)
Bar code reader:	Inputting data quickly and accurately from pre-printed codes on products?

You get the idea? List all the components and give good reasons why they are needed. The list might be quite long, it may only be standard hardware, but it must contain every device that your proposed system requires in order to function as you have specified. Although not required it often a useful idea to give an indication of cost as well.

On the marking scheme...

9. A complete and justified list of hardware requirements? 3

What software is needed to make your solution work?

This section is easier to write since there will only be a couple of products to mention. However, it is important that you explain exactly why you have chosen them, indicating what they are need for and how they will be used. If you are writing your own software in LOGO, Visual Basic, HTML, php... then it is vitally important to highlight why you are choosing to write this code with a reference to what you have mentioned in section in **Alternative Solutions**.

On the marking scheme...

10. A complete and justified list of software requirements? 3

THE SOLUTION

You might feel that the solution, the finished 'product' is the most important part of the project and that this is where you will be awarded the most significant number of marks. In fact 25 of the 50 marks (implementation, testing, documentation, system evaluation and development) are given to the actual production of the solution – the working spreadsheet, database, web pages..... As you have seen from everything written above, the whole idea of the coursework is to test your abilities of investigation, analysis and design and so the marking scheme reflects this.

So, how is your solution marked?

Does the solution work? How well has it been constructed?

The most important concerns are to do with how you have converted your solution into a working interconnected system.

Does the solution work? If you have described the algorithms in the plan accurately this should not be a problem. Don't forget, the idea of planning is to help you put together the final product correctly so there should not be anything happening in the wrong order, or affecting the wrong items of data.....

When you have constructed the sheet or form, have you used the most appropriate method? The wizards will help you with many tasks – use them, that is what they are there for. Do not try and produce your own Visual Basic version of coding behind a button if a wizard will build it for you. Firstly you might not be successful with your attempt at VB, secondly, we are looking at how well you can use the tools within the software to develop the solution. In a spreadsheet, for example, have you created a macro that could have been found through the toolbars, or have you written a long formula that actually already exists as a function?

Have you provided enough information throughout the solution to ensure that it is fully 'readable' as a solution? In other words, is there sufficient annotation to allow someone to follow the solution, even though they might not be fully conversant with the language/code of the solution itself. For example, are there comments in your style sheet or within the procedures of a VBasic module?

Testing the solution

This is vital. You have to be sure that the solution you have created works as you wanted it to. You have to ensure that the objectives you set out to achieve have been fulfilled. The only way to do this is to test the solution thoroughly. Of course you will need to show that the algorithms link up together: that one form leads on to another; that values are carried across spreadsheets in a workbook; or appropriate actions are taken in response to a 'submit' button on a web page for example, but this section deals more with the actual data that you type in. The questions that are asked of you concern the way in which your solution deals with data that might not be expected as well as data that it does expect. You cannot test your solution with just data that is reasonable – it has to be idiot proof!! How you handle this part is worth 14% of the entire project mark.

There are two parts to this section: a testing plan; the production of test data.

A testing plan is a methodical and logical walk through the solution testing its processing of data. The way in which the plan is presented is up to you. A table listing all the tests you plan to carry out with a reason is acceptable. List the test, together with why you are doing the test, and how the test helps to show a specific objective from your list (**Defining The Problem, point 5**). For example you may have a calculation to update stock levels so you would state that you are going to test product code and quantity entry to test the calculation and creation of a new stock level, in line with the objective to improve the accuracy in maintaining stock levels.

Alongside the plan, and usually integrated with it, are your test results showing the data item you are entering, what you expect to happen, and what did happen.

There are three types of test data: standard, extreme and abnormal.

Standard data: data that you expect in normal working.

Extreme data: this is standard data, but has values on the limits or extremes of a range. For example you may have a range check for validation purposes - does the database or spreadsheet react correctly to values at the extremes of this range check? You are testing the accuracy of your 'formulas'. A problem often occurs when you have fields or cells that contain zero and you then use that field or cell in a calculation. You must demonstrate that your solution can cope with values that are acceptable but at the limits of those acceptable values. For example, if you are looking at the scores (as percentages) that students achieved in a test the limits will be 0 and 100, but you must show that both 0 and 100 will be accepted and processed properly.

Abnormal data: Data which is not of the correct type. For example what happens when a word is typed into a numeric field. Equally, a value that falls outside a range must be rejected. Looking again at the student test scores a value less than 0 (a negative number) or greater than 100 must be rejected. This type of data should be rejected with some sort of error or 'help' type message.

This can seem like a very difficult and boring part of the report. For each type of test you can produce some screen shots which can show clearly what happens – evidence of your claims! Some testing can be demonstrated within your user guide perhaps. However, you do not have to show screen shots of every single test for every field or cell that accepts data. You are demonstrating that you understand the necessity of having a checking mechanism to ensure that the data that the program uses is reasonable, sensible and acceptable – which is, after all, the meaning of the word 'validation'.

On the marking scheme...

Implementation

- | | | |
|-----|--|---|
| 11. | Is the solution clearly explained in relation to the requirements of the system? | 2 |
| 12. | How accurate and successful is the solution? | 3 |
| 13. | Is the solution fully coded/annotated | 3 |

Testing

- | | | |
|-----|---|---|
| 14. | How complete is the test strategy? | 3 |
| 15. | How good is the testing of normal, abnormal and extreme data? | 4 |

DOCUMENTATION

Technical Documentation

This is all about how easily the solution could be modified - especially by someone else. If you give this solution to someone else do you also give them enough information to enable them to successfully change or develop your solution?

In this section you need to present the 'internal workings' of your solution as a report complete with a contents page, separate modules clearly documented.... If you have produced a spreadsheet, for example, you will need to clearly present the formulas that you have created with an explanation of what they do and what the various cell references might mean. Any macro that you have produced will also need to be presented and explained. In a database you will need to print out and explain the coding behind command buttons if you have added to the code that the wizard generates. You will need to present and explain the table design which shows, for each field, any formatting, default value or validation that is applied. If you have defined any combo boxes the presented list and its associated field need explaining. Think about all the different elements of your solution. What would someone need to know about these elements if they set out to try and make any changes to the solution?

On the marking scheme...

16. How useful is the technical documentation for making changes or developing the solution

2

User documentation

When you have completed the solution and are happy that it works you need to produce some instructions explaining how to use it. If you buy a new watch or a DVD player you expect some instructions: an operation manual, an instruction booklet. Whenever you buy some new software you need something to let you know how to make it work for you. Often there are 'help' features, but you are not going to build this into your solution. You are going to write an instruction booklet. This is called a 'User Guide'. This will be quite a long part of the project report, but will contain screen shots as well as text. You must assume that the user has just switched on their computer and is staring at the desktop. What do they do?

Guide them through your solution. Break it into small sections. If you have produced a database for a Rental DVD Shop then you will need sections on: adding a new DVD, borrowing a DVD, adding a new member, returning a DVD... and so on. As you produce this guide include some examples of what might be required to be typed in. If you have a section explaining how to add a new product to your shop's stock list, why not produce the user guide in such a way that it demonstrates exactly what will be seen on the screen, what data needs to be entered.... These are called *sample runs*. The best way to test the success of the User Guide is to ask a friend who knows nothing about your solution to sit down and try and use it. They should follow the instructions exactly as they are written. See what happens!

On the marking scheme...

17. How well could a novice use the solution by following the instructions provided?

3

EVALUATION & FUTURE DEVELOPMENT

This is concerned with the solution you have now completed. You have already produced an evaluation of the original system. What you discovered there has led you to the solution you have created. There is a very simple way of completing this section. Think back to that first evaluation (look back at ***Analysing/ Describing the Present System, Evaluation of Present System***). You compared the working of the original system with a list of objectives. That list of objectives should also apply to this system. The objectives themselves should not have changed. Produce an evaluation in the same way that you did for the original system. It should, of course, be much more positive this time. In this section you get marks for being honest. If the system does not work as well as you had hoped, then say so. The evaluation is a description of how well the solution works. The design might be great but your skill level may have stopped you from getting things working properly. That is fine, but say so. Do not make claims for your solution that are simply not true. The person marking the project will be able to see how well your solution performs.

Of course it is also true that even though you have produced the solution there are things that could be done to improve it: the forms for data entry might benefit from being redesigned; some validation routines might be added; some additional reports might be useful. You must present some ideas about how the system could be improved in the future. These do not have to be detailed or technical. They should convey to the reader the impression that you have considered the fact that some alterations might be beneficial, with a reasonable idea of how they might be achieved.

On the marking scheme...

- | | | |
|-----|---|---|
| 18. | How sound is the candidates evaluation of the solution? | 3 |
| 19. | How well are opportunities for development considered? | 2 |

OTHER CONSIDERATIONS

... a final word of warning on behalf of your teachers.

There is a great temptation to seek help from other people – either friends within school or within your family. It would be unrealistic to suppose that you would reject help. Indeed we will gladly give you all the help and encouragement that you need – but we will not do things for you. Equally you must realise that the work that you are presenting must be your own work. It is sometimes the case that we see work that we believe is not entirely that of a particular student. It is quite easy to identify because we know your capabilities and the speed at which you work. We do keep a very close eye on what you are producing. If we have any doubts about your work at any stage we will question you about it. We must satisfy ourselves that what we are marking is the product of your efforts. When you finally submit the project to us we must be confident that the mark awarded reflects accurately your abilities and the amount of effort that you have put into it.

It might also be the case that you have produced a fantastic project, but the problem that you set yourself in the first place was not that difficult. In producing the solution you have not really tested yourself or extended your skills.

And so, **on the marking scheme** there is:

20. Teachers may reduce the total mark awarded in cases where too easy a problem leads to an overvaluation, or cases where candidates have received help beyond a reasonable and acceptable level.

A reduction of the total can be made up to a maximum of 10 marks.

Finally.....

If you have made it this far, congratulations. As stated at the beginning, this project may well be the most significant piece of work you have ever had to complete. It will have taken a considerable amount of time and effort. You may well be exhausted and fed up of it! If you have taken advice, produced and kept to a realistic time frame then you should have made it to the deadline your teacher has set. Having handed in the project report you should also give yourself a pat on the back. There should be every reason to believe that you have achieved a mark of 40+ out of the 50 available. So, as you prepare to sit down for the written exam that means you already have 20+% in the bag – a great foundation on which to build a commendable overall grade.

If you have any comments, suggestions..... please contact me: colin@ictgcse.net

I would also welcome copies of project reports that you have produced so that, with your permission, I can add and expand the examples used in this project guide.

I wish you every success with your project.



Colin Stobart

Who am I?

I have worked as a teacher of Computer Studies/ICT (and some maths) since 1985. In 1987 I became Head of Computing at The English International College, Marbella, in the south of Spain before leaving for The British School of Kuwait in 1999 where I was Head of ICT and later Publications/Web Manager. In 2005 a return to Spain beckoned with Head of ICT and Assistant Head posts. I returned to the UK in 2008 and am currently teacher/housemaster at a boarding school in Dorset. Since 1985 I have taught and moderated the CIE syllabus 0420 (7010) and 0417, alongside AS and A2 with Edexcel. In Kuwait I also acted as the moderator for 0420 at the British Council. I am an Assistant Examiner at CIE on 0417/3 and a QPEC member of 0417 papers 2 and 3.



This guide is based on my experience of looking over, marking and moderating literally 100s of coursework projects. It is my own personal advice and is written in the light of the current syllabus requirements and my judgement of what is needed in order to gain high marks. If you have any constructive comments please contact me at: colin@ictgcse.net

APPENDIX 1



Mrs Bouchette's Dog Grooming Salon

Situation outline

Mrs Bouchette has a part-time dog grooming business. Dogs come to her for a hair cut, washing, nail clipping and so on.

She keeps all the information about dogs that come in for clipping or other treatment, such as ears cleaning, nails clipping... held on index cards. These details include the dates when the dog has been clipped together with information about rabies injections, special considerations or requests. For example a lot of dogs have very sensitive paws. These dogs need careful handling because hair grows under the paw between the foot pads. This hair needs cutting out because grass seeds and other bits and pieces can get caught and easily lead to infections. The index card also contains name and contact details of the owner.

Additionally she needs to keep a 'history' of the treatment she has given a dog because of checks for ticks, ear infections and so on.

Her appointments diary serves as her bookings record. Although Mrs Bouchette only regularly deals with about 50 dogs there is a lot of information building up about each dog.

She is considering making this business her full-time occupation. She realises that the information that she keeps is vitally important and already a lot to handle. Should the business be successful it is obvious that she will need to consider changing or improving her record keeping system so that all information is easily accessed and well structured.

Problem Statement

Mrs Bouchette has a paper-based record keeping system. At present, she stores data about dogs that she clips on individual index cards. Bookings for an appointment with a dog are kept in her appointments diary. Mrs Bouchette realises that the historical record of treatments and visits that each dog has made is gradually becoming quite extensive. While the system does work reasonably satisfactorily Mrs Bouchette has decided that she wants a system where record keeping is more systematic, easily searched and structured. This project is going to investigate ways in which the data can be more effectively stored.

Problem Outline

Mrs Bouchette would like to have a record keeping system where information is retrieved easily, and where the data kept about each dog is more structured. At present, although she deals on a regular basis with only about 50 dogs this is set to grow considerably when she takes the business full-time. The information about each dog is on one index card stored in dog's name order. Many owners have more than one dog so her index cards are cross referenced and contain duplicate information. She does not have a separate index of owners. Mrs Bouchette keeps a lot of information about each dog such as contact information, rabies injections and so on, but the information she keeps about a dog's treatments and visits will grow rapidly. For this reason the volume of data is going to outgrow her present index card file. Additionally her appointments diary is going to become more extensively used and although the diary still provides a good booking method, it is not easily searched.

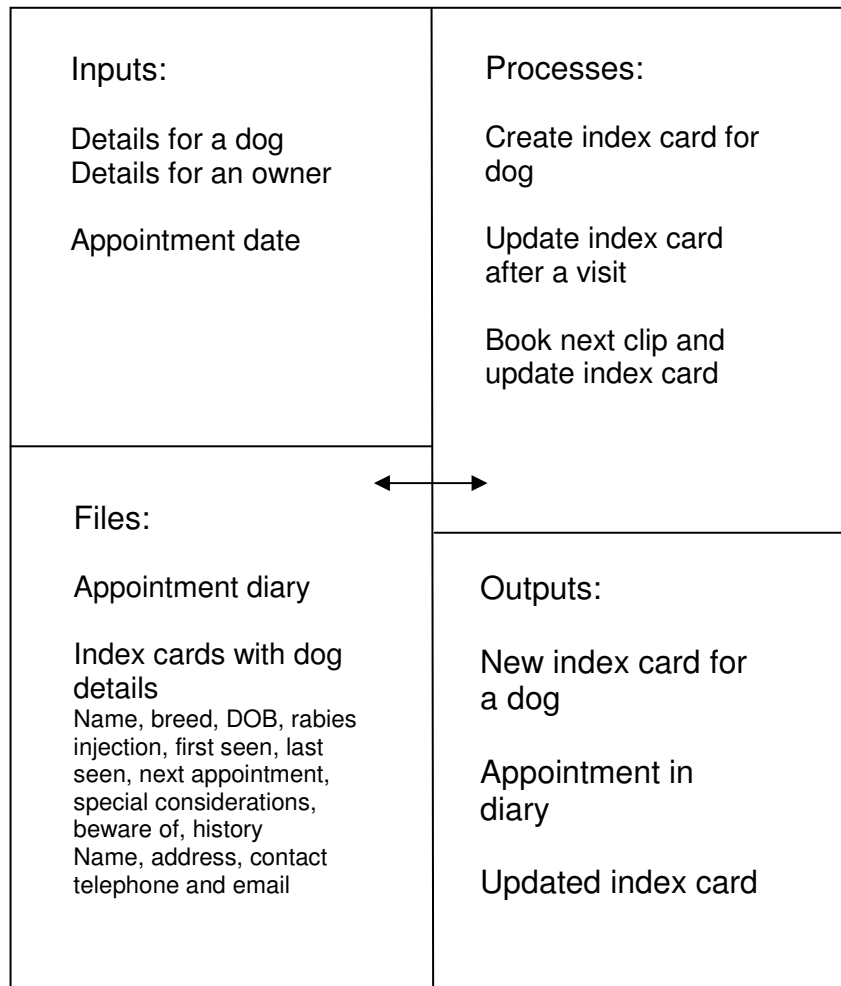
Full and accurate information is required about each dog which combines, or links, owner and dog details. This is needed because if an owner has more than one dog they will normally bring them for grooming on the same day. Having an automatic report produced of dogs due in the coming week, or those who have missed an appointment would be a great help to Mrs Bouchette. Usually dogs are seen twice a year so future appointments could easily be automatically generated. She is keen that the system is easy to use since she does not want to spend too much time entering information or defining search criteria. She really needs the details for a dog (their appointments and history) to be more accessible, accurate and easily searched.

Objectives

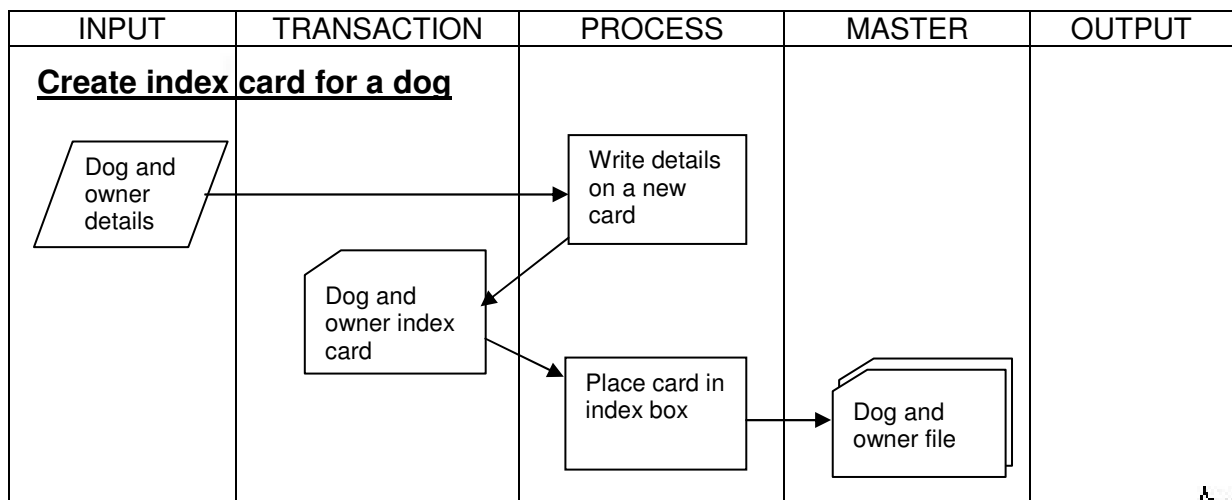
The system should:

- Allow for a complete and accurate set of data be held for each dog*
- Enable quick and easy search to dogs and their history*
- Enable a quick search for an owner and all their dogs*
- Ensure that the details for a dog are both easily read and recorded*
- Produce reports for dogs due for an appointment and those with missed appointments*
- Automatically generate future appointments*

Defining the problem

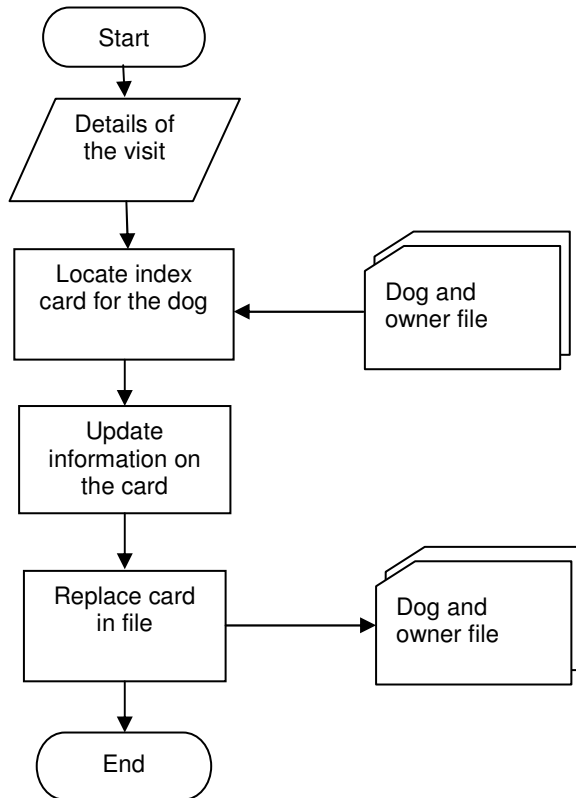


Example system flowchart:



Example flowchart

Update index card after a visit



Once a visit has been completed Mrs Bouchette has to update the index card for the dog.

She has to locate the card in the index file. These cards are in the order of dog's name.

After pulling the correct card out of the file the new details are added.

The card with the updated information added is then replaced into the card index file, ensuring the index order is maintained.

Evaluation of the present system

The system at the moment is quite simple. There is not a great deal of data but a problem is going to arise as the system is expected to grow considerably.

The data kept for each dog is straightforward and complete, but over time will expand to make the need for more than one card per dog necessary. The history element will become quite a list over time. At present the card index is stored by the name of the dog and although just now no two dogs have the same name this is unlikely to always be the case. With the possibility that one dog's details will spread across more than one card, making sure that the follow-on cards are referring to the right dog might become a problem.

There is duplication of data. Since an owner might have more than one dog, that owner's details will be stored on each of their dog's index cards. Apart from the problem of data duplication it also means that making sure that the details are consistent is important. Although this at the moment does not present a problem, it does not mean it is something that should be passed as a potential problem, given the rate at which people tend to change mobile phone numbers and email addresses.

The appointments diary is searched and updated manually, with new appointments added in response to requests, either by telephone or, as an owner is leaving, they may book another appointment for 6 months ahead. This is a simple system but does not allow any searching, for example by dog or owner name. At the moment Mrs Bouchette does not keep a record of dogs that have missed an appointment although the owner will have been contacted at the time. This is something that Mrs Bouchette realises could easily be solved but would involve creating a new list of some sort.

This is a simple system that has been developed as Mrs Bouchette has gone along. Its simplicity is something that would allow anybody to use it and this must be a feature of any new system. Its main drawbacks are:

- the expected increase in data volume which cannot adequately be saved on an index card*
- the appointments book being a bound volume which does not allow any searching to take place*
- the index cards becoming messy with manual updating*
- a dog's visit history will expand beyond the capacity of the index cards*

There are a number of ways that the system might be adjusted to resolve some of these points:

- redesign the format for keeping dogs' details and record on A4 paper*
- use a loose leaf binder for an appointments diary*

However, these suggestions are just minor and perhaps temporary developments and Mrs Bouchette is looking for something which, while perhaps not radical:

- does allow the data expansion expected*
- remains as simple to use as possible*
- allows data to be searched and updated quickly*

Alternative solutions

Some suggestions:

Reorganise the manual system

It should be possible to redesign the way in which data is stored. At the moment there are two collections of data: index cards containing the details of dogs; the appointments diary.

The index cards could be replaced with an A4 sheet that is carefully designed to contain all the required information. These sheets would be one per dog and be held in a lever arch folder, which can hold 100s of sheets. Larger sheets would have room for plenty of information, including new items such as passport numbers, microchip details, vet contact.... These sheets will need storing in some order – preferably a reference number so that duplicate names do not cause a problem.

...at this point put in an example of how this might look

The diary should still remain but there should be a similar level arch folder holding details of owners. These sheets can contain all the necessary contact and personal details of the owner. This file should be stored in an order: surname and first name may suffice.

...again include a possible layout of this sheet

There will need to be a separate sheet prepared which provides an index and cross reference to dogs and their owners.

As an owner calls to arrange an appointment, these 2 files, along with the diary, if kept close to the telephone, will easily provide enough information to answer any query.

Advantages of this system:

- Minimum redesign – two new forms that are cross referenced*
- Simply used/learned*
- Low cost*
- Time element for data transfer short*

Disadvantages of this system:

- Information can only be stored in one order*
- Cannot generate reports automatically*

Build a computer database using Microsoft Access

A database could be built which stores all the required data. The data that the computer stores would be the same as outlined in the other alternative solutions. The difference lays in the way in which the data can be searched and analysed.

Data will be stored in a number of tables: one for the details of dogs; one for owner details. Information can be entered via a form which will update the relevant table.

...include here a sketch of the form layout. It must not be a screen shot of your finished project because you are still looking at what could be possible and is an obvious give-away to the report reviewer that you are basically working backwards – providing a solution and then producing the history of the story/development afterwards.

Once the data is stored in the database the true benefits of the system will be clear. Information will be able to be accessed very quickly because there will be various search facilities available. If Mrs Bouchette needs to find the details of individual dogs then she could search for a dog – either by name or owner. She could see multiple dog details for one owner. The reporting facility would allow her to automatically see which dogs are due to be seen in the coming week, or which have missed appointments and whose owners need contacting.

...include some outline designs of reports or queries that could be generated, indicating their use and link to objectives.

The main drawbacks are the fact that this system needs to be designed and built from scratch which may take time. It will however be created to exactly Mrs Bouchette's wishes. She already owns a computer but makes very little use of it – for letter writing and producing a few posters. She does not have any experience of using a database program but the interface can be designed to be 'idiot proof' and make using it very simple.


Advantages of this system

- Quick and easy data access*
- Quick and accurate analysis*
- Easily customised and automatically generated reports*

Disadvantages of this system

- Will take time to design and make ready – including data entry*
- Cost of buying database software*

Planning – list of tasks needed to be completed

Task	Estimated days needed	Total days of project
Talk to Mrs B. to get an idea of her requirements, collect samples of cards... <i>Need to make sure that I am aware of the expansion that she is predicting, not just the type and quantity of data now. Get a copy of an index card and a typical page from her appointments diary. Explore the type of system she has in mind. What sort of layout would be most helpful to her when it comes to printouts (order of listing). Maybe draft some sketches with her of screen input forms...</i>	1	
Create a list of all the data required for the new system – check with Mrs B. and ensure that this data can be used to produce output requirements. Knowing what data is collected/stored at the moment, what extra needs to be collected in future to enable the system to function as predicted. Ensure that as much data is collected now because collecting extra pieces for each dog or owner later will be a harder task.	2	3
Design record structures for required tables and think of the relationships needed to produce requirements. Consider record keys needed, relationship type between tables, data types for fields needed, any masks for data, calculated fields, validation routines...	2	5
 <p>All other steps considered like this: designing HCI, data entry, testing....</p>		
Write documentation Create user and technical documentation. Get some feedback from a 'dummy' user to test the effectiveness and accuracy of the user guide.	5	34
Test with Mrs B. Sit down with Mrs B. and run through the system with her demonstrating functionality. Get honest evaluation from her with suggestions of modifications and/or future developments.	1	35
Modifications? Action on evaluation needed?	5	40
Total time about 8 school weeks		

Note:

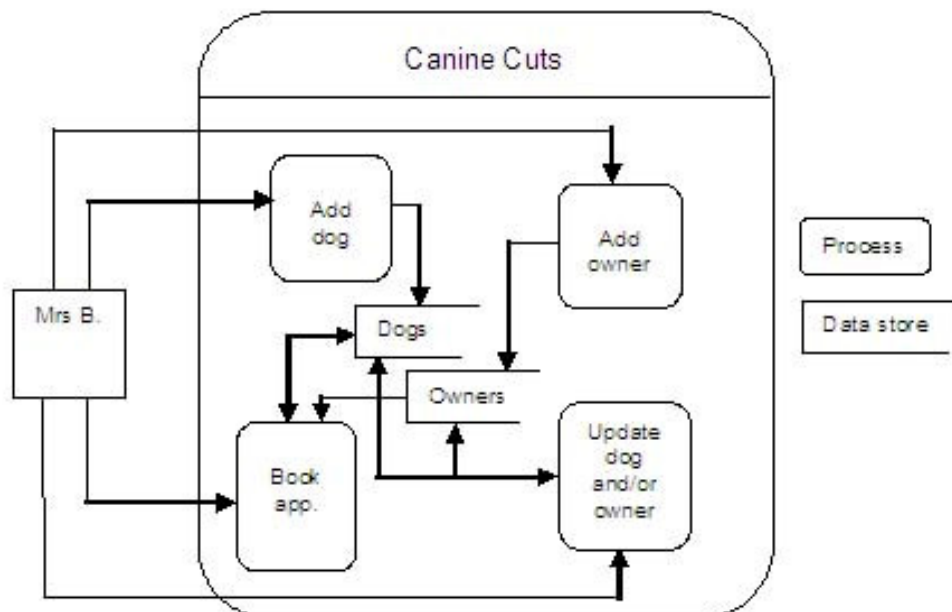
This list can then be converted into a Gantt chart or some other CPA chart. There are plenty of good websites with examples. The best way to create a Gantt chart is using a spreadsheet package.

Describing the solution

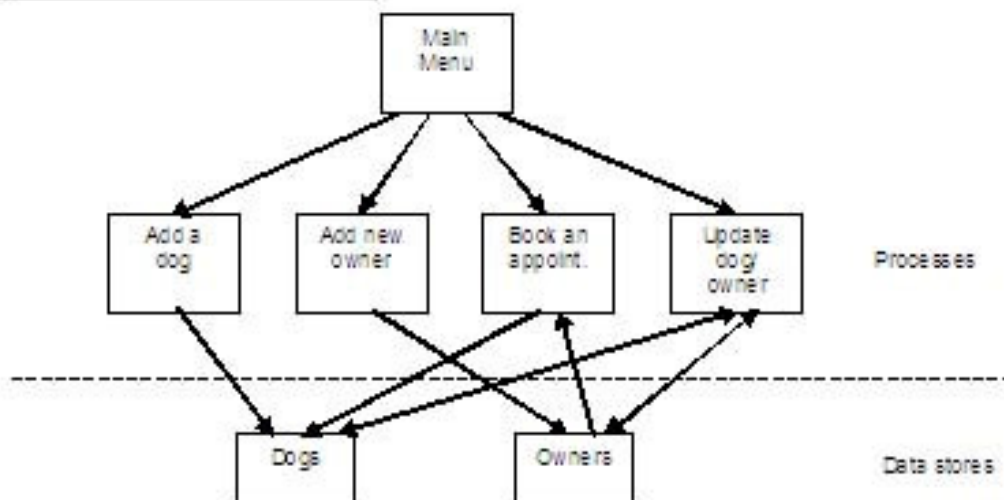
You will need a number of diagrams – of varying detail. The type of diagram you choose to use depends on what you have been taught, are comfortable with, which you think more easily describes the process/ subsystem....

Examples:

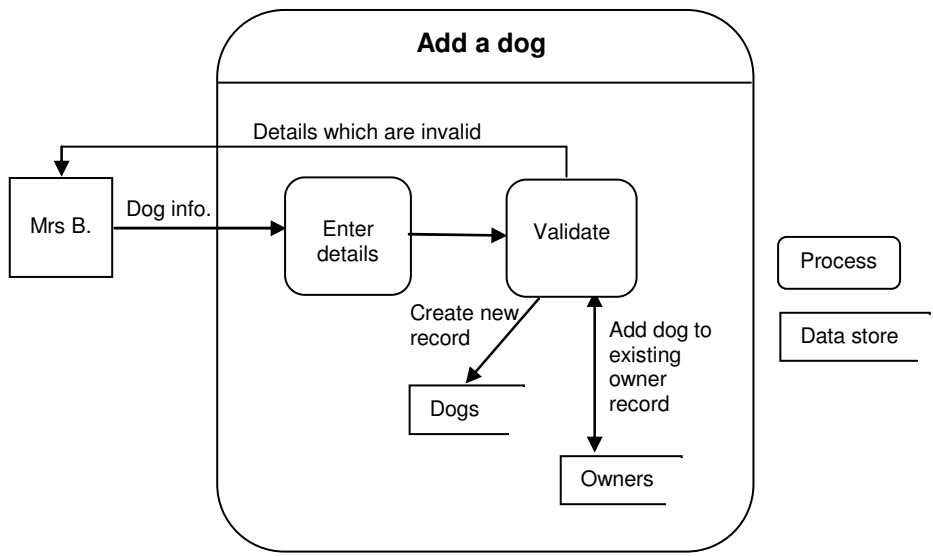
Data Flow Diagram



Structure/block diagram



Data Flow Diagram



Form Layout: Add a dog

Dog ID, name, DoB

Owner ID, name

Clip dates: first, last, next Overdue?

Special details, important info

Treatment history

Text entry for:
 Dog ID, name...
 Special details...
 Treatment history

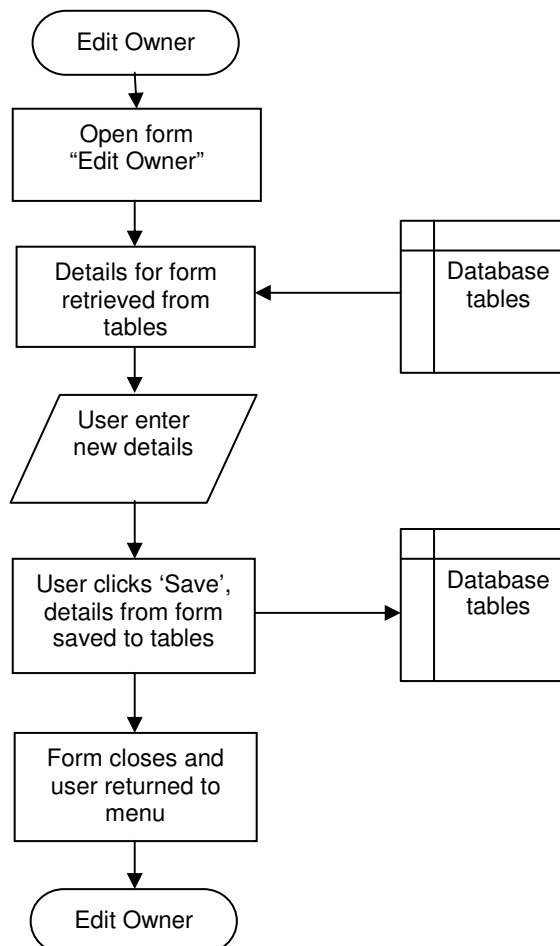
Owner ID to be text entry but also have a button taking you to full details for that owner.

Clip dates first and last clips entered normally but next clip date to be automatically calculated. Overdue calculated.

Pseudocode for recording details of an appointment

- Form 'Dog Details' is opened when 'Dog Details' is selected from main menu.
- Form opens showing all details from the pets table
- Enter new details to end of 'History' text box
- Enter today's date into 'Last clip'
- Date for 'Next clip' is automatically calculated - which is today's date plus 6 months.... : LastClip+182
- Form is closed by clicking button to return to menu.

Flowchart for updating an Owner's details



Hardware & Software

Some suggestions...

Mrs Bouchette already has a laptop computer system, of 4 years old, which comprises:

Hardware

- 25Gb hard drive
- 512Mb RAM
- DVD writeable drive
- HP Inkjet printer capable of 18ppm (b&w) and 12ppm (colour), 2 years old
- USB mouse

Software

- Microsoft Windows XP
- Office 2003 (Standard Edition): Word, Excel, Powerpoint, Outlook

.....

It does not make any sense to change the laptop. The storage capacity is perfectly adequate as is the processor speed.

The laptop has a touch pad mouse but the USB mouse makes moving the on-screen pointer, and selecting items on screen much easier.

Mrs Bouchette had, in early discussions, mentioned a scanner. This is not a present system requirement. There is no need for input of drawings, photographs, or documents of any kind. There may be, in the future, a need for inputting photographs of dogs she looks after. This could be a future development but was not something that Mrs Bouchette asked to be considered at this stage.

The printer is not old and again it does not make sense to change it. The new system will not require a large volume of print, nor will it need great quality. Colour printing isn't an issue but there might be possible future developments where this might be handy – pictures of dogs Mrs Bouchette clips, publicity material or invoices that she issues.

.....

Microsoft XP is a stable and familiar Operating System. Since it comes bundled into the laptop there is no point considering an upgrade to the new Vista OS. Mrs Bouchette is happy with XP and is used to how it works. Office 2003 will work fine with Vista but since there is no point changing the OS just for the sake of it (and spending £100+), we will recommend sticking with the XP OS already installed.

There will be a need to upgrade the version of Office because Microsoft Access will be needed. Other database software could be used (Filemaker Pro – about £250), but it would be sensible to use Access because future developments could require data transfer between Office components and also because it is a popular piece of software – meaning that later changes and development could be carried out easily. It will be necessary to purchase Office Professional (brings in Access and Publisher). Cost would be about £180-200. Addition of Publisher could be useful in the future if publicity material is considered.

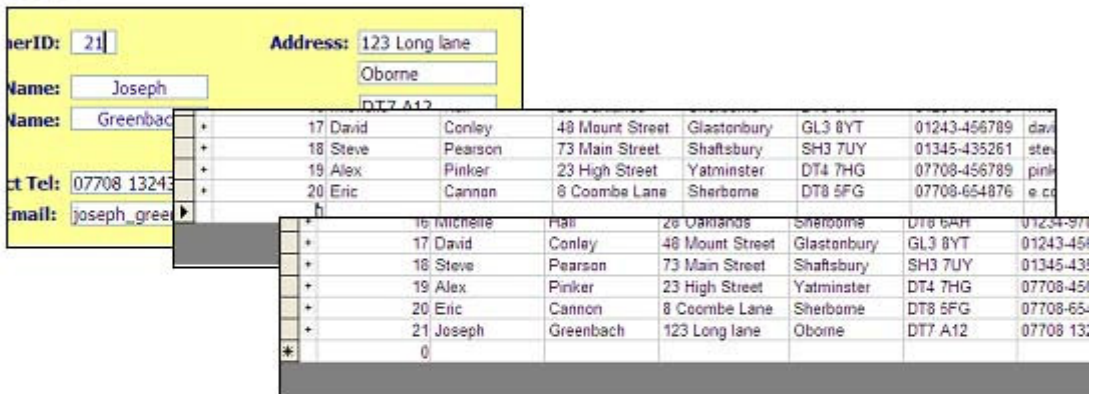
Testing

Test#	Data Item(s)	Value	Data is..	Testing for	Expected	Actual
frmNewOwner						
1	Owner number	12	Abnormal	Owner already exists	Error message	Error message
2	Valid data for record		Normal	Addition to table "tblOwners"	Saved	Saved
frmNewDog						
3	Owner number	30	Abnormal	Owner does not exist	Error message	Error message
4	Dog number	5	Abnormal	Dog already exists	Error message	Error message

Test 1



Test 2



Test 3



Test 4



User Guide

Introduction

Welcome to the user guide for "Canine Cuts". This user guide is going to show you how to use all the various aspects of the program. It will take you step-by-step through the adding and editing of data for dogs and their owners, it will show you what reporting facilities there are.

First of all when you open the database on the computer you will see this menu screen:



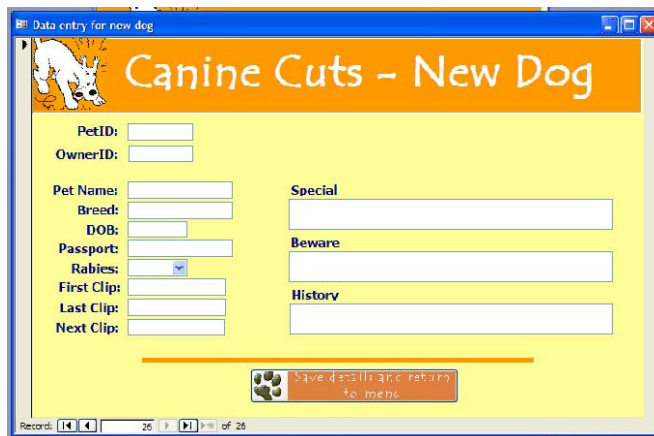
...from this menu select the option you want by clicking on the paw button.

To leave the database and return to your desktop click on the button "Leave Canine Cuts".

Adding a new owner

You need to add a new owner before you can assign a dog to them.

Select "Add a new owner's details" from the main menu and you will see this form.



The screenshot shows a web browser window titled "Data entry for new dog" with a yellow background. The main heading is "Canine Cuts - New Dog" with a small dog icon. The form contains the following fields:

- PetID:
- OwnerID:
- Pet Name:
- Breed:
- DOB:
- Passport:
- Rabies:
- First Clip:
- Last Clip:
- Next Clip:
- Special:
- Beware:
- History:

At the bottom center is a button labeled "Save details and return to menu". The status bar at the bottom shows "Records: 26 of 26".

As you can see there is a lot of information that you need to have to hand before you can complete this form.

You should know the Owner ID as they will already be entered into the database. If you try to assign a dog to a non-existing owner, an error message will be displayed when you click on the "Save..." button.

Notes on data to be entered.

Pet ID: This should be a new unique number. Hint - look at the record selector at the bottom of the form. In this case you could use an ID of 26.

Dates....DOB (Date of birth), First Clip, Last Clip: in format dd/mm/yyyy

Pet name, Breed, Passport, Special, Beware, History are all free text format.

Rabies: this is to indicate if the dog has a current rabies vaccination, choice is simply "Yes" or "No"

Next clip is automatically filled in when you put a date in "Last Clip". Next clip is just 6 months from the date entered as last clip – which will normally be the date that you have clipped the dog.

"Special", "Beware", "History" are fields where you describe what has been done, what you have to watch out for....

For example in "Special" you might include information about what the owner requests – "Teddy Bear" cut for a Yorkshire Terrier, has shampoo allergy use owner's own, do not strip the tail....

In "Beware" include such things as the fact that the dog might have sensitive paws so be careful clipping hair between pads, difficulty last time washing....

In History place the date and the treatment that has been carried out with information perhaps to carry forward to next time, such as whether ticks have been removed so look for them again next time....

You want to know who should have been clipped but haven't been.

Select "Overdue" from the main menu and you will see this:

Dogs who are overdue a clip
These dogs should have been clipped by today: 03/03/2009

Family Name	First Name	Contact Tel	Contact Email	Pet ID	Pet Name	Breed	Last Clip	Clip Was Due
Finler	Alan	07708-465789	pinky_allie@pook.ie	19	Jack	Yorkshire Terrier	20/09/2008	20/01/2009
Sherrington	Alan	01234-645672	a_and_p@hbar.ie	21	Spot	Dalmation	02/09/2009	02/02/2009
Mears	Clare	01234-890123	breast@rope.com	6	Jewi	Scottish Water Dog	20/09/2008	20/02/2009
Stobart	Colin	07708-123456	colin@diggee.net	22	Willms	Rat Terrier - Gobian	02/06/2008	18/01/2009
Cannon	Eric	07708-864378	s.cannon@lodge.ie	12	Fial	Rat Terrier - Gobian	02/06/2008	18/01/2009
Thomas	Mike	01345-234567	mthomas@stuv.com	4	Sam	Yorkshire Terrier	10/06/2008	10/12/2008
Green	Monika	01843-123751	greenie.m@barav.ie	3	Mindi	Yorkshire Terrier	06/09/2008	02/02/2009
Davis	Cliver	01243-234567	dillidavis@over.ie	2	Kita	Japanese Arita	10/05/2008	10/11/2008
				15	Esodie	English Setter	30/06/2008	28/02/2009

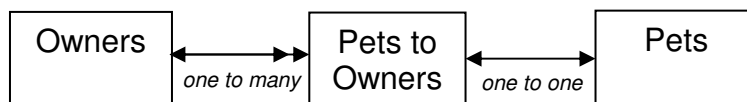
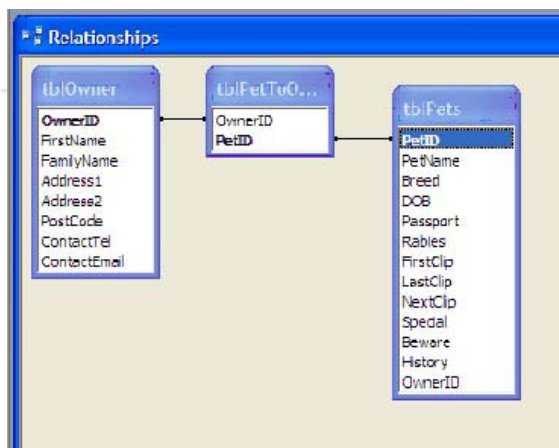
Click on the print button to print a copy out. Click on "Close" to return to the main menu form.

Table definitions:
tblPets

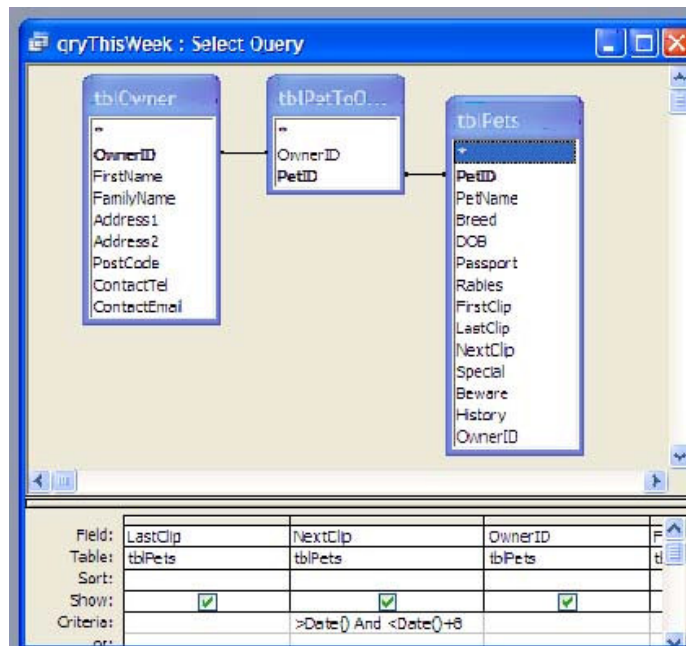
Field Name	Data Type	Description
PetID	Number	RECORD KEY, unique number
PetName	Text	Free text
Breed	Text	Free text
DOB	Date/Time	Format dd/mm/yyyy
Passport	Text	Free text
Rabies	Yes/No	Combo box with choices: Yes, No
FirstClip	Date/Time	Format dd/mm/yyyy
LastClip	Date/Time	Format dd/mm/yyyy
Special	Memo	Free text
Beware	Memo	Free text
HISTORY	Memo	Free text
OwnerID	Number	Integer - RECORD KEY in tblOwners

Form layout:
frmNewOwner

Relationship diagram



Query for working out who is due a clip this week: qryThisWeek



VBasic behind "Save details..." button on frmNewDog

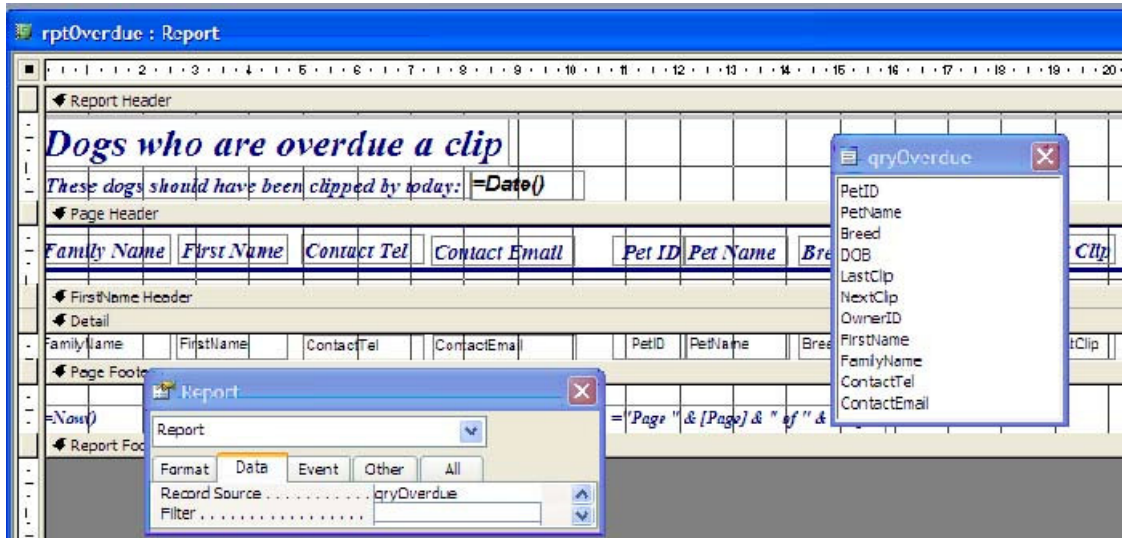
```

Private Sub Command30_Click()
    On Error GoTo Err_Command30_Click
    'Get owner ID into tblOwner
    Text46 = Text33
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70
    DoCmd.Close
    Exit_Command30_Click:
    Exit Sub

Err_Command30_Click:
    Select Case Err.Number
    Case 3022
        'duplicate dog ID
        MsgBox "You have entered a dog identity number which already exists." + Chr(13) + Chr(13)
        + " Please try another ID." + Chr(13) + Chr(13) + "Hint: look at the record number in the record
        selector below.", , "--- Canine Cuts --- Sorry, an error has occurred."
    Case 3101
        'owner ID doesn't exist
        MsgBox "You have entered an owner identity number which does not exist." + Chr(13) +
        Chr(13) + " Please check the owner ID and enter again.", , "--- Canine Cuts --- Sorry, an error
        has occurred."
    Case Else
        'another error so display message raised by error code
        MsgBox Err.Description, , "--- Canine Cuts --- Sorry, an error has occurred."
    End Select

Resume Exit_Command30_Click
End Sub
    
```

Design for rptOverdue



Evaluation and Future Developments

Evaluation

Some suggestions.....

The solution as I have created it will fulfil all the objectives that I set for a system. It is not a terribly complicated system but it works well and does have great potential.

Mrs Bouchette wanted a system that was easy to use, that data entry was simple and accurate, that retrieving details of either dogs or owners would be quick. The system does each of these things and also provides the reports that she requested.

.....

The system provides a simple way of connecting dogs to an owner, one of the difficult things in the old system. The old system was going to soon have a problem with the volume of data that was kept for one dog. The new system allows a very large amount of data to be stored for one dog all in one place – in the old system information for one dog would soon outgrow one index card and that was one area of concern, how to keep the information accurate and easily ordered.

.....

The new system does not have many features such as validation routines or automatic data entry but that is the nature of the system and the data within it. Most of the data entry is of a reporting kind and the facility for doing this is easy. The system does trap errors such as duplication of record keys – giving a new owner or dog an existing ID number. The system will tell you if you want to add a dog to a non-existing owner. These errors generate a Canine Cuts specific error message box, produced by routines I have written in Visual Basic rather than simply relying on the jargon laden message boxes that Access produces.

.....

I wanted to be able somehow to validate the email address to make sure that it was correct but wasn't able to do it. Maybe I could have put a second data entry field for entering an email address for a second time, like sometimes happens on web pages, and then check that they were equal, but then I thought that it was a messy way to do it on something like this.

.....

The routine for adding a dog to an owner is a bit different to how I wanted it to be. My idea was to be able to do it somehow through the owner detail form (Edit an owners details), where there is the subform with the dogs name(s) in it. I couldn't get this right and that is why it has turned out as it has. However, I think that the solution as it is at the moment is pretty good and does make things quite simple really.

Future Developments

Some suggestions....

It would be a good idea for the dog's detail page to have a photograph a photograph. This could be scanned from the dog's passport. Mrs Bouchette had mentioned if we needed a scanner earlier on. This would be a use for one. Scanners are pretty cheap - £50 range, or even we might think about buying a scanner/printer/photocopier (so-called 3-in-1) for not much more. Another way to get a photo in would be to use Mrs Bouchette's digital camera and then upload the photo, either by cable or a card reader. In the database a new field will need to be created to link a photo to a dog's record (field type will be OLE).

.....

The future appointments idea could be extended. At the moment Mrs Bouchette can print out a report telling her of appointments in the coming week. Why not change the date in the query for this to say a month before. The owners could then be contacted to remind them. It should be possible to connect this query output to a mailing system to create an automatic mail-shot. Mrs Bouchette uses Outlook as her email system so there should be some way to manage this. It is certainly worth investigating.

.....

At present there is no information regarding the vet that an owner uses for their dog. Another table could be created for vet details that are then linked to the owner (many-to-one, owner to vet). This would be useful if there is some problem while the dog is with Mrs Bouchette. Such problems could be to do with tick infections or quite simply accidental cuts.

.....

At the moment all owner and dog details will remain on file 'for ever' as there is no way through the current menu to delete a dog or an owner. Inevitably an owner may move away from the area, dogs will die... At some point the menu needs to be extended to allow deletions to take place. Dogs will have to be deleted from owners first, or when an owner is to be deleted a form appears listing the dogs as a double check. If okayed then the process will delete from both dog and owner tables. Some policy will need to be created as to how to reallocate old owner and dog IDs.

**Tackling the
COMPUTER STUDIES
Coursework
Project**

ALL RIGHTS RESERVED. No part of this book may be reproduced, stored in a retrieval system, or transmitted by any other means: electronic, mechanical, photocopying, recording, or otherwise, without prior written permission from Colin Stobart.

©2009 Colin Stobart, colin@ictgcse.net