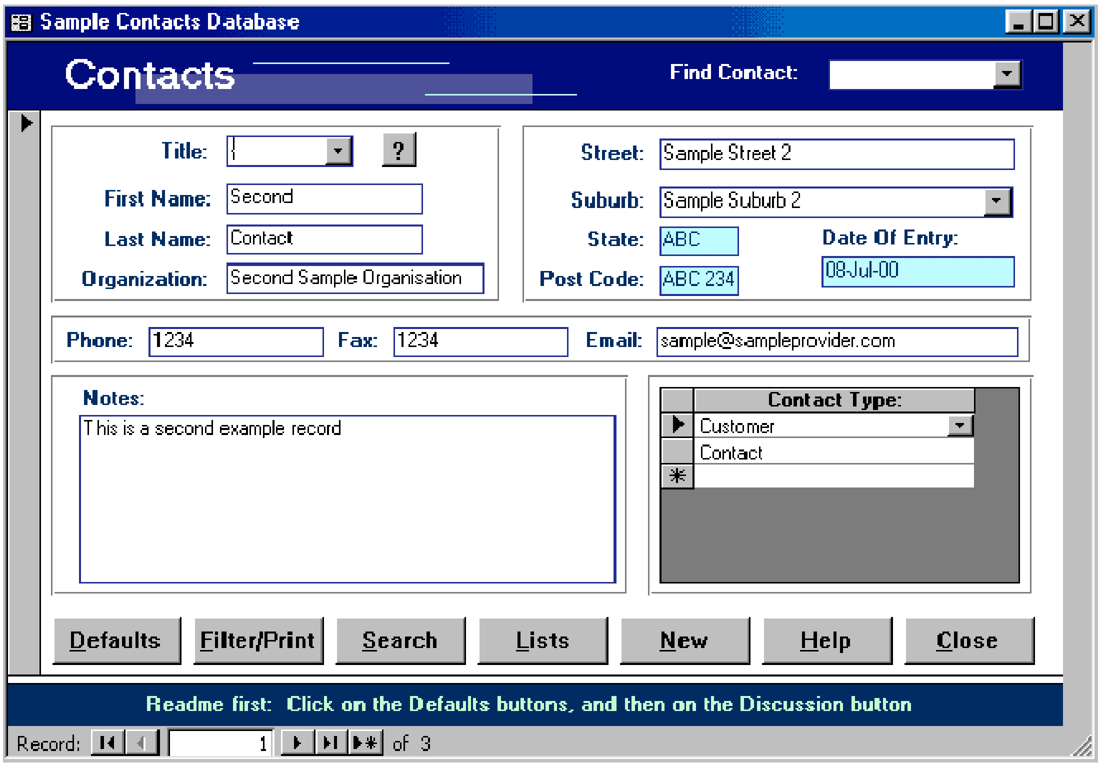
Functional Database Design

A Toolkit for Developers



Sample Contacts Database

Contacts

Find Contact:

Title:

First Name: Second

Last Name: Contact

Organization: Second Sample Organisation

Street: Sample Street 2

Suburb: Sample Suburb 2

State: ABC

Post Code: ABC 234

Date Of Entry: 08-Jul-00

Phone: 1234

Fax: 1234

Email: sample@sampleprovider.com

Notes:

This is a second example record

Contact Type:
Customer
Contact

Readme first: Click on the Defaults buttons, and then on the Discussion button

Record: 1 of 3

Written by:
Mark O’Reilly

Functional Database Design

Initial Draft Contributors:
- Mark O'Reilly
- Neil Schwenke

Reviewed by:
- David Abercrombie

Thanks to:
- David Abercrombie
- Neil Schwenke
- Stuart Garner, Knowledge Base
- Ruth Christie and Denise Taylor
- Phil Frankel
- Kathy Vare
- Phil Whiteley
- Rob Brown and MSA Business Systems
- Queensland University of Technology
- The Brisbane Institute of TAFE
- The Moreton Institute of TAFE
- Jordan Russell for Inno Setup
- The Microsoft® Corporation

This does not imply endorsement of this book by any of these organisations or individuals.
This text is not sponsored, endorsed, or affiliated by or with Microsoft® Corporation.

Published by MAJOR Software and Text Publishing, Brisbane, 2001
www.majorsoftware.com.au

ISBN: 0 9578725 2 6

Contact:
- Mark O'Reilly
  contact@majorsoftware.com.au

Where this text has been made available in 'soft copy' and can be altered and amended as required, you are requested to cite any variations to the original material clearly and fully. Mark O'Reilly and MAJOR Software and Text Publishing cannot be held responsible for any adjustments and amendments made to the original text.

If you find any omissions, oversights or errors within the text, or wish to offer suggestions, please email Mark O'Reilly at contact@majorsoftware.com.au.

Contents

Analysis and Design

Initial Analysis and Design Tasks

The author considers Functional Database Design to be a slice or portion of the larger systems analysis and design process for database applications. It specifically focuses on the design of appropriate application screens and reports in accordance with the system Data Flow Diagrams and table design.

Outside the scope of Functional Database Design, yet fundamental to the design and development of a database application, the following preliminary tasks would be completed by a systems analyst:

1. Make a preliminary determination of what the client wants their database or computer system to do.
2. Analyse the requirements and determine the feasibility of the database project.
3. Carefully clarify who the users are, and what their individuals needs will be.
4. Create a set of Data Flow Diagrams that describe the flows of information around the organisation or the part of the organisation that is requiring the database.
5. Create a data dictionary that describes these flows in terms of their component data elements and respective volumes.
6. Based on the information flows, the data flow diagrams and the data dictionary, define the entities, attributes and relationships for the database.
7. Construct an Entity Relationship Diagram (or equivalent), and design correctly normalised tables.
8. Create these tables with appropriate entity and referential integrity within your Database Management System.

Functional Design

The design process and the creation of database tables for a database application are well documented - there are many excellent texts around for the study of Systems Analysis and Database Design. This text is not one of those! This text will aim to assist in the planning of database screens and reports.

The design of database tables is fundamental to the success of any database application, as are the foundations to any building. However, functional design is also a critical component which needs to be planned with due care.

Functional design focuses on the layout of menus, data entry screens and the various reports. It also incorporates the navigation and automation within an application. These are almost art forms in their own rights.

The steps being suggesting for Functional Database Design are:

1. With close reference to your Data Flow Diagrams, identify your *application's* Objectives and Tasks.
2. With reference to these Objectives and Tasks, roughly decide on the individual screens and reports, and identify an optimised navigation system between them.
3. With reference to the nature of the business application, decide on a 'look and feel' or style for the interface.
4. With reference to the various Tasks of the database, sketch the individual screens and report layouts.
5. Review the plan - check back with your Data Flow Diagrams and what you have planned in steps 1 to 4 to ensure your design will enhance and optimise business performance.
6. Develop a prototype of the database application.
7. Walkthrough the plan and prototype with the client.

It is acknowledged at the outset that what is being offering here is **not new** - database development has been around for many many years. This text may be useful as a supporting document and workbook to accompany your choice of System Analysis and Design text(s).

Data Flow Diagrams

It is not the intention of this text to discuss the development of Data Flow Diagrams. This process is likely to be well presented in your Systems Analysis and Design Text(s).

Nevertheless, a brief discussion might be valuable. There can often be a resistance by students and even developers to spend time on the development of Data Flow Diagrams, especially for small database applications. The diagrams are sometimes considered a nuisance to create, or may not offer any intrinsic value.

Nevertheless the process of developing Data Flow Diagrams - the process itself - has the intrinsic value of **helping to identify the questions that have not been asked or answers that have not been heard**. During initial interviews with a client, there can be a considerable amount of discussion. It is often the case that some questions do not get asked, do not get answered fully, or the answer is forgotten (oh to be human…).

When creating the Data Flow Diagrams, there will be external entities, flow lines, processes and data stores that the analyst will have some uncertainty about. The skill is to identify when these moments of uncertainty occur and to stop ones-self from just putting in an entity, line, process or store - and then assuming it is correct. In making time to create Data Flow Diagrams, this time also serves to check that all the information about the system is in.

Data Flow Diagrams can also provide detail that assists in the identification of the database tables. The data stores specifically provide leads that help illuminate the database table design, compliment the development of Entity-Relationship diagrams and support the process Normalisation.

And finally, **another of the intrinsic values of Data Flow Diagrams is in relation to Functional Database Design** - assisting with the identification of screen and report content, application navigation and general interface design…

A Sample Contacts Database

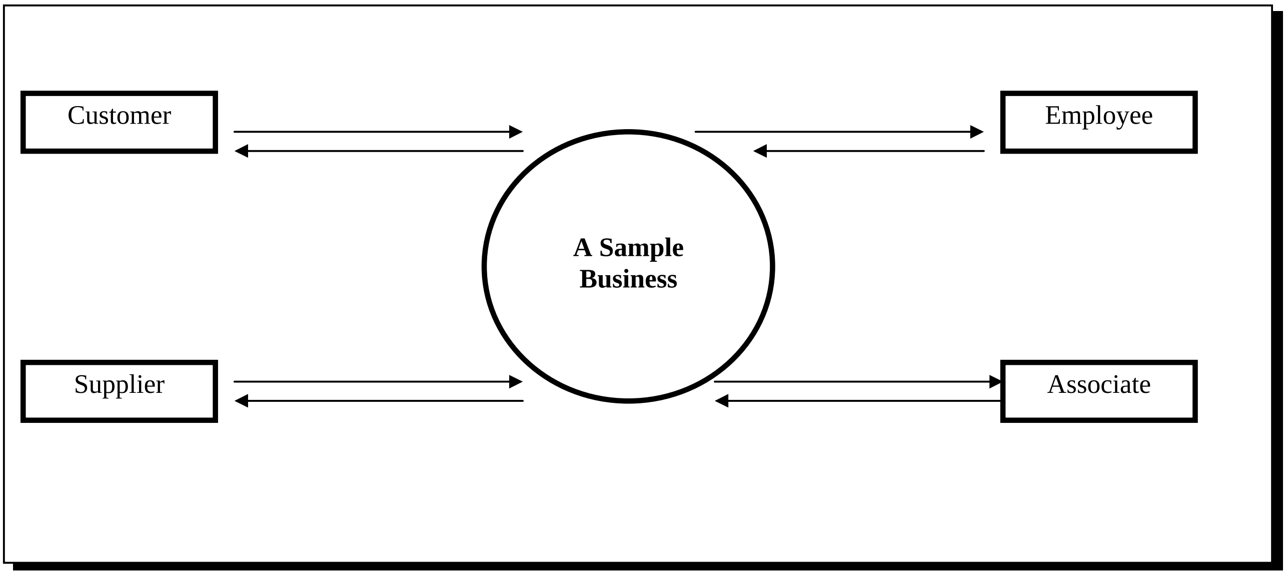
In order to illustrate the process of Functional Database Design, a sample Contacts Database is discussed. It was developed for illustration purposes, using Microsoft® Access.

Let us imagine that your client has a business that has a steady flow of development projects. These projects might be the building of cubby houses or pergolas, the installation of networks, or the organisation and management of events such as weddings. Let us also say that your client currently manages their invoicing and scheduling using other existing software.

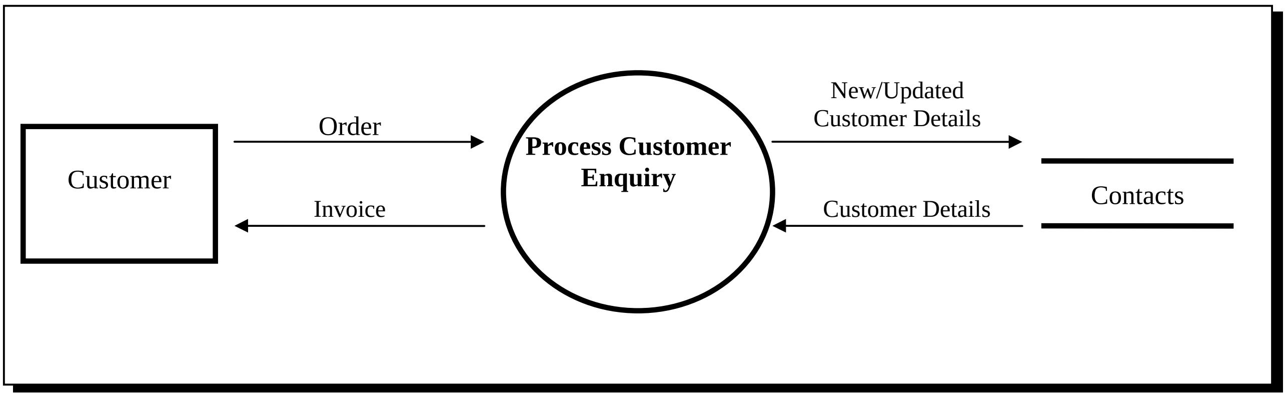
The client has asked you to develop for them a relatively simple database for the management of contacts - customers, suppliers, employees, associates, and/or subcontractors etc.

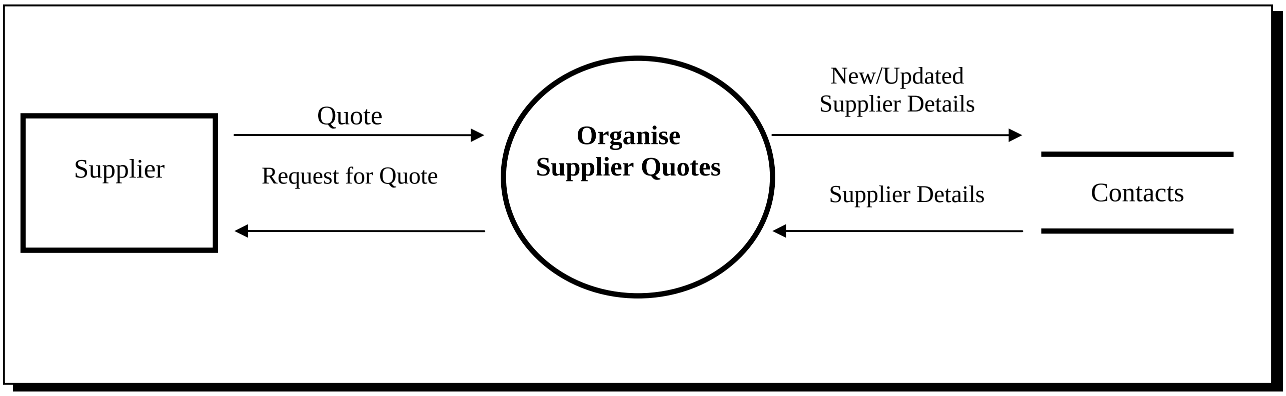
After a couple of interviews you develop a set of Data Flow Diagrams relevant to this development. Some partial Data Flow Diagrams are presented below:

Context Diagram (starter diagram only)



Sample portions of a Level 0 Diagram





1. Objectives and Tasks

Reading Data Flow Diagrams

Data Flow Diagrams provide valuable data in relation to the screens required for your database application. They can assist in identifying the number of screens, how they might need to be linked, and what data might need to be entered and viewed on each screen.

For instance, an employee responsible for ‘Process Customer Enquiry’ (as illustrated on the previous page) would need to be looking at a computer screen that allows them to add, edit and view the customer’s details. The screen would also need to be fitted with efficient search facilities so that existing Customer details could be found quickly.

Further, an employee responsible for the ‘Organise Supplier Quotes’ process would need a screen that allows them to an add, edit, search and view Supplier details.

In both these processes, the Data Flow Diagrams suggest that details of Customers and Suppliers could be located and displayed, and that new and updated details could be entered. The similarity in these processes might suggest that the data stores be joined to make one large data store for all contacts - holding details of customers, suppliers, and employees.

This may also be reflected in the normalisation of the database tables.

Objectives

The next step is to generate a list of objectives specifically for the database application. These can be gleamed from the overall system objectives and goals, the Data Flow Diagrams and other information gained at the interview(s). Objectives are broad statements that identify what a small database application or a component of a larger database is aiming to achieve. For instance, the objective of the Contacts database (discussed above) might be stated as:

> *To allow the business to manage the details of all business contacts - customers, suppliers, and employees.*

Other examples of objectives (for other databases) might include:

1. To process 6 customers at the counter at one time, at less than 20 seconds per customer.
2. To create and electronically send weekly orders with minimal human intervention.
3. To maintain data on clients, employees and sales representatives.
4. To automate appointment schedules for all employees.
5. To statistically analyse the weekly takings of each section with a view to determining reordering requirements.
6. To maintain a catalogue for employee and client use.
7. To present a professional image for the organisation.
8. To maintain a registry of video loans with a view to analysing loan patterns by time of the day, day of the week, time of the year, by specials, and by customer home postal area.
9. To speed the update of patient records, minimise the hardcopy duplication, and reduce re-entry of data.
10. To maintain stock records across the 5 satellite stores within the organisation, and provide for central control from the main store.

Tasks

Each of the objectives can be broken up into a number of tasks. These tasks might well align closely with specific screens and/or reports in your database application. (Initially, you may wish to identify one task for each screen or report in the application.) You can think of tasks as being jobs or functions that the database application must perform for the users.

For instance, the tasks for the Contacts database (discussed above) might be stated as:

1. To add, edit, and review details of Customers.
2. To add, edit, and review details of Suppliers.
3. To add, edit, and review details of Employees.

etc…

This breakdown of tasks would reflect that the customers, suppliers and employees are stored in separate tables and have separate data entry screens. Alternatively, the tasks could reflect that the contacts are all placed in one table and filtered by Contact Type as required. Such alternate tasks might be stated as:

1. To add, edit and view details of business contacts.
2. To search for a business contact by suburb or contact type.
3. To locate a business contact using a search string.
4. To manage various lookup lists.
5. To print a full listing of contacts.
6. To print lists of contacts filtered by suburb and/or contact type.
7. To print mailing labels filtered by suburb and/or contact type.

While developing your tasks it is important to refer to your entity -relationship diagram (or table design) to ensure that:

1. screens and interfaces have been provided for all data entry and management, and…
2. all views and reports can be created from your entities and relationships.

Other examples of tasks and their associated objectives might be:

Objective:
To maintain records of customers, orders and products.

Associated Tasks:

1. To allow the entry, sorting and searching of information relating to customers.
2. To allow the entry, sorting and searching of information relating to products.
3. To allow the entry, sorting and searching of information relating to customer orders.
4. To provide a hardcopy list of customer information.
5. To provide a mailing list for all or selected customers.
6. To produce a product catalogue for sending to customers.
7. To generate a marketing style letter addressed to the individual customers.
8. To provide statistical analysis of customer orders by location, by sales, and by organisation type.

Objective:
To process employee pays.

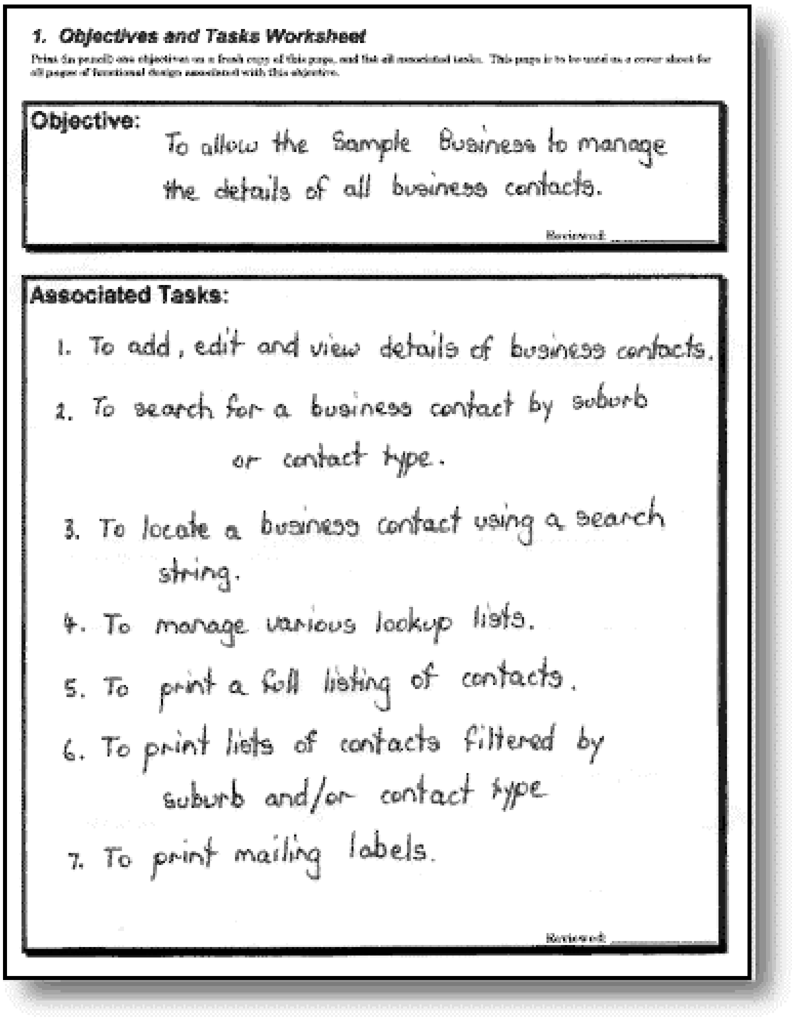
Associated Tasks:

1. To record sales reps against customer orders.
2. To allow for the entry of weekly employee hours and rates of pay.
3. To generate a weekly pay slip for each employee.
4. To generate a weekly tax report with individual employee tax payments and a management summary.
5. To generate appropriate weekly superannuation documentation and an employer contribution summary.
6. To generate weekly, a detailed management report of employee wages, and an associated summary report.

Objectives and Tasks Worksheet

In Appendix A is a worksheet titled: **1. Objectives and Tasks Worksheet.** This worksheet is for handwriting the objectives and tasks for your client's database application. Use one copy of this worksheet for each objective and under each objective list the respective tasks. An example worksheet has been completed for the Contacts database and is shown below:

NOTE: the section on the bottom right of the worksheet titled: 'Reviewed:' is to be completed during Step 5 - Review the Plan.



1. *Objectives and Tasks Worksheet*

Print (in pencil) one objective on a fresh copy of this page, and list all associated tasks. This page is to be used as a cover sheet for all pages of functional design associated with this objective.

Objective: To allow the Sample Business to manage the details of all business contacts.

Reviewed: ______________

Associated Tasks:

1. To add, edit and view details of business contacts.
2. To search for a business contact by suburb or contact type.
3. To locate a business contact using a search string.
4. To manage various lookup lists.
5. To print a full listing of contacts.
6. To print lists of contacts filtered by suburb and/or contact type
7. To print mailing labels.

Reviewed: ______________

2. The Initial Draft

Translating Objectives and Tasks

In developing the Objectives and Tasks, you may have given some thought to the individual screens and reports and the navigation between these within your application. The initial process of deriving the Objectives and Tasks and considering the best screens and reports for your application is probably the most difficult step (though it will become easier with experience).

This step is a first draft at identifying your screens and reports and deciding how they will be linked within the application. This step is a creative one and it is important that you do not get caught up in any technical details and considerations.

What you need to do is to **imagine that you are the officer executing a given Data Flow Diagram process** and that you are looking at one of more computer screens trying to manage the process. Imagine yourself looking at the computer screen and doing the job of the officer executing the process.

What sort of data would you like to work with on the primary screen? Would you need to add data, edit data, view data? What data? What secondary data would be presented on the computer as you add/edit this data? What other information would you like to access on popup screens or in lists as you work?

To stress again - **this is a creative process (a visualisation) and requires you to use your imagination.**

As thoughts come to mind, you need to jot these down on paper (you can often draw, write and edit on paper faster than on a PC). The idea here is to promote your creativity, remove any technical shackles, and ensure that you can record your ideas quickly and without significantly interrupting your visualisation.

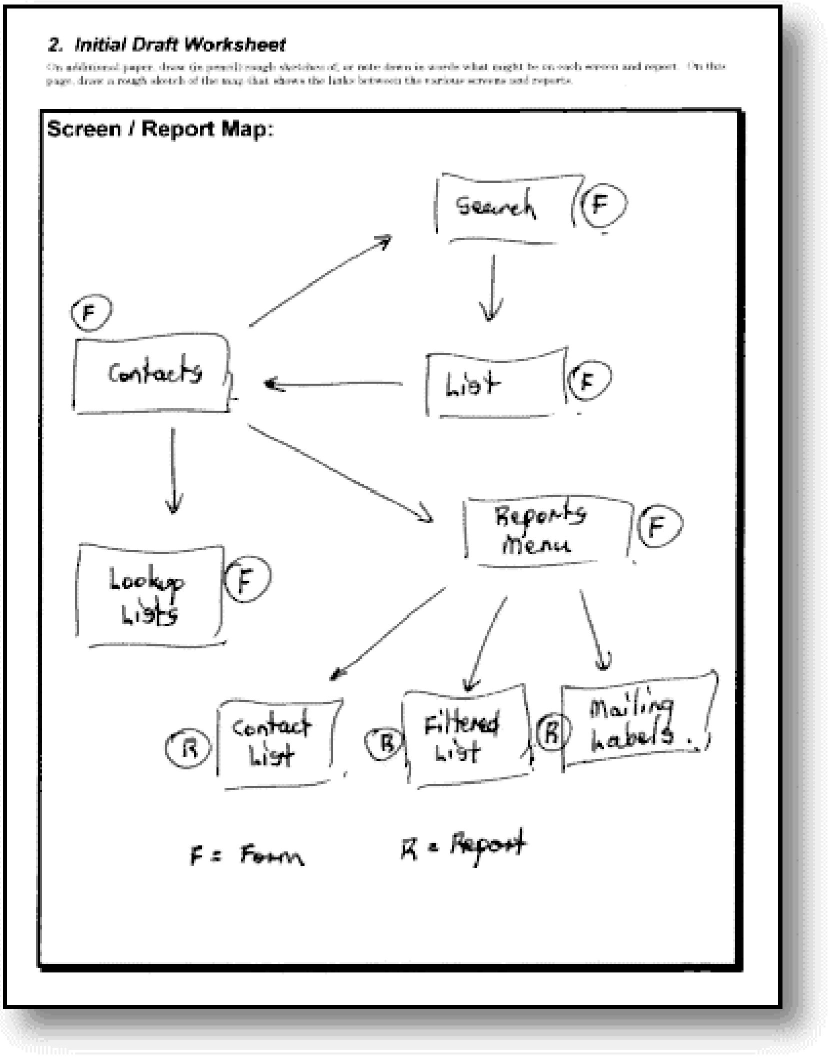
What you write and draw is really up to you. However, a couple of suggestions might be:

- Make some very rough sketches of the individual screens and reports.
- Draw rectangles for the screens and reports and write a couple of words or sentences suggesting what data might be on each.
- Draw a simple map of how the screens and reports are linked.
- Cut out your rough sketches and stick them together so you can flip the popup screens and reports over the top of the primary screen(s).

Initial Draft Worksheet

In Appendix A is a worksheet titled: **2. Initial Draft Worksheet.** This worksheet is for drawing a map of the components within your application. Additional paper can be used for sketching the various screens and reports.

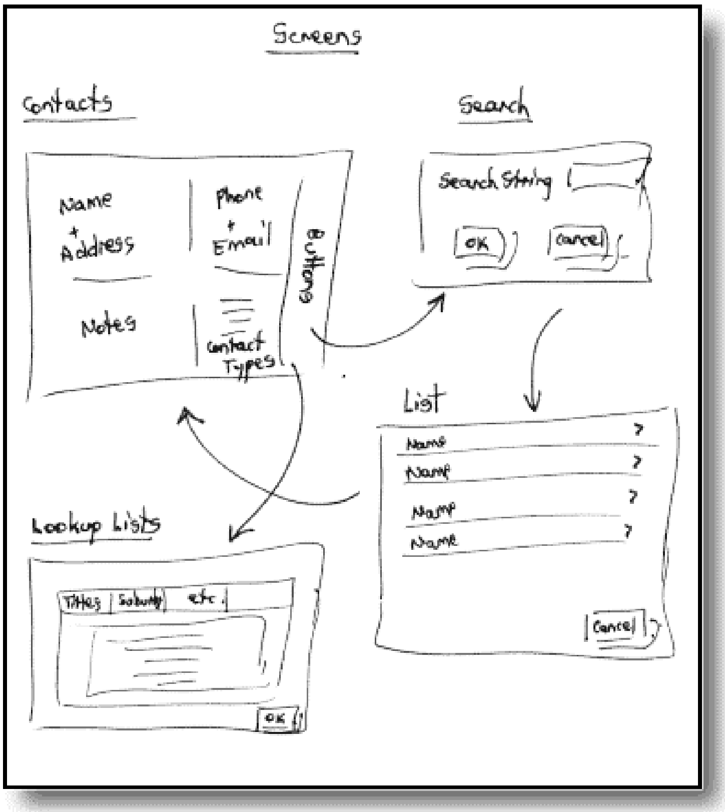
An example first draft for the Contacts database is given below.

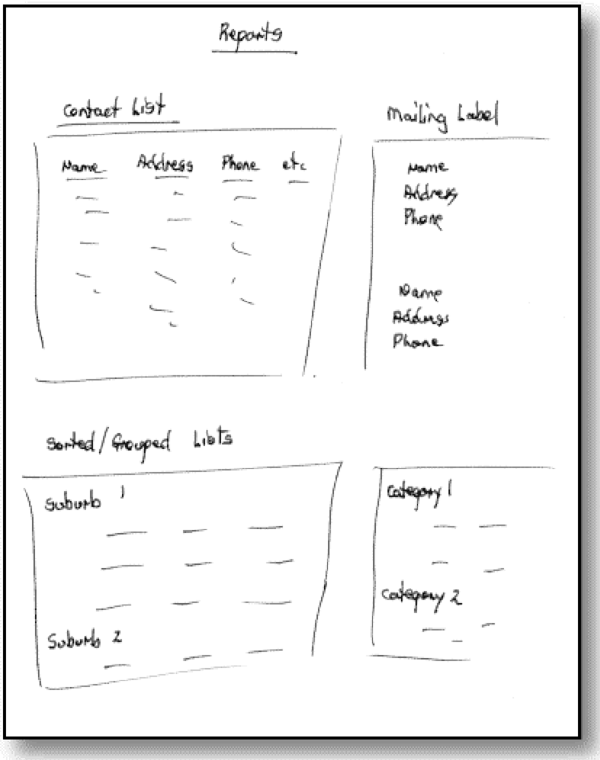


2. *Initial Draft Worksheet*

On additional paper, draw (in pencil) rough sketches of, or note down in words what might be on each screen and report. On this page, draw a rough sketch of the map that shows the links between the various screens and reports.

Screen / Report Map:


Screens
Contacts
Search
Name + Address
Phone + Email
Buttons
Notes
Contact Types
Search String
OK
Cancel
List
Name
Name
Name
Name
Cancel
Lookup Lists
Titles
Suburb
etc.
OK


Reports
Contact List
Name
Address
Phone
etc.
Mailing Label
Name
Address
Phone
Name
Address
Phone
Sorted / Grouped Lists
Suburb 1
Suburb 2
Category 1
Category 2

3. The Interface Style

Templates

One suggestion for standardising the 'look and feel' of your screens and reports is to create templates of each in advance. These templates could then be copied from one application to the next, or could form the basis of a 'boilerplate' on which you build each new database.

There is usually a finite number of screen and report types that you would commonly use in an application. For instance, some common **screens** are:

- A standard Data Entry screen
- A Main/Sub screen
- A List screen
- A Lookup List screen
- A Filter screen
- A Menu screen

Depending on the operating system and the development environment, screens can have:

- title bars with captions,
- form headers which can include the current date and time,
- fields for adding, editing and/or viewing data,
- dropdown lists and list boxes,
- buttons or function keys for screen navigation and various record operations,
- status bar help, and…
- screen footers.

Reports are usually portrait or landscape, and although most are for A4 paper, some can be for other paper sizes. As a minimum, it can be helpful to develop two A4 report templates - one for portrait and one for landscape reports. Within each of these templates, standard report and page headers and footers should be created.

Report headers and footers will often include:

- a descriptive report title,
- date and time of printing,
- page numbering,
- the name of the application,
- the name and various details of the client's organisation.

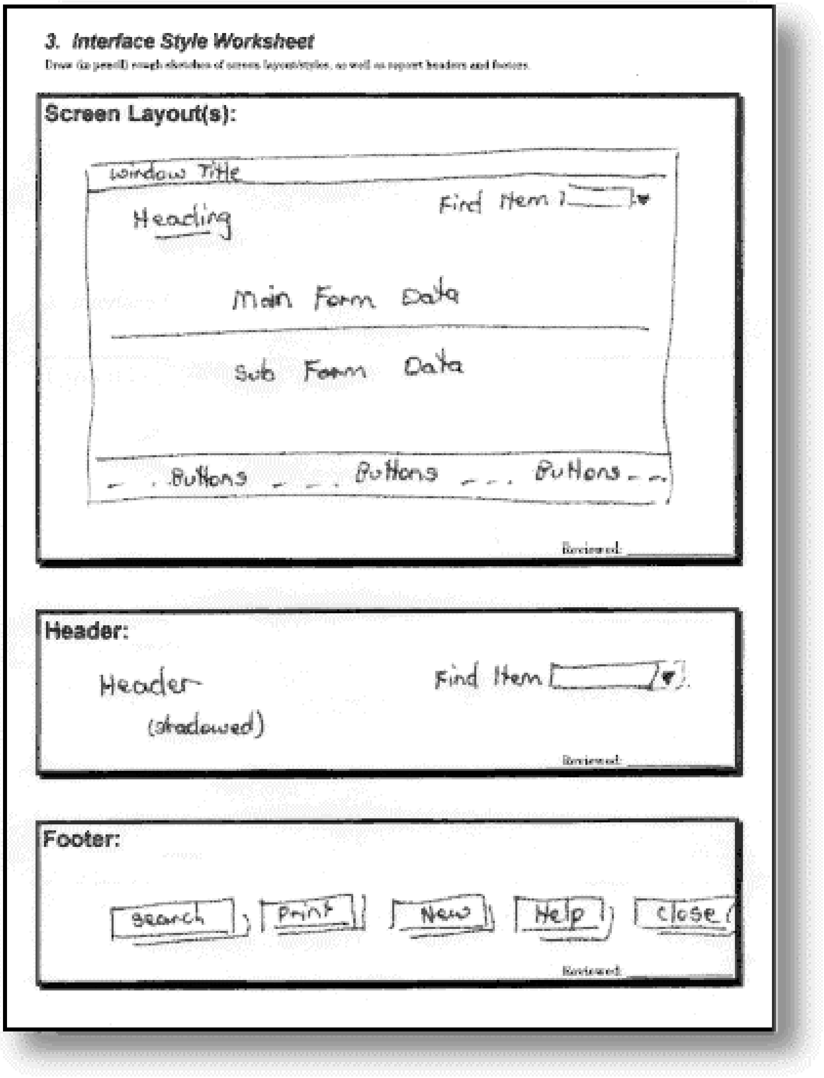
As with the Initial Draft, the planning of the Interface Style is a creative process. It requires some quiet reflection on the nature of the application and some deliberation as to what may be needed on the various screens and reports. A little extra time spent on this step can save considerable time later, and can help ensure consistent look and feel across your application.

Interface Style Worksheet

In Appendix A is a worksheet titled: **3. Interface Style Worksheet.** This worksheet is for roughly sketching a number of standard screens for your application, such as: a menu, a data entry screen and a popup filter screen.

The worksheet also allows you to roughly sketch the header and footer for your screen and report templates.

A sample template for the Contacts database is given below.



3. Interface Style Worksheet

Draw (in pencil) rough sketches of screen layouts/styles, as well as report headers and footers.

Screen Layout(s):

Window Title
Heading
Find Item
Main Form Data
Sub Form Data
Buttons
Buttons
Buttons

Reviewed:

Header:

Header
(shadowed)
Find Item

Reviewed:

Footer:

Search
Print
New
Help
Close

Reviewed:

4. Screens and Reports

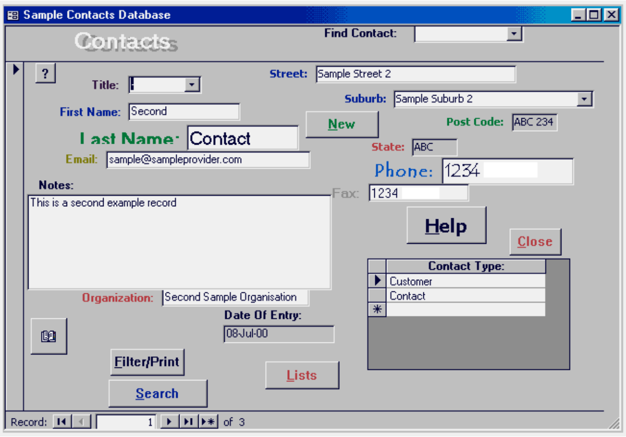
Developing Screens

This step - the development of screens and reports - ties in closely with each of the previous steps. The Objectives and Tasks and the subsequent Initial Draft will directly impact on the number of screens and reports and suggest the data that might appear on each. The Interface Style will impact on the general ‘look and feel’ of the various items, and should reflect the considered templates.

The screens presented below have been broken into a number of groups for the purposes of discussion. The discussion will focus on both the Application Objectives and Tasks as well as the Interface Style. These screens were developed in the Microsoft Access database - Contacts_Interfaces.mdb - a copy of which has been supplied with the text.

Primary Data Entry Screens

Dishevelled



Sample Contacts Database

Contacts

Find Contact:

Title:

First Name: Second

Last Name: Contact

Email: sample@sampleprovider.com

Notes: This is a second example record

Organization: Second Sample Organisation

Date Of Entry: 08-Jul-00

Street: Sample Street 2

Suburb: Sample Suburb 2

Post Code: ABC 234

State: ABC

Phone: 1234

Fax: 1234

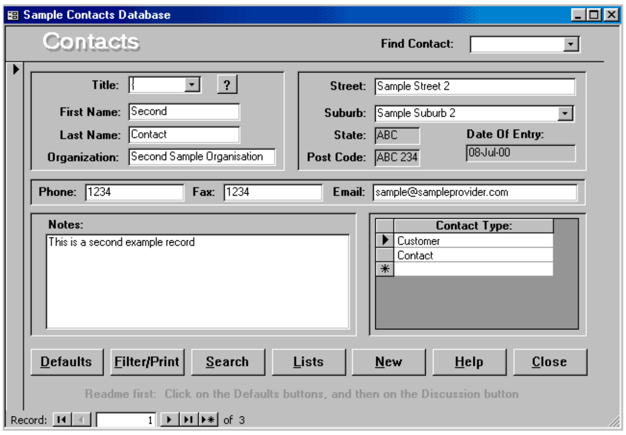
Contact Type:
Customer
Contact

Record: 1 of 3

The first task identified for this database application was: ‘To add, edit and view details of business contacts’. The following example screens should allow a user to do this.

This particular screen shows that using too many colours, too many fonts, too many button sizes and styles and a dishevelled placement of controls can be very distracting and not conducive to productive use in a business environment.

Sunken Boxes



Sample Contacts Database

Contacts

Find Contact:

Title:

First Name: Second

Last Name: Contact

Organization: Second Sample Organisation

Street: Sample Street 2

Suburb: Sample Suburb 2

State: ABC

Post Code: ABC 234

Date Of Entry: 08-Jul-00

Phone: 1234

Fax: 1234

Email: sample@sampleprovider.com

Notes: This is a second example record

Contact Type:
Customer
Contact

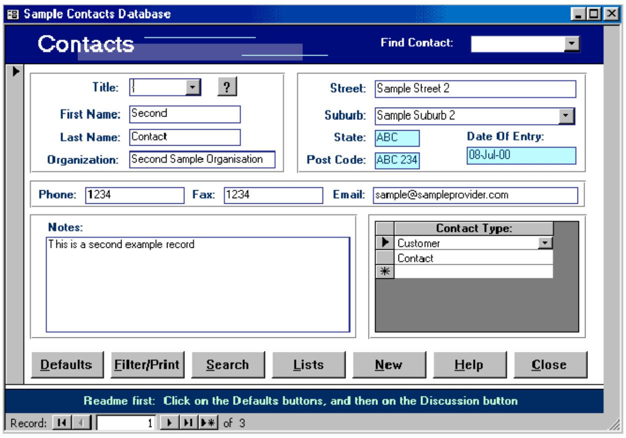
Readme first: Click on the Defaults buttons, and then on the Discussion button

Record: 1 of 3

This screen also fulfils the requirements of the first task. It allows the entry of contact details, whether they be a customer, supplier or employee. It also allows the entry of none, one or more Contact Types per contact. These could be used to filter and sort the various contacts later.

The interface style uses a simple header with shadowed title, and a ‘Find Combo Box’. The dominant style within the screen is sunken boxes, several of which are used to group related items.

Flat



Title:
First Name: Second
Last Name: Contact
Organization: Second Sample Organisation
Street: Sample Street 2
Suburb: Sample Suburb 2
State: ABC
Post Code: ABC 234
Date Of Entry: 08-Jul-00
Phone: 1234
Fax: 1234
Email: sample@sampleprovider.com
Notes: This is a second example record
Contact Type: Customer
Contact

Readme first: Click on the Defaults buttons, and then on the Discussion button

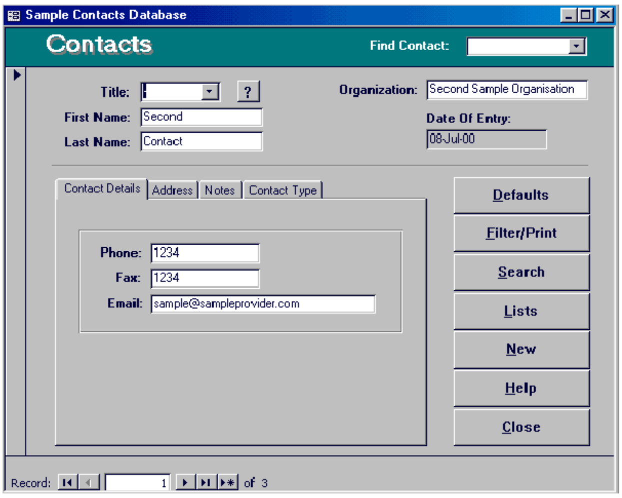
Record: 1 of 3

Screens conducive to use in a business setting can present formal colours and styles. This screen for example, uses a reasonably formal business colour in the header and footer and an unobtrusive design.

The placement of the data controls promotes data entry and review, and the boxing helps the user identify related controls. The flow of the screen is left to right and top to bottom (a Z format), and the controls present similar size, colour and use of font.

All components would appear to be in proportion to the screen and to each other, and present a clear message that this is a data entry screen for Contact details.

Tab Format



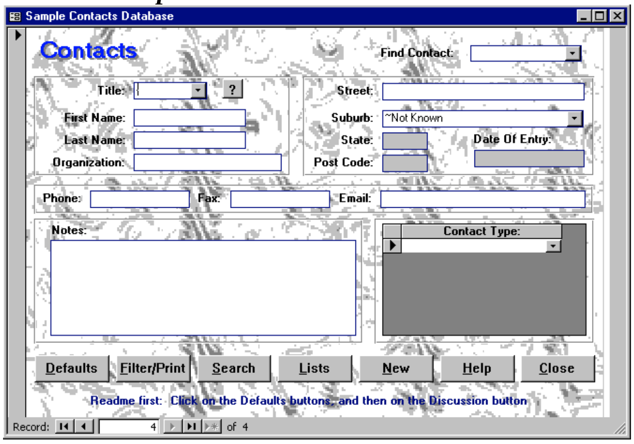
Title:
First Name: Second
Last Name: Contact
Organization: Second Sample Organisation
Date Of Entry: 08-Jul-00
Phone: 1234
Fax: 1234
Email: sample@sampleprovider.com

Record: 1 of 3

When a large number of fields need to be placed on one form, a tab control can be used to manage the various groupings. This has been exampled here. Note that the most commonly accessed data is placed on the main part of the form and the first tab page.

This form also shows the buttons being placed on the right hand side, and sized so they create the appearance of a 'solid' panel of buttons.

With Backdrop



Suburb: ~Not Known

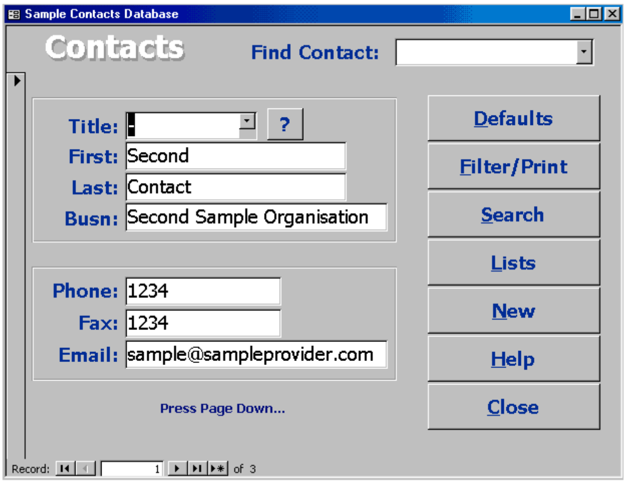
Readme first: Click on the Defaults buttons, and then on the Discussion button

Record: 4 of 4

There may be a preference to create the screen(s) with a more fancy or graphical style. Some database management systems provide graphic images, designs and development tools for this purpose. Such images and designs can be effectively used to enhance consistent screen and report design.

Notwithstanding this, care should be taken not to allow the background design to encroach on the functionality or usability of the screen or report.

Large Controls


Sample Contacts Database

Contacts

Find Contact:

Title:

First: Second

Last: Contact

Busn: Second Sample Organisation

Phone: 1234

Fax: 1234

Email: sample@sampleprovider.com

Press Page Down...

Record: 1 of 3

Business settings can demand specialised screen designs. For instance, a database system might be used in a hectic business environment requiring personnel to enter or view data quickly as they move from one task to another. Large controls, large fonts and large buttons may facilitate such data entry. Clear grouping of data fields and buttons can also be helpful to these types of interfaces.

Other factors that may impinge on the screen design may be: culture or organisation-specific colours and designs, use of grey scale or VGA screens, and human constraints such as colour blindness.

In planning the data entry screens a range of items can be clarified that may set the basis for more formal design or project documentation. These may include:

Item	**Contact's Data Entry Screen Example**
Objective	To allow the business to manage the details of all business contacts - customers, suppliers and employees.
Task	To add, edit and view details of business contacts
Window Title	Sample Contacts Database
Screen Title	Contacts
Screen (form) Name	frmContactsMF
Sub-Screen Names	frmContactCategoriesSF

Underlying Tables and Queries (possibly SQL)	**ContactsMF**:
	SELECT DISTINCTROW Contacts.ContactID, Contacts.DateOfEntry, Contacts.TitleID, Contacts.LastName, Contacts.FirstName, Contacts.OrganizationName, Contacts.Street, Contacts.SuburbID, Contacts.PhoneNumber, Contacts.FaxNumber, Contacts.Note, SuburbList.Suburb, SuburbList.State, SuburbList.Postcode, SuburbList.Country, Contacts.Email
	FROM SuburbList INNER JOIN Contacts ON SuburbList.SuburbID = Contacts.SuburbID
	ORDER BY Contacts.LastName, Contacts.FirstName;
	
	ContactCategoriesSF
	SELECT DISTINCTROW Categories.CategoryID, Categories.ContactID, CategoryList.CategoryText
	FROM CategoryList INNER JOIN Categories ON CategoryList.CategoryID = Categories.CategoryID;

Field Descriptions

Field Descriptor	Manditory	Justify (L/C/R)	Field Type	Size	Conditions / Validation / Format
Title	Y	L	Combo Box List	6	Must be one of: Mrs / Miss/ Mr / Prof / Doctor / Sir / Madam etc…
First	Y	L	Text	20	
Last	Y	L	Text	20	
Business Name	N	L	Text	50	
Phone	N	L	Text	20	Format: (00) 0000 0000
Fax	N	L	Text	20	Format: (00) 0000 0000

etc…

Automation

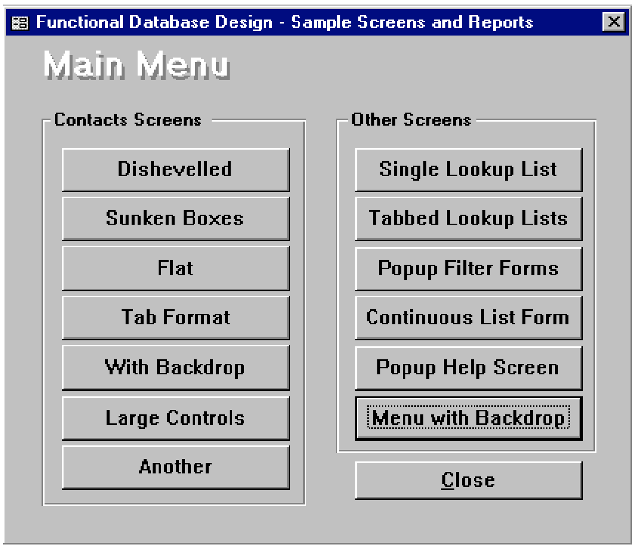
Object	Event Trigger / Source	Function
cboFindContact	After Update	To find a client within the database and display their details on the screen.
cmdNew	Click	Save the current record, clear the screen in preparation for a new entry.
cmdHelp	Click	Display Help for this screen.
cmdClose	Click	Save the current record and close the screen.
etc…		

The Automation table or chart (above) is a variation of a T.O.E. chart (Task - Object - Event), which you may prefer to use. The T.O.E. chart promotes the identification of the task or action that you wish to perform first; and then it requires you to identify the object and the event source to accomplish that task. In comparison the chart above assumes you already know the object and event source and are specifying the function that is to be performed.

You can plan your various screens and reports using the worksheet supplied in **Appendix 4. Screens and Reports Worksheet**.

Menu Screens

Plain

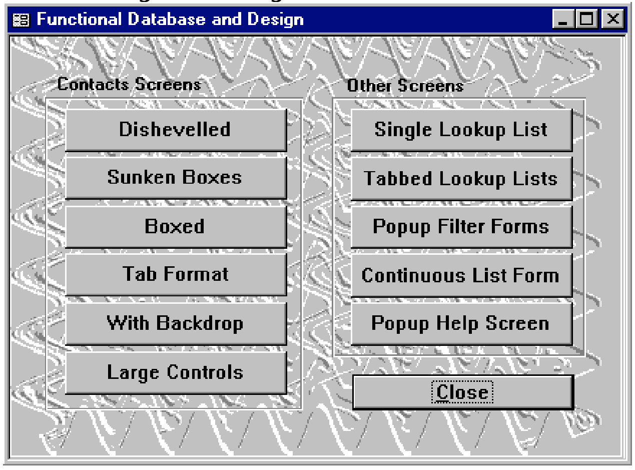


Menus, whether standalone or as part of a data entry screen, need to be clearly labelled and grouped. The captions, labels and instructions should enhance usability and clarify the navigational structure of the application.

The button captions should match with the headers on the various screens and reports subsequently displayed.

A standalone menu screen (such as that shown left) may need to be of a dialog style - not allowing the user to maximise, minimise and/or resize.

With a Background Image



Much of the functional navigation of an application is governed by the various menus. Due consideration must be given to the number of clicks required to navigate between related screens, reports and application components.

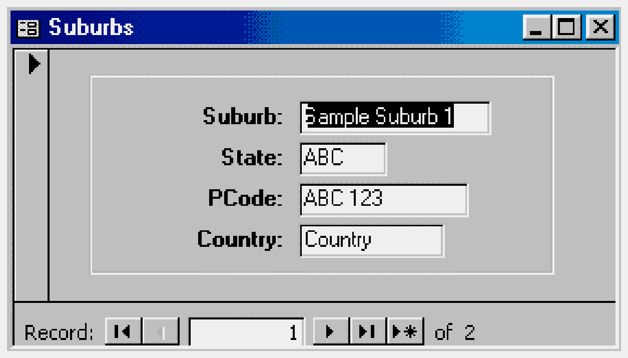
In relation to 'look and feel', design consideration should match those presented for the data entry screens. Notwithstanding this, the Main Menu is the 'front door' to your application and should present a strong first impression.

In many database applications there are screens that a user would use regularly, for instance the Contacts Screen. There are also some screens that the user would visit on first use of an application but only periodically there after. An example of this type of screen might be a data entry screen used to add local suburbs and postcodes.

The question you must then answer is: 'what order do we place the buttons on the Main Menu?' 'Should we place buttons to access the **commonly utilised screens** at the top of the menu?' Or: 'Should we put buttons for the **initial setup screens** at the top of the menu?' Or: 'Is there some other arrangement?'

Lookup Lists

Single Record

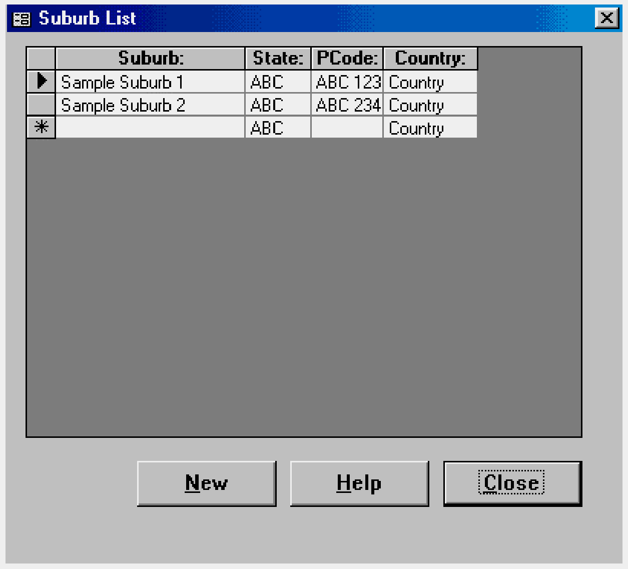


Suburb:	Sample Suburb 1
State:	ABC
PCode:	ABC 123
Country:	Country

There may be quite a few ‘primary’ tables and/or lookup lists in your database. Examples of these might include a list of local suburbs and postcodes, a list of Contact Types, or a list of Business Types.

Screens for the data entry of these tables/lists might display one item at a time, or allow the user to edit several records at once.

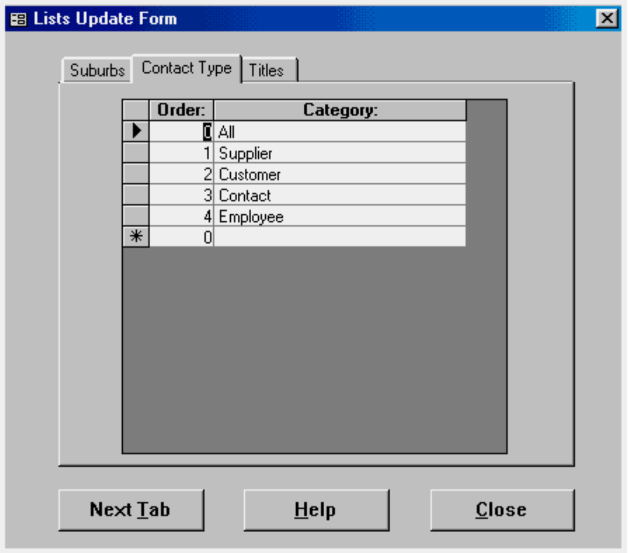
List



Suburb:	State:	PCode:	Country:
Sample Suburb 1	ABC	ABC 123	Country
Sample Suburb 2	ABC	ABC 234	Country
	ABC		Country

A tabular or list style screen can promote usability when the number of fields in a lookup list is small. Being able to move between fields as well as between records can promote input and editing efficiencies.

Tabbed Screen

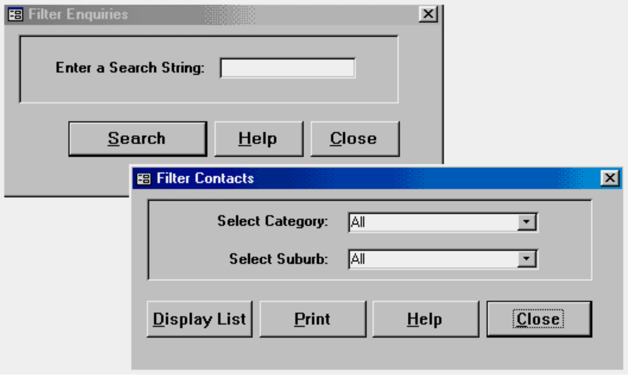


Order:	Category:
0	All
1	Supplier
2	Customer
3	Contact
4	Employee
0	

A tab style form can be used to house multiple list forms with significant savings and efficiency in form ‘real estate’. Buttons relating to specific lists can be added within specific tabs, while other buttons can be placed outside of the tab control.

Filter and List Screens

Filter Screens



Enter a Search String:

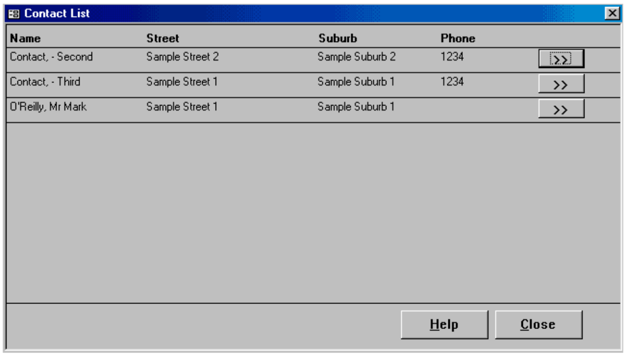
Select Category: All

Select Suburb: All

Other screens that may enhance an application’s functionality are popup filter and list screens. The popup filter screens can play a critical role in narrowing down large volumes of data for display in both ‘list’ screens and in reports.

For instance you may wish to filter out certain records in the Contacts database. This might be achieved using a generic search string on multiple fields, or by using one or more combo boxes, say on Suburb and Client Type.

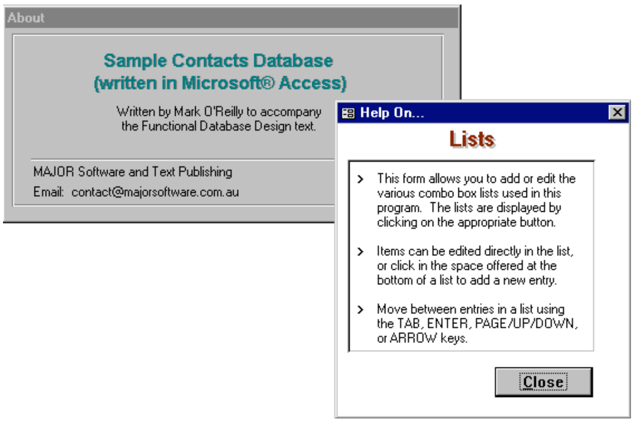
Continuous List of Data



Name	Street	Suburb	Phone
Contact, - Second	Sample Street 2	Sample Suburb 2	1234
Contact, - Third	Sample Street 1	Sample Suburb 1	1234
O'Reilly, Mr Mark	Sample Street 1	Sample Suburb 1	

List style screens can be useful for displaying search results and for allowing a user to select an entry and view the extended record details. You can build extensive functionality into this type of form including search filters, sort buttons, editing capabilities and record selection.

Popup Help and About Screens



Sample Contacts Database (written in Microsoft® Access)

Written by Mark O'Reilly to accompany the Functional Database Design text.

MAJOR Software and Text Publishing
Email: contact@majorsoftware.com.au

Lists

- This form allows you to add or edit the various combo box lists used in this program. The lists are displayed by clicking on the appropriate button.
- Items can be edited directly in the list, or click in the space offered at the bottom of a list to add a new entry.
- Move between entries in a list using the TAB, ENTER, PAGE/UP/DOWN, or ARROW keys.

An ‘About’ box or screen has become a fairly standard addition to many programs and does allow you to identify ownership, acknowledgements and contact details, among others.

Application help can be added as a standard help facility - consistent with the operating system standards, or in the form of simple popup help screens. The former can be the more ‘professional’ if planned, developed and integrated fully, and if due consideration has been given to the user’s needs, the operating system standards, and to appropriate design and layout of on-line documentation.

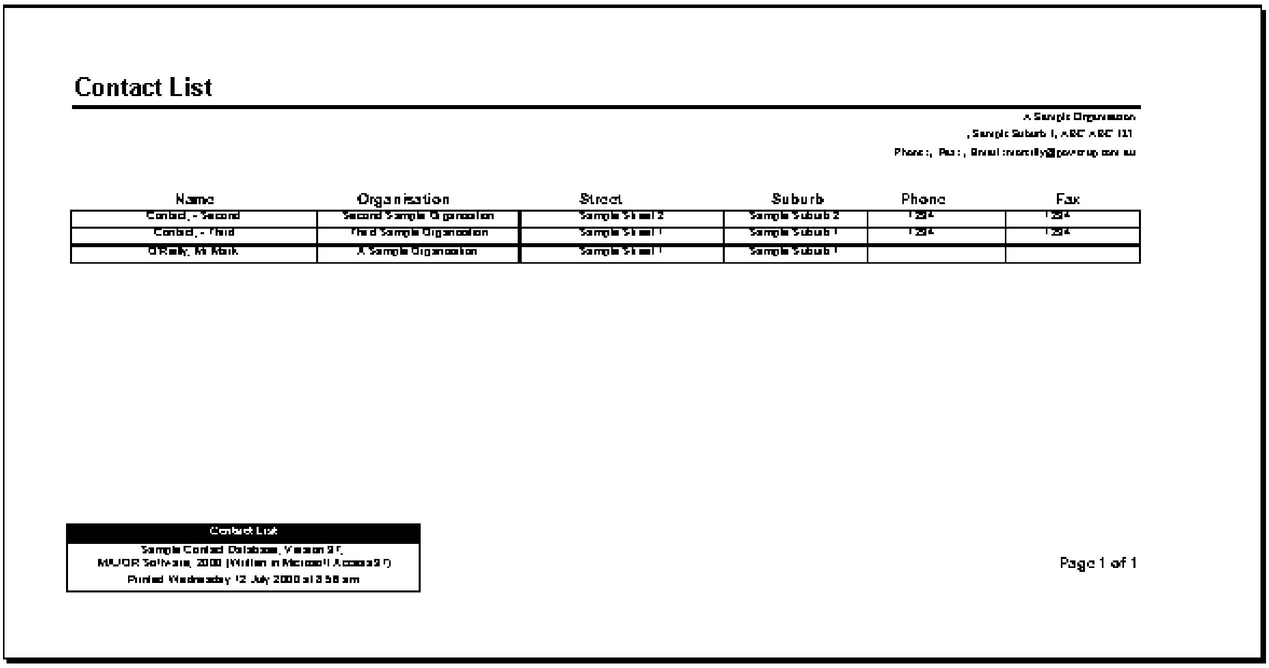
Developing Reports

Many of the principles relating to the design of screens can also be applied to reports. Reports need to be logically and clearly laid out, read left to right, top to bottom and show consistency in the placement of items, in heading styles, in the use of fonts and in the application of colour. Consistency should be applied within the individual reports, but also between the various reports of the application.

Reports often display standard information within the headers and footers of both the report and the individual pages. You need to identify a list of items that are to be placed and then decide where they should appear. For instance:

Item	On Each Page	Header/Footer
Report Title	No	Header
Client Details	No	Header
Date/Time of Printing	Yes	Footer
Page Number	Yes	Footer
Name of Application	No	Footer

- You need to identify what should appear only once at the start of the report, eg: the report title.
- You need to identify what should appear only once at the end of the report, eg: column totals, or maybe an 'end-of-report' notice.
- You need to identify what should appear on each page and whether it appears on the top or bottom of the report.



Contact List

A Sample Organisation
, Sample Suburb 1, ABC ABC 111
Phone: , Fax: , Email: moreilly@powerup.com.au

Name	Organisation	Street	Suburb	Phone	Fax
Contact, - Second	Second Sample Organisation	Sample Street 2	Sample Suburb 2	1234	1234
Contact, - Third	Third Sample Organisation	Sample Street 1	Sample Suburb 1	1234	1234
O'Reilly, Mr Mark	A Sample Organisation	Sample Street 1	Sample Suburb 1		

Contact List
Sample Contact Database, Version 97,
MJOR Software, 2000 (Written in Microsoft Access 97)
Printed Wednesday 12 July 2000 at 3:58 am

Page 1 of 1

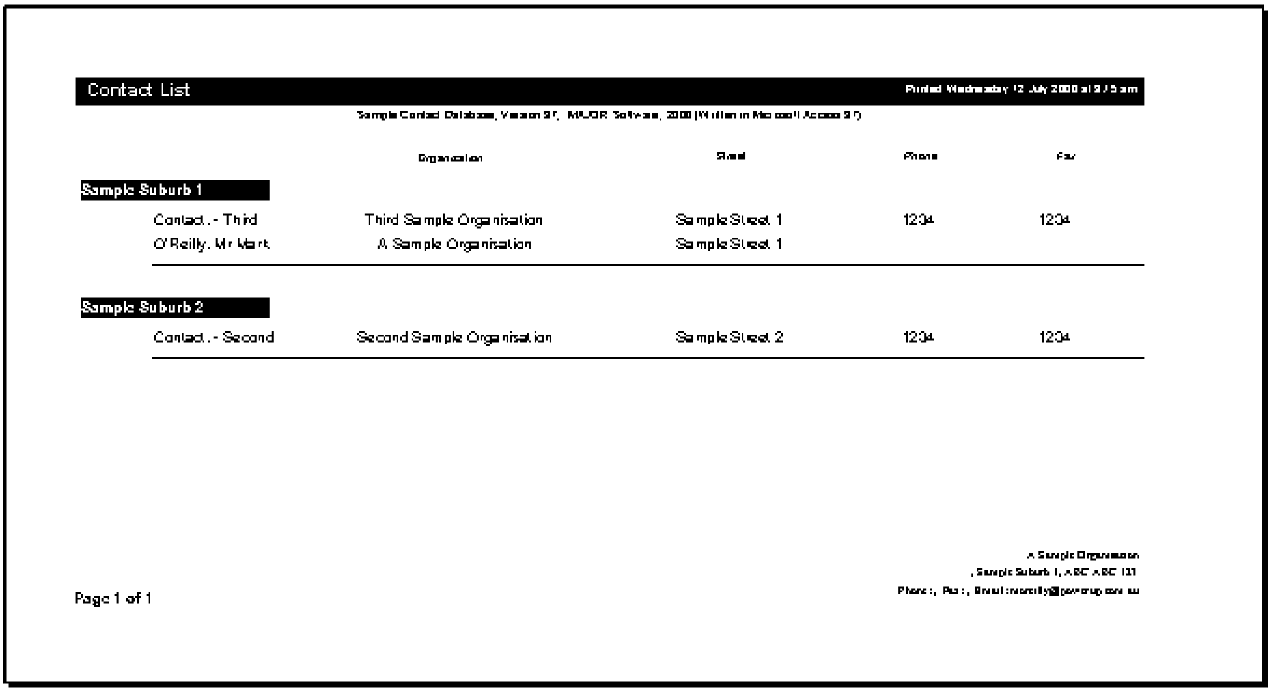
The layout of the data itself also brings with it a number of considerations:

- The output may be a simple list. You might consider that fields be contained within a table, or that individual records have a line separating them. Such lines and tables can enhance readability, and yet can be used subtly.

- The output may need to be grouped under one or more headings. The headings can be made to stand out by using varying fonts, bolding, banners, dividing lines and/or indentation. As the number of levels of headings increases so too does the need for careful design.

- Some reports may have designs other than tabular. Such reports may include letters, memoranda, mailing lists and invoice-style reports. The planning of the headers and footers may be quite different for these reports:

    - ◊ A letter or memo might have client letterheads pre-printed on the paper, and so no header and/or footer may be required
    - ◊ Mailing lists may be directly printed on labels, and so again headers and footers may not be required.
    - ◊ Invoices, quotes and other financial or client statements may also be output to pre-printed forms. The data lines, as well as the headers and footers, would need to line up and be printed in the correct locations.



Contact List

Printed Wednesday 12 July 2000 at [illegible] am

Sample Contact Database, Version [illegible], MAJOR Software, 2000 (Written in Microsoft Access [illegible])

	Organisation	Street	Phone	Fax
Sample Suburb 1				
Contact, - Third	Third Sample Organisation	Sample Street 1	1234	1234
O'Reilly, Mr Mark	A Sample Organisation	Sample Street 1		
Sample Suburb 2				
Contact, - Second	Second Sample Organisation	Sample Street 2	1234	1234

Page 1 of 1

A Sample Organisation
, Sample Suburb 1, ABC ABC 123
Phone: , Fax: , Email: moreilly@[illegible]

As with the screens, you are able to use the ‘Interface Style Worksheet’ and the ‘Screen and Reports Worksheet’ (Appendix A) to plan the various aspects of your application reports. You can use the former to plan the report template(s) and the later to identify the report specific information. An example of the later is given here:

Item	**Contact List Report (grouped by Suburb)**
Objective	To allow the business to manage the details of all business contacts - customers, suppliers and employees.
Task	To print a full listing of contacts, grouped by suburb
Window Title	Contact List Report
Report Title	Contacts
Report Name	repContactListR
Sub-Screen Names	N/A

Underlying Tables and Queries (possibly SQL)	**ContactsMF**:
	SELECT DISTINCTROW Contacts.ContactID, [LastName] & ", " & [Title] & " " & [FirstName] AS Name, Contacts.OrganizationName, Contacts.Street, SuburbList.SuburbID, SuburbList.Suburb, Contacts.PhoneNumber, Contacts.FaxNumber
	FROM Titles INNER JOIN (SuburbList INNER JOIN (Contacts INNER JOIN Categories ON Contacts.ContactID = Categories.ContactID) ON SuburbList.SuburbID = Contacts.SuburbID) ON Titles.TitleID = Contacts.TitleID
	WHERE (((SuburbList.SuburbID) Like IIf([Forms]![ContactFilterF]![cboSelectSuburb]<>1,[Forms]![ContactFilterF]![cboSelectSuburb],"*")) AND ((Categories.CategoryID) Like IIf([Forms]![ContactFilterF]![cboSelectCategory]<>0,[Forms]![ContactFilterF]![cboSelectCategory],"*")));

Field Descriptions

Field Descriptor	**Mandi tory**	**Justify (L/C/R)**	**Field Type**	**Size**	**Conditions / Validation / Format**
Suburb	Y	L	Text	6	Group Heading 1
First	Y	L	Text	20	
Last	Y	L	Text	20	
Business Name	N	L	Text	50	
Phone	N	L	Text	20	Format: (00) 0000 0000
etc…					

Automation

Object	**Event Trigger / Source**	**Function**
Report	On-Open	To check that there are contacts to be reported. If there are no contacts, do not display or print the report
ContactsFilterForm - OK button	Click	It would be useful to use this table to identify any Filter screens linked to the report…
etc…		

Naming Conventions

Different programmers develop different naming conventions for their screens, queries, reports, combo boxes etc. You will realise that such conventions become critical as the applications you develop become larger and larger. For instance, you need some way to delineate tables from queries with similar names in a list, be able to delineate a control name from a field name, or find a combo box control easily in a control list. Tables and queries underlying screens and reports should have matching names so they can be linked by inspection.

A range of naming conventions have been published world wide and reprinted in various developer publications. For instance, Microsoft Visual Basic and Access developers make reference to such naming conventions as the ‘Hungarian’ created by Charles Simonyi, the Leszynski and Reddick (L/R) convention, the Leszynski Naming Convention (LNC), and the Reddick VBA Naming Convention. Details of these can be found in some of the references listed at the end of this text.

It is suggested that you research naming conventions relevant to your development platform and either use one as presented, or adapt one as required.

Screen / Report Worksheet

In Appendix A is a worksheet titled: **4. Screen and Reports Worksheet.** You would fill in one copy of this worksheet for each screen and report within your application. Sample data for one of these worksheets has been presented earlier in this section.

5. Review the Plan

Once the general plan has been created it needs to be reviewed and cross checked against the original specification. Please make the effort at this point to **stop** and take the time to reread and review your work:

- Reread the original specification carefully.

- Reread your objective(s) and tasks and convince yourself again that they match the specification.

- Check your initial draft and convince yourself again that it fully matches the specification in all aspects.

- Review your interface style - you may now reflect differently on your initial design, and wish to adjust it.

- Carefully review each screen and report identified in Step 4 - Screens and Reports. Check that each links to one or more of your tasks and that as a set, fulfil both the requirements of the original specification and the objective(s).

- Imagine yourself completing tasks in each of the processes in your Data Flow Diagrams and convince yourself that your screens and reports are optimised for the tasks required of them:

    - ◊ Check that the data entry screens can capture all data to be collected.
    - ◊ Check that the standalone menus and various screen buttons allow for optimum passage through the application.
    - ◊ Check that the reports supply the information - filtered, grouped, sorted and/or summarised - as specified.

- Check that all screens and reports have a consistent design and appear to be part of one homogenous application.

- Check each screen and report separately and convince yourself that each one reflects its inherent function and identity. (Eg: does the client details screen look and feel like a client details screen?)

- As you work through the review, tick or date each of your planning worksheets, in the space provided (titled ‘Reviewed:’).

6. Database Prototype

There are quite a number of graphical user database applications currently available that could be used for creating a prototype of your application. The Contacts database that comes with this booklet might be considered a prototype. It would be viewed and reviewed by the client, and the client’s feedback would be used to shape and polish the final product.

A prototype may present facades of the various database menus, data entry, filter and lists screens, and possibly sample reports. The prototype components may lift data from functional tables, or might have sample data presented in unbound fields or labels. It may be useful for buttons on menus and other screens to react to user input, allowing your client to familiarise themselves with the flow of the application.

For some small to medium database applications, the prototype can be created in the same Relational Database Management System (RDBMS) as that used to develop the final product, thus saving significantly on development time.

7. Analysis and Design Review

With the completion of your plan, its review and your prototype, you are ready to formally walkthrough the analysis and design with your client.

The creation of reports, the making of presentations and the management of walkthroughs is well covered by many Systems Analysis texts. To touch on a couple of points:

- You need to be clear on why you are meeting and what you are needing to walk through. It can often be an advantage to check with the client as to what they are expecting to see, hear and discuss.

- Presentation of a formal design document may be appropriate and may have been specified. However, for small and/or time critical developments, rough design notes, a brief overview document and/or a prototype might be all that is required. Check with your client prior to commencing the analysis and design.

- You may be nervous prior to the walkthrough, but this may pass as you focus on the content of your analysis and design. Focussing client attention on the prototype, various design documents and notes, and/or some presentation slides can also help.

- The larger the development and the larger the pool of people involved, the more formal you will need to be in your management of the walkthrough. You will need to stick closely to the agenda, carefully manage interjections and questions, and present all material logically and sequentially.

- You must assume that there will be parts of the analysis and design that the client will want to change. Although you have poured your heart and soul into the project and want to be protective of it, you must be able to let go… - the client has to live with the application long after you have developed it. In a way, your client is your mentor - take on what he/she says and accommodate his/her perspectives.

Functional Database Design Activities

On the pages that follow are a number of examples and activities to reinforce the concept of Functional Database Design. Two case studies with full solutions are provided for review, discussion and debating purposes:

- Brenda’s Basketball Courts
- Vincent’s Video Rentals

Two further case studies are provided for the purposes of reinforcing the skills discussed in this booklet:

- Erin’s Electronics Engineering
- Chris’ Chess Competition

Notes

Please note that the case studies are provided as examples only. The scripts are purely hypothetical and are scoped (down) for the purposes of illustrating Functional Database Design. They are not likely to accurately reflect the necessary management requirements of their respective sport, hobby or business.

A sample approach to the development of Data Flow Diagrams, components of a System or Data Dictionary, Entity Relationship Diagrams and Normalisation has been provided in a set of additional notes titled: Brenda’s Basketball Courts - a Systems Analysis Example. The scope of these additional notes however, along with the resulting design, is only a subset of that presented in this document.

A Suggested Activity

Each of the case studies presented on the following pages is based on a script that depicts a hypothetical client’s specification. An alternate activity might be to view an existing database example (possibly one supplied with your database development product) and to write a hypothetical script that might reflect the initial client’s specification.

Case Study Example 1

Brenda's Basketball Courts

Brenda manages a set of basketball courts. At the start of the season she receives a draw from the LBCG (Local Basketball Co-ordinating Group) which she places in her top draw for review purposes.

Brenda must ring the various team managers (who are responsible for their individual teams) to verify games the week before they are played. She tells each manager the date and court of their next game. The manager indicates the team's intent to play.

The manager faxes a list of players to Brenda so she can organise the scoring sheets in advance. She stores the faxes and the scoring sheets in her filing cabinet.

The manager verifies the player list when the team arrives to play. The fouls and goals are recorded during the game. After the games, Brenda generates a set of summaries of the individual score cards. Brenda stores the score cards and the summaries in her filing cabinet.

Brenda sends copies of the score card summaries to the LBCG and to the respective team managers.

Brenda has asked your development team to develop a database application that will allow her to manage all the above activities.

Data Storage Requirements

In relation to the storage of data, Brenda indicated that she would require:

- The database to manage multiple draws, including weeks and games within each draw.
- Complete lists of players for each team, and shorter lists identifying players competing in each game. 'Usual' or 'standard' player positions should also be recorded.
- Goals and fouls to be recorded against participating players.
- Team and current manager details to be recorded, including manager name, mailing address, phone, fax and email.

Data Entry Requirements

In relation to the entry of data into the database, Brenda indicated that she needs to:

- Add the season draw, including weeks, teams, and games, at the start of each season.
- Add all players and team managers prior to the first game.
- Add or edit player and manager details during the season, as required.
- Create a new player list for each team each week. The ability to copy all players to a new game list (and then delete those players who are not participating) would be beneficial.
- Adjust final positions and participating player lists quickly, prior to the games.
- Record fouls and goals against the participating players as the games proceed. The score-keepers should be able to enter the game time against each such event.

On-Screen Review Requirements

In relation to the information to be displayed on screen, Brenda suggested that she would utilise:

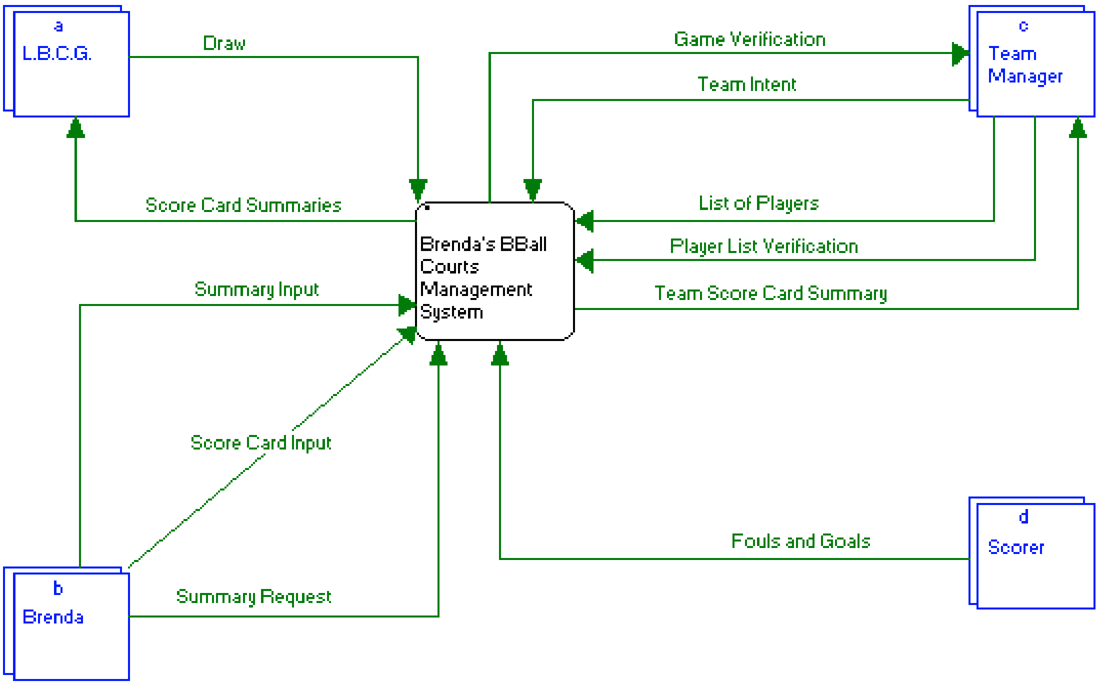
- A screen that displays all participating players, their cumulative goal and foul tallies as games proceed.

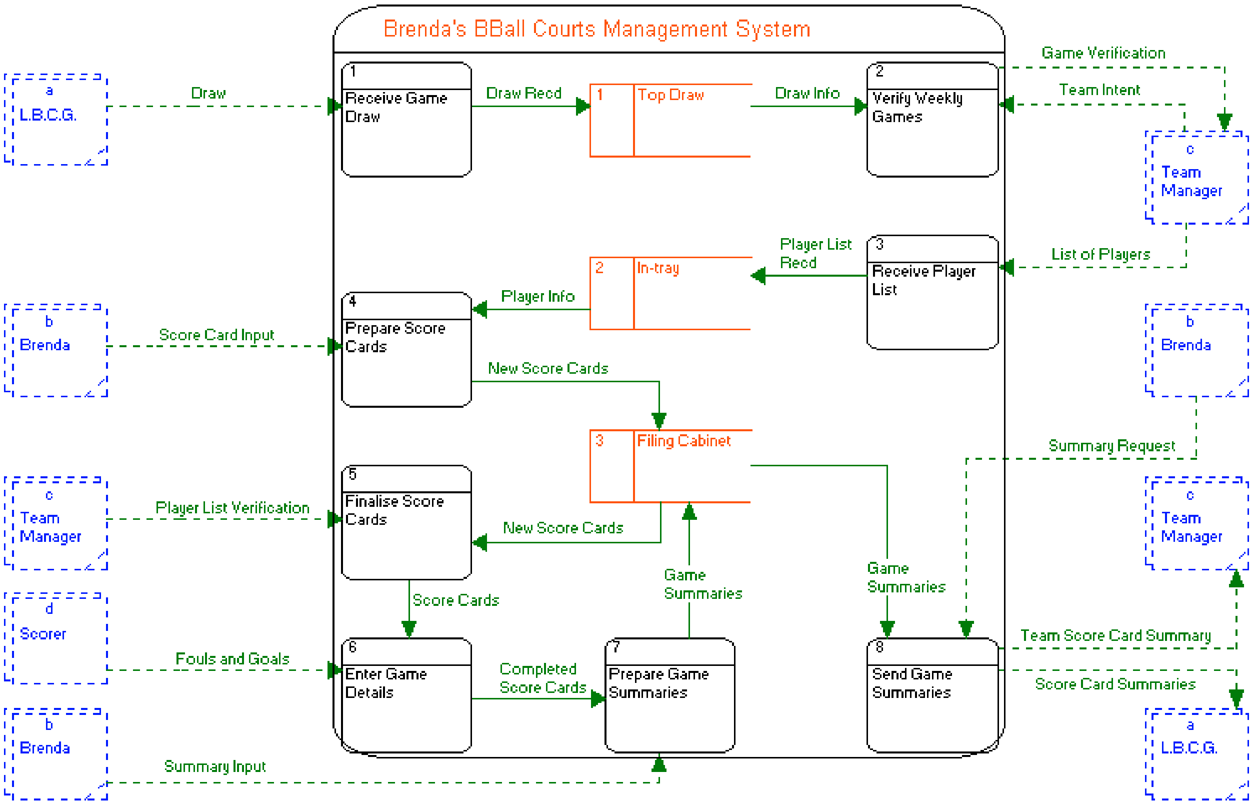
Reporting Requirements

In relation to printed output, Brenda identified the following database reports:

- A copy of the game draw, grouped by week and by game within that.
- Standard team details. The user should be able to print all teams or a particular team. The detail presented on the report should include: team details, current manager details and the standard player list, including their usual position numbers.
- A weekly results summary, grouped by game, and participating team within that. The report should include all team details: team name, manager, players, fouls and goals. The fouls and goals should be tallied by player and by team, and the winning team should be clearly identified.
- A player statistics report listing tallies of fouls and goals per game for the season, grouped and sorted alphabetically by player name.

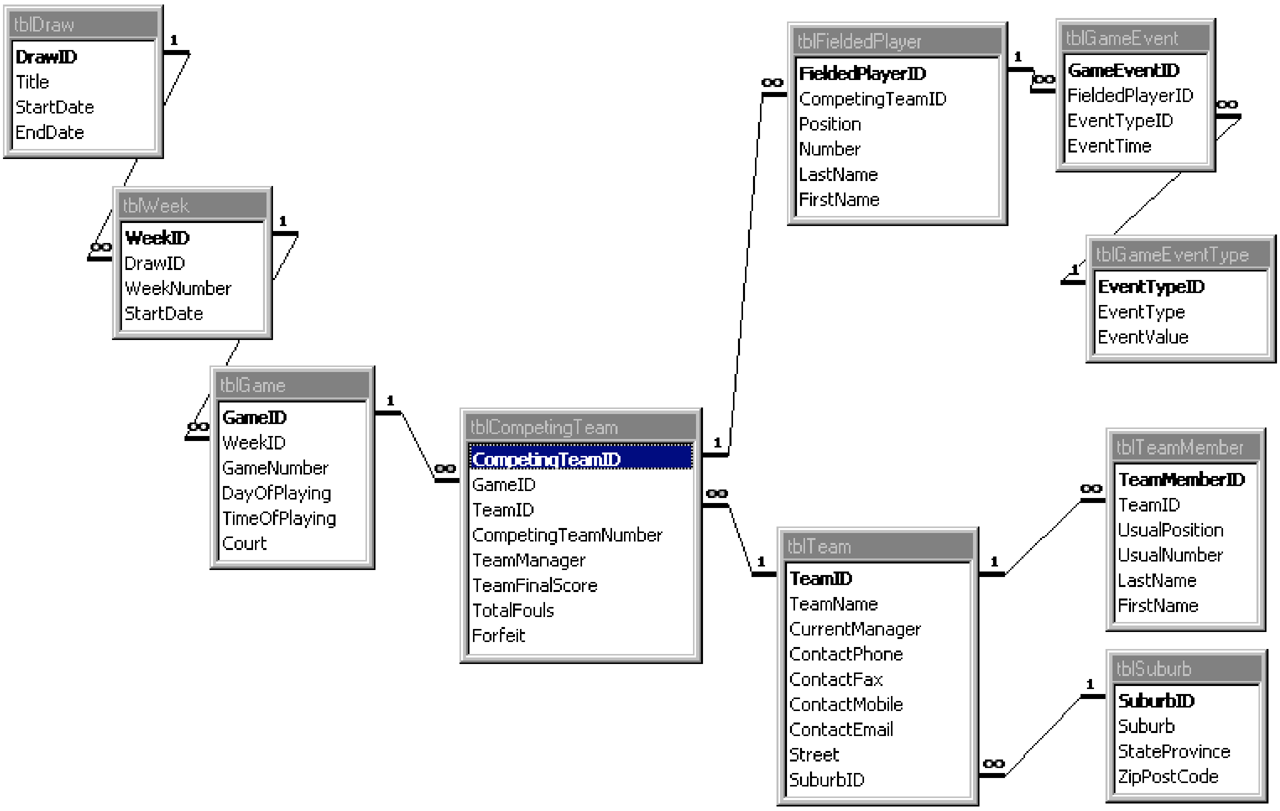
Data Flow Diagrams





Data Flow Diagrams prepared using the Ascent CASE Tool. (Visit: www.knowledgebase.com.au)

Database Tables



Discussion

Objectives and Tasks

As part of the broader systems analysis and design for Brenda's Basketball Courts a number of objectives, tasks and milestones may have been identified and detailed in a project management chart. Within this larger process however, it may have been decided that the development of a database system is necessary. Such a system would have to manage the information associated with the various draws, the various teams and for each 'game in progress'.

This offers us an **objective** for the functional design of this database:

To manage draw, team and game details for Brenda's Basketball Courts.

Reviewing the Level 0 Data Flow Diagram in conjunction with the table design provides insight to the tasks that this database system must complete. Taking each of the processes in the Level 0 DFD:

Receive Game Draw: When the draw is received by Brenda, she or a delegated officer must enter the details of the draw into the computer system. A Data Flow line indicates that the draw is received from the LBCG while a second data flow line indicates that the draw is transferred to the Top Desk Draw. In introducing a computer system, a copy of the draw would also need to be stored within the database system. The task linking the computer operator to the database might be: Add, edit and view all draw details.

Verify Weekly Games: In this process an officer is accessing the draw details and ringing the various team managers for verification of their intent to play. In relation to the computer system, the operator must be able to view the draw details, and possibly edit the game details to reflect any forfeits. Therefore the task associated with the functional database design for this process might read: View the draw, and edit competing team details.

Receive Player List: In this process the operator would need to add team details to the database. We must keep in mind that Brenda indicated that she wished to manage a set of *'standard'* team details, and copy or link these team and player details for each individual game. Therefore the operator in this process would need to complete two tasks: 1. Update any changes to the *'standard'* team details (which would have been entered prior to the first game of the season), and… 2. Transfer or link the *'standard'* team details for the next game of the season.

Prepare and Finalise Score Cards: In moving to a computerise scoring system, the operator would merely have to finalise the team and player detail for this week's games in accordance with information provided by the various team managers. To qualify one of the tasks presented above, the operator would need to: 1. Add, edit and view *'standard'* team details, and… 2. Copy or link *'standard'* team and player details for the current game, and provide the ability to edit or delete such details.

Enter Game Details: In this process the operator would need to record the various goals and fouls as the game proceeds or at any time after the game. The opportunity presents itself to record such game events as games proceed, though the scorer(s) would need access to PCs or laptops court side, and would need to be able to add the events quickly and efficiently so as to keep up with the game play. The task associated with this process then would be: Efficiently add and edit game events.

Brenda has also specifically asked that the game progress be presented on screen on request at any point throughout a game. This requests adds a further task: To view game statistics as the game proceeds.

Prepare and Send Game Summaries: This process requires the printing of a game summary report. The task might read: Print Weekly Results Summary report.

A number of reports come out of these processes or have been specifically requested by Brenda. These are:

1. A detailed copy of the game draw
2. Standard team details
3. Weekly results summary
4. Player statistics

This discussion has brought a number of tasks to light. These have been listed on a sample **Objectives and Tasks Worksheet**, presented after the case study discussion.

The Initial Draft

Based on the considerations of the previous section and the initial table design, a rough map of the database application can be drafted. A sample draft has been provided on the **Initial Draft Worksheet** included with this case study.

Some considerations relating to this preliminary design might include:

The operator has to add, edit and view the draw details. The draw details are made up of draw details, weeks, games and competing teams, and these are linked one to many in a cascading manner: one draw, many weeks; one week, many games, one game, two teams.

One to many relationships can often be well represented on main-sub forms, or main-detail screens. The *one* side of the relationship is presented at the top of the screen, while the *many* side is presented in a detail or sub section of the screen, possibly as a list.

This type of screen could be used effectively in this instance. One screen could be used to add, edit and view a draw, along with a list of the associated weeks. Another form could be used to present a specific week and allow the operator to add, edit or view the respective games. Similarly with an individual game and the two competing teams.

The linking or the navigation between these screens is the next decision. It is possible to build the three main-detail screens and provide separate access to each from a main menu. The difficulty presented in this instance, is that the draw and week will have to be reselected on the subsequent screens before games and teams can be added or edited.

Alternatively the main menu might offer access to the Draw-Weeks screen. Once a specific week has been selected on this screen, the Week-Games screen could be accessed, automatically displaying the selected week. Similarly, once a game had been selected the associated competing teams could be accessed.

The Detailed Draw report might be accessed from the Draw-Weeks screen. In this way, the user would open this screen from the main menu, select the required draw from a *Find* combo box, and print its detail without further selection.

To add, edit and view 'standard' team details, one (possibly two) screen(s) need to be provided to access both the team and the associated player details. Again the one team-many players relationship suggests a single main-detail screen. And since the detail associated with each player is restricted to a small number of fields, the player details could be comfortably addressed in a list-style sub or detail section within the screen.

The *'Standard'* Team Details report could be linked in a natural way to this *'standard'* team data entry screen.

To copy the *'standard'* team details for each competing team, it is possible to add automation and one or more screens to the end of the cascading screens already suggested for managing the entry of Draw details. However, we already have quite a number of screens in this cascade: main menu - draw screen - week screen - game screen. To add further screens to this current cascade might not be the most optimal implementation, especially as the process of transferring and updating the competing team players occurs at a distinctly different time to that of the entry of the Draw details.

An alternative might be to create a new screen which allows efficient selection and reselection of the required draw, week and game. This could be done using a cascade of combo boxes, list boxes, and/or sub or detail screens (or any combination thereof). We can assume that the competing teams and their player details may require significant space within the screen, and so these navigation components might need to be optimised for space. Since the operator would more often need to flip between the individual games for a given week, the selection of draw and week could be provided within combo boxes, while the selection of games could be provided within a list box or detail screen.

It might also be possible for this new screen to manage (or launch additional screens to manage) the **efficient addition and editing of game events**. If this were to be considered, we would need to optimise both the navigation and the screen real estate. Space would be needed for the navigation components, for the competing team details, and for each of the players lists. The game events could also be housed and managed within this screen, though an additional popup screen might work equally well.

The Game Progress screen, the Weekly Results Summary report and the Player Statistics report might all be launched via buttons on this new competing team / game progress management screen. Since the draw, week, and game would already be selected, these further components could be linked as required.

The **Initial Draft Worksheet** (included with this case study) offers a view of how these consideration might be presented within a set of initial rough sketches.

The Interface Style

This database is for management purposes within a set of basket ball courts. Therefore the screen design could incorporate graphics such as a basketball, a court, a basketball net, basketball players, or any combination thereof. Use of a 'light', consistent, and efficient interface might be appropriate:

- Possibly a bright, cheery and/or 'sporty' interface, not detracting from the functionality and purpose of various screens and reports.
- Consistent use of graphics, colours, buttons and other controls.
- Consistent heading and footer styles on both screens and reports.
- A quality splash or menu screen that clearly links the product with the management of a set of basketball courts.

In illustrating some of these points, a **sample solution** has been developed, and might be accessed at this time to enhance further discussion.

For instance, **the sample solution** makes use of a basketball graphic, white backgrounds with lots of 'white' space. The primary graphic is relatively broad and long leaving substantial gaps on the bottom left of each screen. It was decided to use these gaps for the placement of screen buttons. This breaks with Microsoft Windows convention - especially the placement of the Close buttons, though this placement might be considered in keeping (somewhat) with the design of a web page.

The sample solution also makes consistent use of **a 'double clicking' feature**. On a number of screens, pale yellow has been placed under some text boxes. Double clicking on these text boxes with the mouse will launch and link an associated screen. For instance, on the Draw and Weeks screen, the user can double click on a Week No. to launch the Week and Games screen. The selected week will be the focus of the new screen.

In relation to the reports, the same basketball graphic was utilised with a consistent header and footer style built around it. In line with appropriate report development, each report presents a heading, a date and time of printing, the name and address of the user, the name of the database system and the page numbering.

In terms of the information presented on the various reports:

- Most required a heading that drew a data element from the underlying screen or filter form. For instance, the Draw report presented the title of the Draw within the report heading, the Team Detail report presented the selected team within the report heading.
- Most reports require one or more groupings. On three of the sample reports, the primary group is highlighted using a grey coloured band across the width of the report. For instance, the Draw report highlights the week grouping, the Weekly Summary report highlights the Game.
- The clarity of presentation of sub-groups is enhanced with the combined use of varied font sizes, indentation and horizontal lines.

The **Interface Style Worksheets** (included with this case study) offer a view of how these consideration might be developed as a set of initial rough sketches.

Screens and Reports

This stage of the design requires more direct focus on the individual screens and reports of the database system. If the development requires detailed documentation then the **Screens and Reports Worksheet** provides a structured framework within which the design and associated documentation can be considered.

Where extensive documentation has not been requested and/or has not been funded, the Screens and Reports Worksheet can provide a guide or checklist against which the various components of the interface are to be considered.

In illustrating the use of this worksheet, one screen and one report from the sample solution has been presented within this case study.

The Prototype and Review

The sample solution is somewhat more than a prototype, although it falls significantly short of a completed database system. Prototypes often need to be functional to the point that the client can **navigate** between the various screens, **visualise** screen purpose and function (including any inherent complex automation), and **view** a number of the reports. It is often valuable to spend some time on ‘finishing’ or tidying up a number of the screens and reports (most notably the opening screens) so that the quality of the final product can be visualised clearly by the client. When a client is not distracted by unfinished or untidy screens and reports, they are able to focus on and provide more direct and constructive feedback on the functionality and output of their system.

The sample solution is provided as a Microsoft Access database - Brendas.mdb.

Objectives and Tasks

1. Objectives and Tasks Worksheet

Print (in pencil) one objective on a fresh copy of this page, and list all associated tasks. This page is to be used as a cover sheet for all pages of functional design associated with this objective.

Project: Brenda's Basketball Courts

Completed By: ___________________________

Date: _______________________

Objective:

To manage draw, team and game details for Brenda's Basketball Courts.

Reviewed: ________________

Associated Tasks:

1. Add, edit and view all draw details.
2. Add, edit and view 'standard' team details.
3. Copy or link 'standard' team and player details for the current game, and provide the ability to edit or delete such details.
4. Efficiently add and edit game events.
5. To view game statistics as the game proceeds.

6. Print a detailed copy of the game draw.
7. Print 'standard' team details.
8. Print a Weekly Results Summary report.
9. Print a Player Statistics report.

Reviewed: ________________

The Initial Draft

2. Initial Draft Worksheet

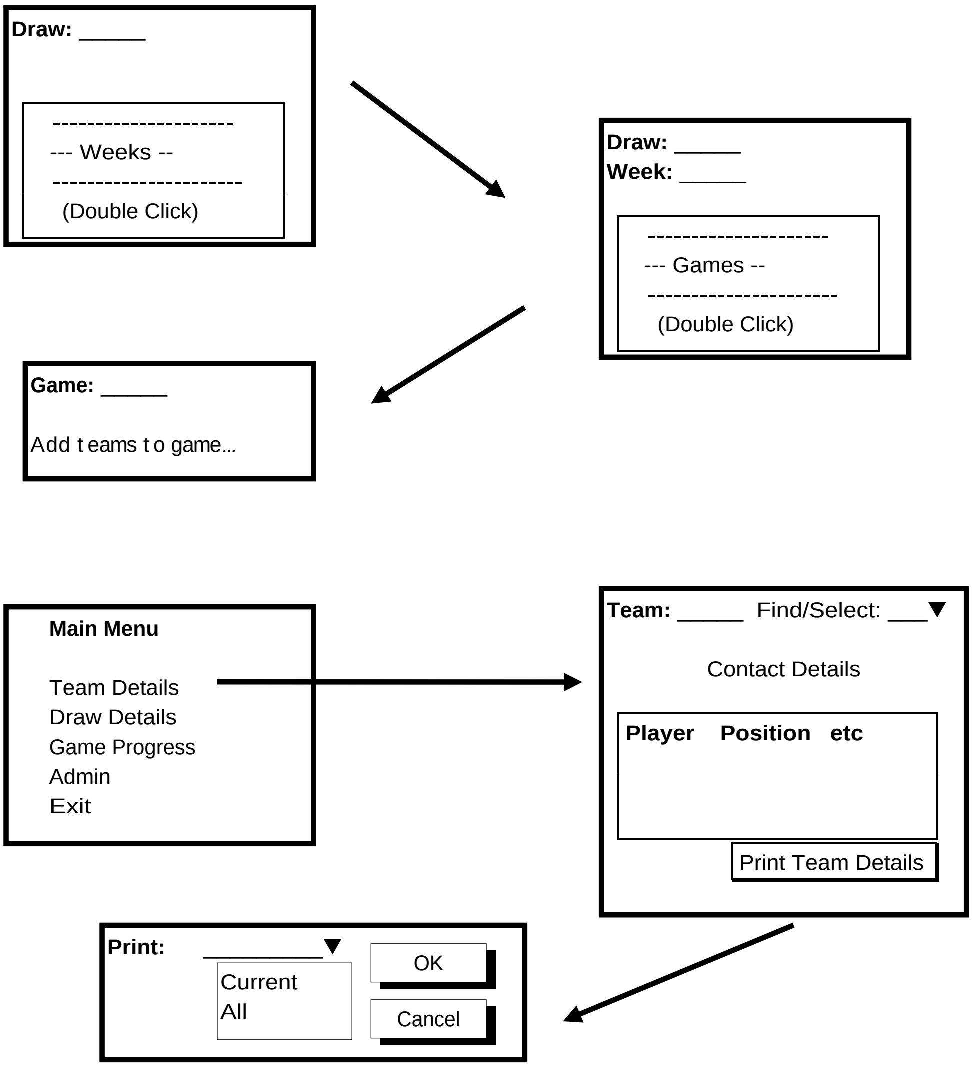
On additional paper, draw (in pencil) rough sketches of, or note down in words what might be on each screen and report. On this page, draw a rough sketch of the map that shows the links between the various screens and reports.

Project: Brenda's Basketball Courts

Completed By: ___________________________

Date: _______________________

Screen / Report Map:



Reviewed: ________________

Note: *Further rough sketches would be developed on paper for 'Managing the Game Progress' screens.*

The Interface Style

3. Interface Style Worksheet

Draw (in pencil) rough sketches of screen layout/styles, as well as report headers and footers.

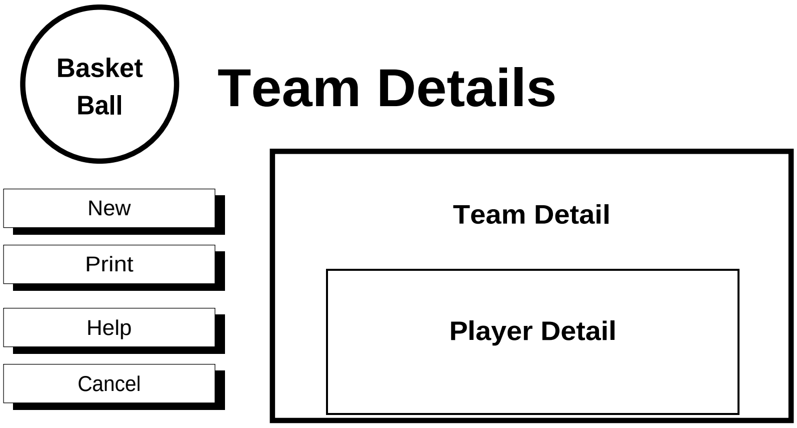
Project: Brenda's Basketball Courts

Completed By: ___________________________

Date: _______________________

Screen/Report Layout(s):

The screens:



Reviewed: _________________

Screen/Report Header:

Logo and the heading

Reviewed: _________________

Screen/Report Footer:

Buttons down the left hand side.

Reviewed: _________________

3. *Interface Style Worksheet*

Draw (in pencil) rough sketches of screen layout/styles, as well as report headers and footers.

Project: Brenda's Basketball Courts

Completed By: ____________________________

Date: ________________________

Screen/Report Layout(s):

Main Heading 1

Sub Heading 1

Report Detail Line ----------------------------
Report Detail Line ----------------------------
Report Detail Line ----------------------------

Sub Heading 2

Report Detail Line ----------------------------
Report Detail Line ----------------------------

Reviewed: ________________

Screen/Report Header:

Basket Ball

Draw: 2002 Season 1

Brenda's BBCourts
Address and Comms
Date and Time Printed

Reviewed: ________________

Screen/Report Footer:

Brenda's BBCourts Database Page...

Reviewed: ________________

4. Screens and Reports Worksheet

On additional paper, draw sketches of screen and report layouts / styles. Fill in the details of the screens or reports on separate copies of this worksheet. Only fill in details relevant to each type of screen or report.

Project: Brenda's Basketball Courts

Completed By: ___________________________

Date: _______________________

Objective	To manage draw, team and game details for Brenda's Basketball Courts.
Task	Add, edit and view 'standard' team details.
Window Title	Teams - Brendas Basketball Courts
Screen/Report Title	Team Details
Screen/Report Name	frmTeamMF
Sub-Screen/Report Names	frmTeamSF
Underlying Tables and Queries (possibly SQL)	frmTeamMF: All fields from the tblTeam and tblSuburbs tables frmTeamSF: All fields from the tblTeamMembers table Reviewed: ________________

Field Descriptions

Field Descriptor	Mandit-ory	Justify (L/C/R)	Field Type	Size	Conditions / Validation / Format / Other
Team Name	Yes	Left	Text	50	
Current Manager	Yes	Left	Text	30	
Phone	No	Left	Text	15	
Fax	No	Left	Text	15	
Mobile	No	Left	Text	15	
Email	No	Left	Text	100	
Street	No	Left	Text	50	
Suburb	No	Left	Text	20	
State/Province	No	Left	Text	4	
Postal Code	No	Left	Text	8	
Player Last Name	Yes	Left	Text	20	
First Name	Yes	Left	Text	20	
Usual Number	No	Centre	Number	Int	
Usual Position	No	Left	Text	20	
					Reviewed: ________________

Automation

Object	Event Trigger	Function
Print Details Button	Click	Open the Team Details Report Filter
Suburbs Button	Click	Open the Database Admin screen and display the Suburb list
New Button	Click	Clear the screen ready for a new entry
Help Button	Click	Display help details
Close Button	Click	Close the Team Details screen
Find Team Combo Box	After Update	Find the team selected within the combo box, and display their details on screen.
		Reviewed: ________________

4. Screens and Reports Worksheet

On additional paper, draw sketches of screen and report layouts / styles. Fill in the details of the screens or reports on separate copies of this worksheet. Only fill in details relevant to each type of screen or report.

Project: Brenda's Basketball Courts

Completed By: ___________________________

Date: ________________________

Objective	To manage draw, team and game details for Brenda's Basketball Courts.
Task	Print ‘standard’ team details
Window Title	Team Details Report
Screen/Report Title	Team: <Selected Team Name>
Screen/Report Name	rptStandardTeamMR
Sub-Screen/Report Names	rptStandardTeamSR

Underlying Tables and Queries (possibly SQL)	Select all fields from tblSuburb, tblTeam and tblTeamMember for the team or teams selected on the Team Details Report Filter screen.
	Alternatively:
	SELECT tblTeam.TeamID, tblTeam.TeamName, tblTeam.CurrentManager, tblTeam.ContactPhone, tblTeam.ContactFax, tblTeam.ContactMobile, tblTeam.ContactEmail, tblTeam.Street, tblSuburb.Suburb, tblSuburb.StateProvince, tblSuburb.ZipPostCode, tblTeamMember.TeamMemberID, tblTeamMember.UsualPosition, tblTeamMember.UsualNumber, tblTeamMember.LastName, tblTeamMember.FirstName
	FROM (tblSuburb INNER JOIN tblTeam ON tblSuburb.SuburbID = tblTeam.SuburbID) INNER JOIN tblTeamMember ON tblTeam.TeamID = tblTeamMember.TeamID
	WHERE (((tblTeam.TeamID) Like IIf([Forms]![frmReportFilter_TeamDetailsF]![cboSelectTeam]=0,"*",[Forms]![frmReportFilter_TeamDetailsF]![cboSelectTeam])));
	Reviewed: ________________

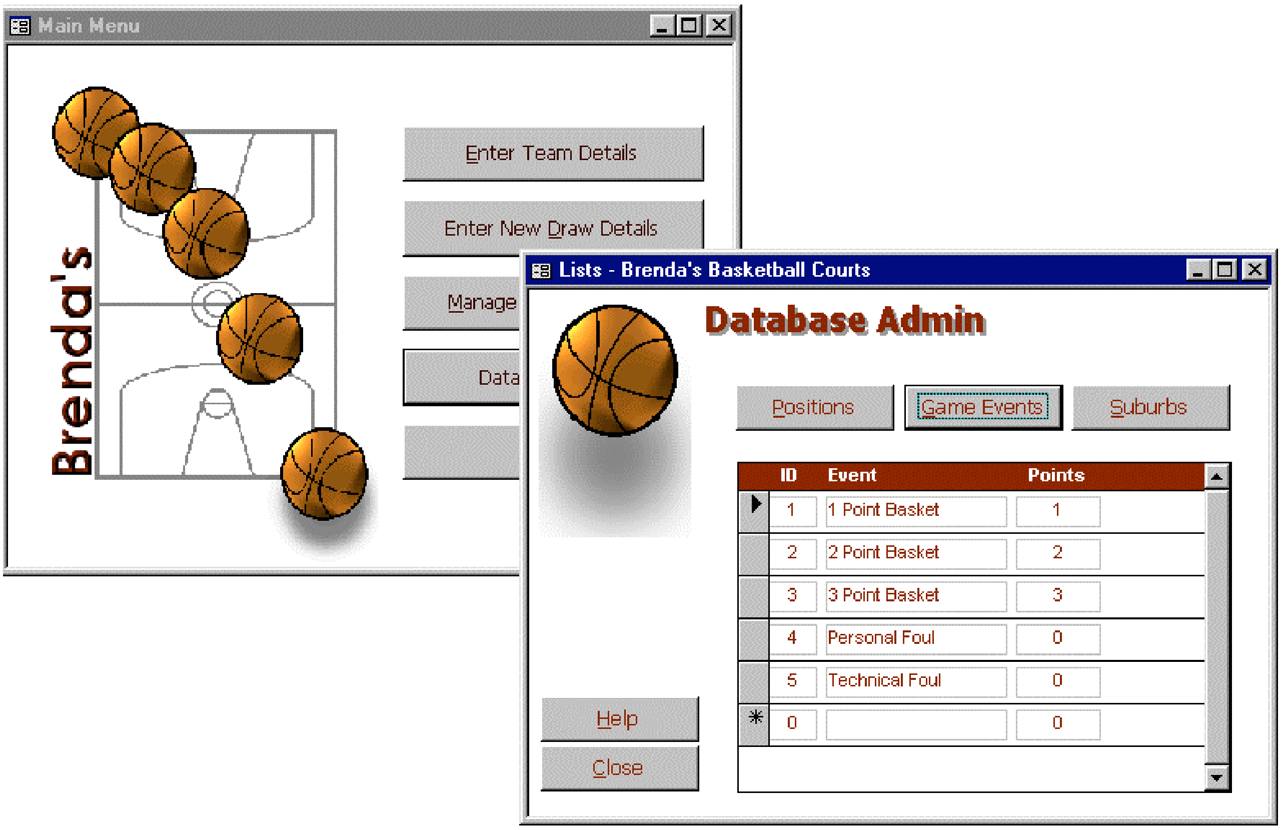
Field Descriptions

Field Descriptor	Mandit -ory	Justify (L/C/R)	Field Type	Size	Conditions / Validation / Format / Other
TeamID	Yes	Left	Number	Long	
Team Name	Yes	Left	Text	50	
Current Manager	Yes	Left	Text	30	
Phone	No	Left	Text	15	
Fax	No	Left	Text	15	
Mobile	No	Left	Text	15	
Email	No	Left	Text	100	
Street	No	Left	Text	50	
Suburb	No	Left	Text	20	
State/Province	No	Left	Text	4	
Postal Code	No	Left	Text	8	
Player Last Name	Yes	Left	Text	20	
First Name	Yes	Left	Text	20	
Usual Number	No	Centre	Number	Int	
Usual Position	No	Left	Text	20	
					Reviewed: ________________

Automation

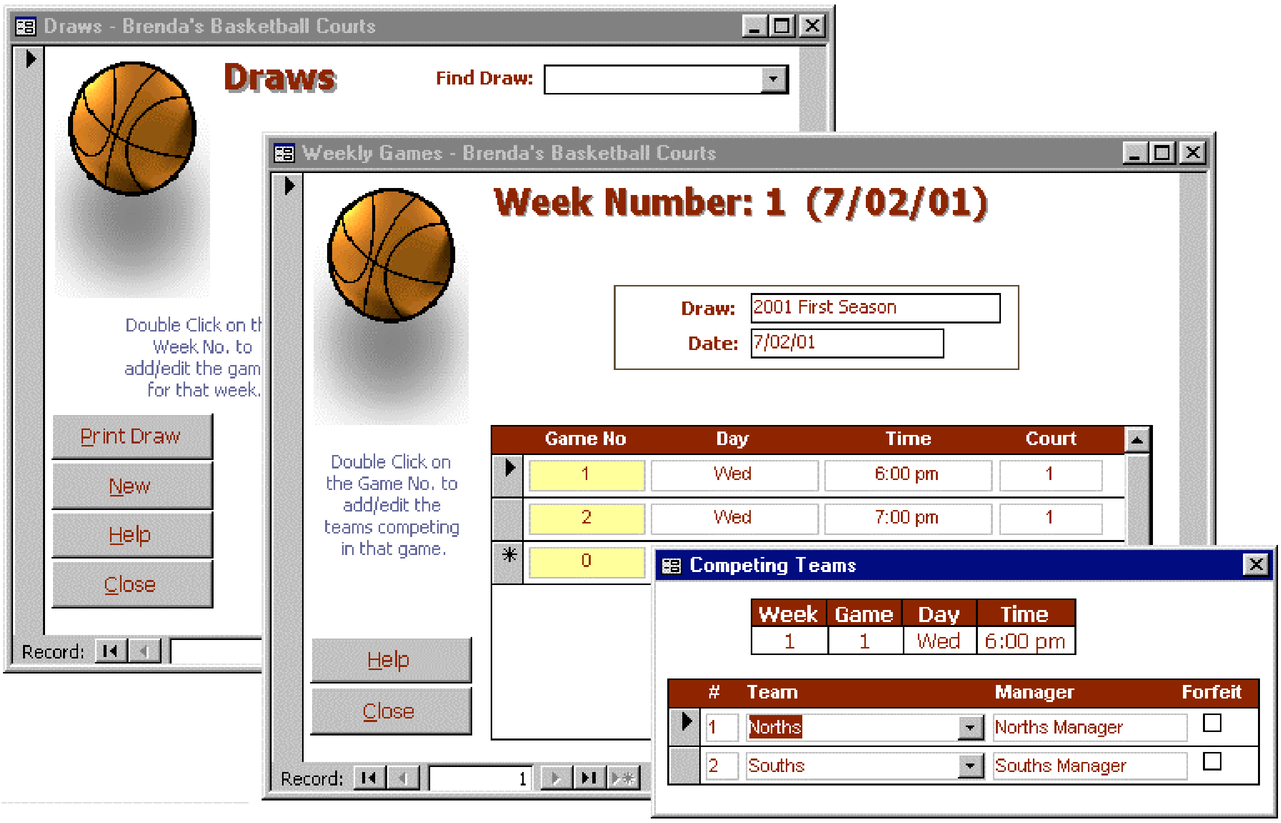
Object	Event Trigger	Function
Report	Open	Maximise the report on the screen
Report	Close	Restore the screen beneath the report to its original size
		Reviewed: ________________

The Prototype



Database Admin

ID	Event	Points
1	1 Point Basket	1
2	2 Point Basket	2
3	3 Point Basket	3
4	Personal Foul	0
5	Technical Foul	0
0		0


Draws - Brenda's Basketball Courts
Print Draw
New
Help
Close
Record:
Weekly Games - Brenda's Basketball Courts
Help
Close
Record: 1
Competing Teams

Draws

Find Draw:

Double Click on th
Week No. to
add/edit the gam
for that week.

Week Number: 1 (7/02/01)

Draw: 2001 First Season

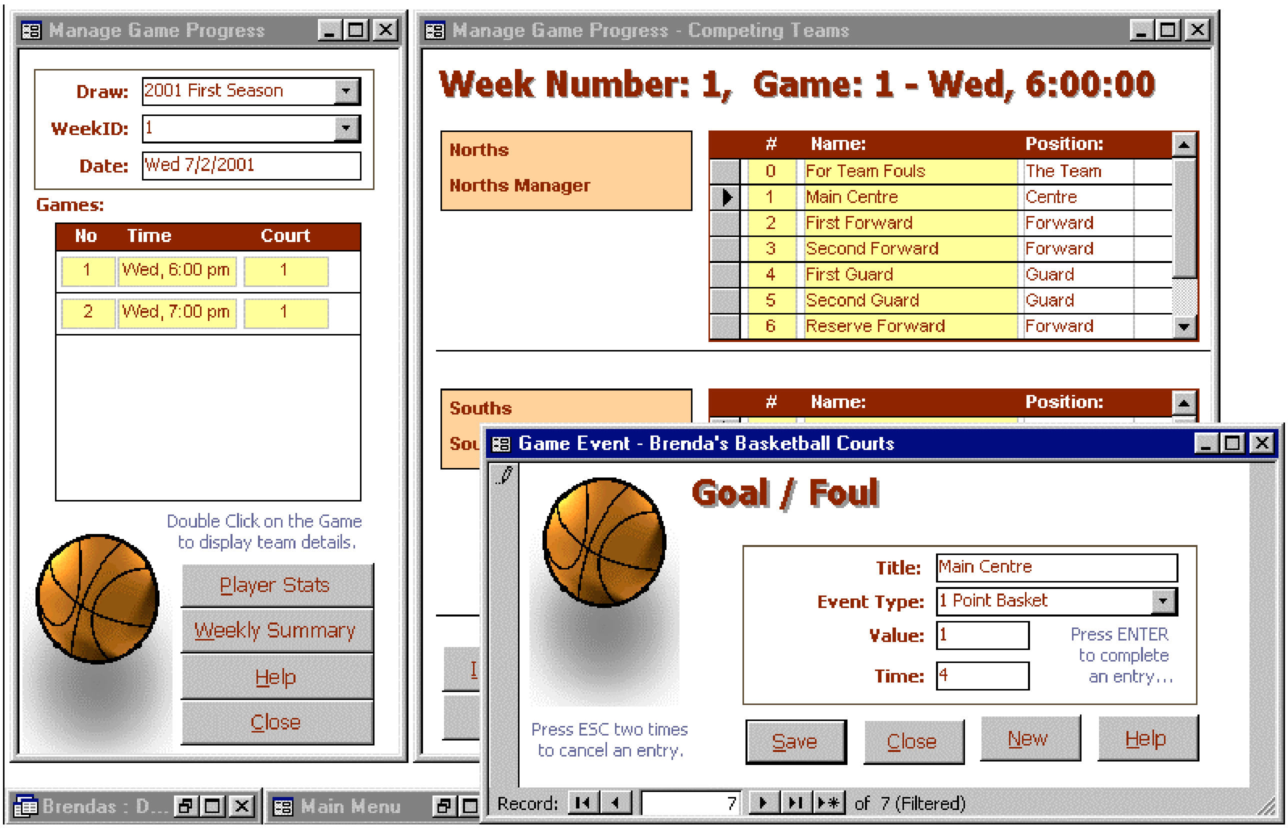
Date: 7/02/01

Game No	Day	Time	Court
1	Wed	6:00 pm	1
2	Wed	7:00 pm	1
0			

Double Click on
the Game No. to
add/edit the
teams competing
in that game.

Week	Game	Day	Time
1	1	Wed	6:00 pm

#	Team	Manager	Forfeit
1	Norths	Norths Manager	□
2	Souths	Souths Manager	□



Manage Game Progress

Draw: 2001 First Season

WeekID: 1

Date: Wed 7/2/2001

Games:

No	Time	Court
1	Wed, 6:00 pm	1
2	Wed, 7:00 pm	1

Double Click on the Game to display team details.

Manage Game Progress - Competing Teams

Week Number: 1, Game: 1 - Wed, 6:00:00

Norths

Norths Manager

#	Name:	Position:
0	For Team Fouls	The Team
1	Main Centre	Centre
2	First Forward	Forward
3	Second Forward	Forward
4	First Guard	Guard
5	Second Guard	Guard
6	Reserve Forward	Forward

Souths

Sou

#	Name:	Position:

Game Event - Brenda's Basketball Courts

Goal / Foul

Title: Main Centre

Event Type: 1 Point Basket

Value: 1

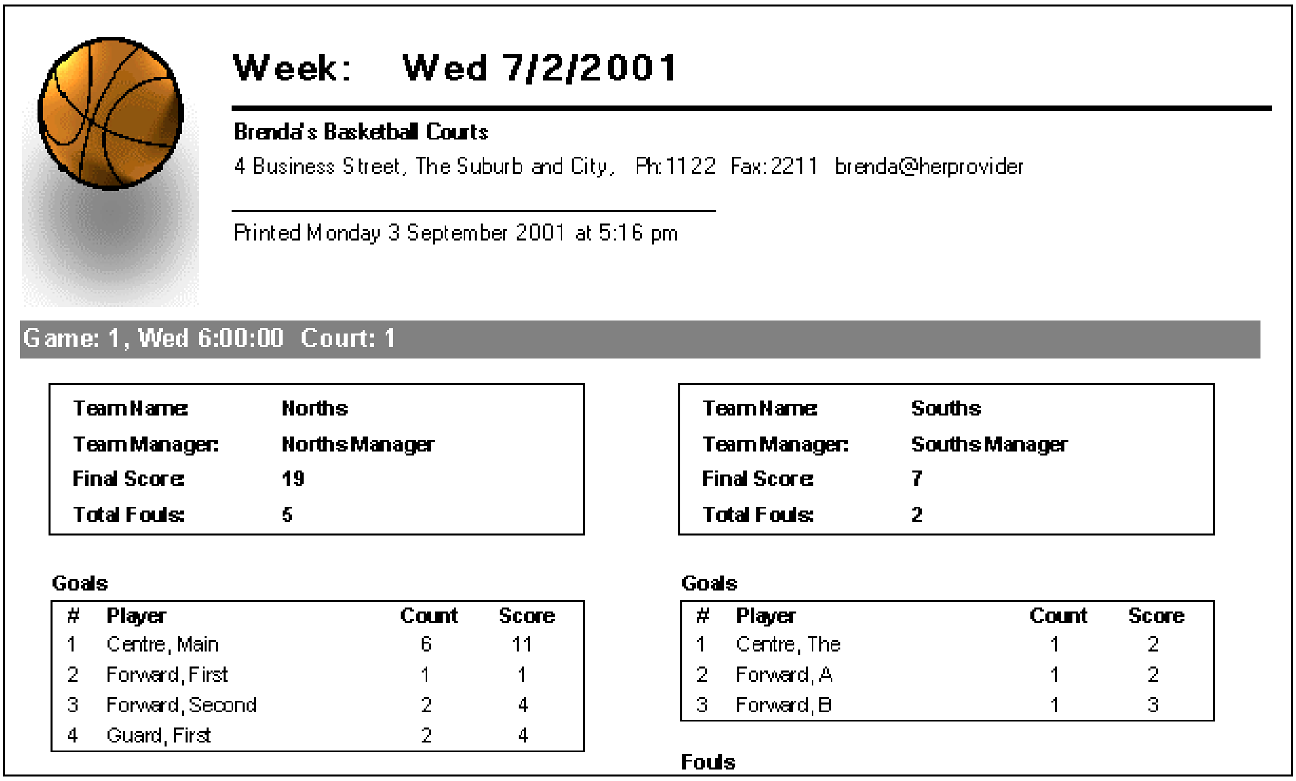
Time: 4

Press ENTER to complete an entry...

Press ESC two times to cancel an entry.

Record: 7 of 7 (Filtered)

Brendas : D...
Main Menu



Week: Wed 7/2/2001

Brenda's Basketball Courts

4 Business Street, The Suburb and City, Ph:1122 Fax:2211 brenda@herprovider

Printed Monday 3 September 2001 at 5:16 pm

Game: 1, Wed 6:00:00 Court: 1

Team Name:	Norths
Team Manager:	Norths Manager
Final Score:	19
Total Fouls:	5

Team Name:	Souths
Team Manager:	Souths Manager
Final Score:	7
Total Fouls:	2

Goals

#	Player	Count	Score
1	Centre, Main	6	11
2	Forward, First	1	1
3	Forward, Second	2	4
4	Guard, First	2	4

Goals

#	Player	Count	Score
1	Centre, The	1	2
2	Forward, A	1	2
3	Forward, B	1	3

Fouls

Case Study Example 2

Vincent's Videos

Vincent is wishing to establish a new section (a mini business) within his current grocery store business - a small video rentals business. He anticipates that he would offer a limited selection of overnight and weekly videos, and would require a single computer to help in the management of the various members, videos and video loans.

When a person wishes to become a member of his video rentals business, a membership application form would be completed - recording home details and the form of identification offered. On completion of this form, the member's details would be recorded within the database, a membership card would be issued, and the new member would be able to borrow one or more videos.

Existing members would come to the loans desk and hand over their membership card and the video(s) they wish to loan. The membership card would contain a member number which would be entered by the attendant at the counter. If the member forgot his/her card, the attendant could ask for the member's name and cross check their membership by asking for their home address or phone number. As part of this membership validation process, the attendant would be able to identify whether the current member has money owing from previously overdue videos. At this point the member would be informed of any outstanding debt.

Once the membership had been established, the attendant would check each video to ensure the video matched that shown on the cover, and that the video rating was appropriate for the borrower. The video number(s) (printed on the side of each video) would be entered into the computer by the attendant. The details of each video would be displayed on screen and the loan would be saved. Once the video loan(s) had been entered, the customer would be told the total amount payable. The customer would pay the required money and the video(s) would be handed to them along with their change and membership card.

Between processing member loans, the attendant would enter the numbers of returned videos into the computer. The attendant would then re-shelve the current batch of returned videos.

The attendant would also be required at times to indicate the availability of one or more videos. Should a video be available, the attendant will mention the video's category, giving the member an approximate location of the video within the shop.

Vincent has asked your development team to develop a database application that will allow him to manage all the above activities.

Data Storage Requirements

In relation to the storage of data, Vincent indicated that he would require the database to manage:

- Member details, including member number, name, address, phone, and identification information.
- Video Details, including video number, title, distributor, rating (general viewing, etc), category (comedy, drama, etc), and rental type (weekly - $2, overnight - $5).
- Video Loans, including member number, video number(s), date of loan, date due, date returned, amount due, amount paid, late fee due, late fee paid.

Data Entry Requirements

In relation to the entry of data into the database, Vincent indicated that he would need to be able to:

- Enter a large number of videos into the database when the business is first established.
- Add new videos and members as required.
- Amend or delete existing video or member data.
- Process video loans and any associated late fees.
- Efficiently process batches of video returns (say up to 20 at one time).

On-Screen Review Requirements

In relation to the information to be displayed on screen when processing video loans, Vincent suggested that he would appreciate:

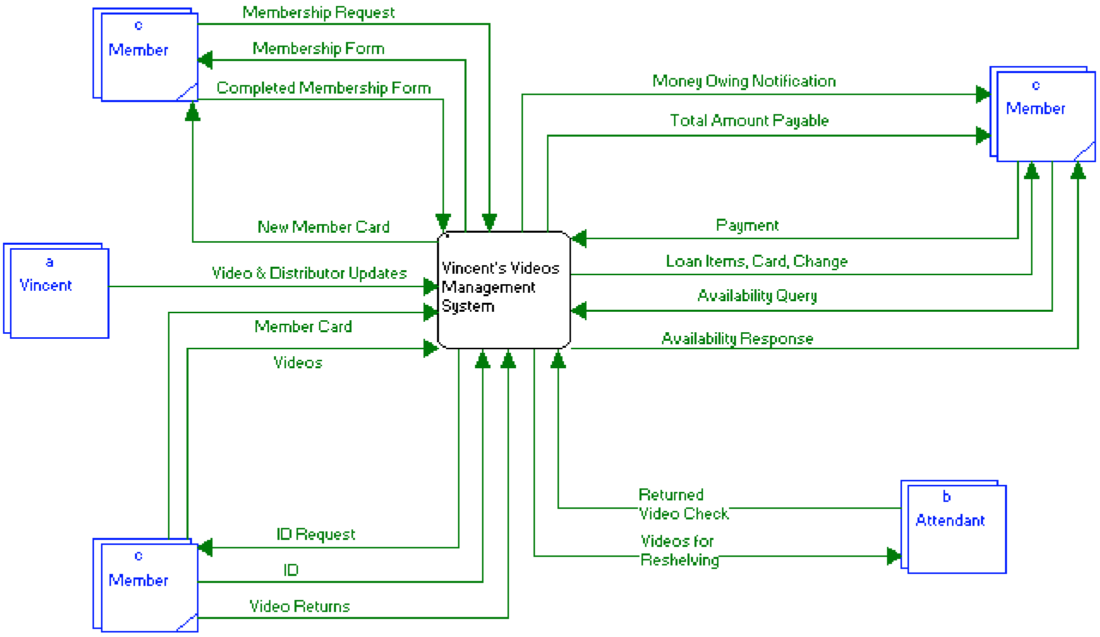
- The members details being displayed on screen once a member number had been entered.
- Overdue fees (if applicable) being displayed once a member number had been entered.
- The ability to find members by either typing in members' numbers or searching on members' names. The search results would need to display member addresses and phone numbers.
- Video title, rating and cost being displayed once a video number had been entered.
- A screen to assist in identifying if a video was currently available for loan.

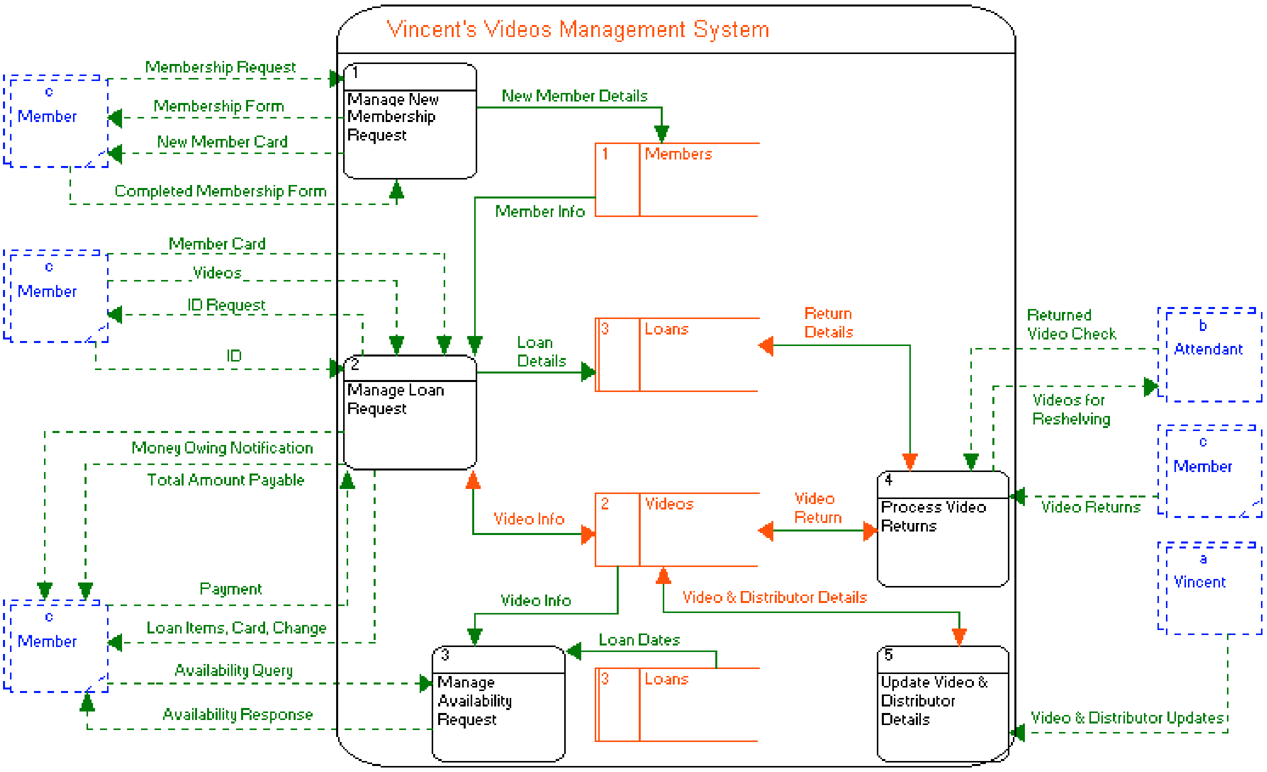
Reporting Requirements

In relation to printed output, Vincent identified the following database reports:

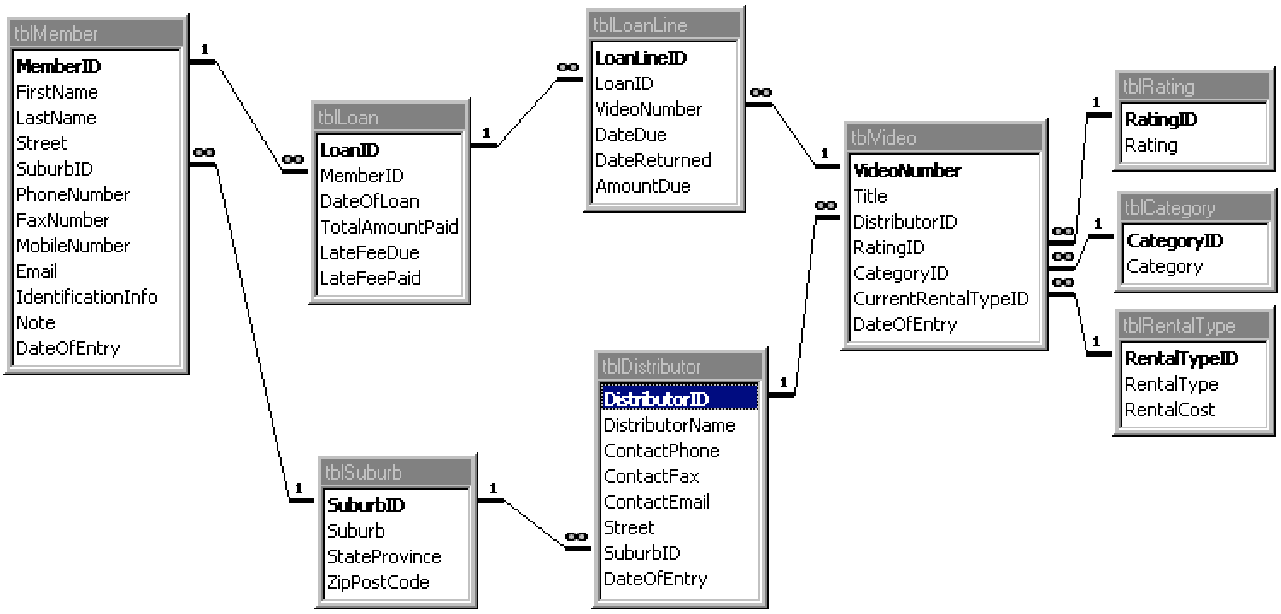
- A report showing all details of a currently displayed loan.
- A report listing currently overdue videos sorted by member name. The report should list member name, phone number, video name, loan date and expected return date.
- A summary financial report listing members who borrowed videos and their final payments, between two given dates, grouped by loan date. The details of individual videos borrowed need **not** be displayed. Total income for each day and the specified period should be presented. Sales tax on the total income should be calculated as 1/11th of the total).

Data Flow Diagrams





Database Tables



Discussion

Objectives and Tasks

Vincent has requested a database to manage the video loans section within his current retail business. An **objective** of this development might be stated as:

To manage members, video stock and loans for Vincent's Video Business

Reviewing each process from the Level 0 Data Flow Diagram in conjunction with the table design:

Manage New Membership Request: This process requires the operator to secure the new member's details on a member form, to enter these details into the system, establish a new member number, and to issue a membership card. This process should be achievable on a single data entry screen, the task being: Add, edit and view member details.

Manage Loan Request: This process requires a new loan event to be recorded, identifying both the borrower and the video(s) being borrowed. The task might be simply stated as: Add, edit and view loan details.

During this process the system needs to alert the operator of outstanding overdue fees associated with the selected member. This might be added as an additional task: Alert operator of overdue fees for the currently selected member.

Manage Availability Request: This process requires the operator to respond to periodic requests from enquirers regarding the availability of one or more videos. The associated task might be stated as: View video availability.

Process Video Returns: This process requires the operator to enter multiple video IDs for videos returned from loan. The task: To efficiently update video loan records to reflect the return of videos.

Update Video & Distributor Details: This process requires the operator to add new video and distributor details, edit video and distributor details and delete videos being removed from stock. The associated tasks might be: Add, edit, delete and view video stock details, and: Add, edit, delete and view video distributor details.

Vincent has requested three reports:

1. Loan details for a specified loan
2. An Overdue Videos Report
3. A Financial Summary Report

Tasks for each of these reports, along with those discussed above, are presented on a sample **Objectives and Tasks Worksheet** presented after the case study discussion.

The Initial Draft

Based on the considerations of the previous section and the initial table design, a rough map of the database application can be drafted. A sample draft has been provided on the **Initial Draft Worksheet** included with this case study.

The operator has to be able to add, edit, view and delete member, video stock and distributor details. These could be added via separate standard data entry screens, or within tab pages on a 'tabbed' screen, accessible from a main menu.

The operator needs to be able to add, edit and view loan details. This might be best managed from a standard 'orderform' style screen:

- Enter loan details at the top of the screen, including the selection of the loaning member.
- Enter the loan line details (data associated with the videos being loaned) at the bottom of the screen.

The loan screen also needs to alert the operator of **overdue fees** for the currently selected member. This could be achieved with a relevant message being displayed within the loans screen once the member has been selected, or as a popup message.

Further, it might be appropriate to link the **loan detail report** to this screen, as both the loan detail screen and loan detail report will contain much the same data.

In regards to viewing video availability, a separate screen could be added and activated from the loan detail screen since this would be the primary screen within the application, and would often be in use when a member approaches the attendant with a query. The popup screen should allow the operator to locate a video's availability promptly, searching by part or all of a video's title. The popup screen should also present the video's category once it is loaded.

In regard to the return of videos, a screen might be activated from the loan detail screen or main menu allowing the operator to enter each video's code and to finalise the update by either clicking a button or pressing ENTER on the keyboard. Details of the member loaning the video along with the loan date might be displayed for review purposes.

In regard to the remaining reports, the Overdue Loans report might be presented automatically at the start of each day, with the provision to be able to revisit the report at any time (should the need arise). The Financial Report should be accessible from the main menu or reports menu and allow the operator to specify a start and end date for the period required.

Dedicated find or search components can be added to each component of the application. In relation to the standard data entry screens, such as members for instance, 'Find' combo boxes could be employed to assist the operator to find records quickly. In relation to the Loans form, a more complex popup search dialog box may need to be employed - requiring the operator to locate the member and then select the loan record that is being sought.

The Interface Style

This is a standard business application, likely to be run in a dedicated manner on one computer or a small network within the Vincent's Video business. As previously suggested, the main Video Loans form might best be presented in a standard orderform style. Many of the other forms could be presented as standard detail forms, or possibly on tab pages within a tabbed form.

The headers and footers should be acceptable to the business and appropriate for a business style application. The business logo might be presented on a splash page and/or the main menu. The business logo could also be presented within the various reports.

Otherwise the various screens and reports should have consistent formatting - similar header and footer styles, and consistent placement and presentation (sizing, colour etc) of controls such as buttons, textboxes, labels and so forth.

The **Interface Style Worksheets** (included with this case study) offer a view of how these consideration might be developed as a set of initial rough sketches.

The Prototype and Review

As with the sample solution for Brenda's Basketball courts, the sample solution for Vincent's Video business is somewhat more than a prototype, although falls significantly short of a completed database system. For instance, the Overdue Fees component within the prototype has only been partially automated so as to reasonably illustrate the functional design rather than the full and correct implementation.

The sample solution is provided as a Microsoft Access database - Vincents.mdb.

Objectives and Tasks

1. Objectives and Tasks Worksheet

Print (in pencil) one objective on a fresh copy of this page, and list all associated tasks. This page is to be used as a cover sheet for all pages of functional design associated with this objective.

Project: Vincent's Videos

Completed By: ___________________________

Date: _______________________

Objective:

To manage members, video stock and loans for Vincent's Videos Business.

Reviewed: ________________

Associated Tasks:

1. Add, edit and view member details.
2. Add, edit and view loan details.
3. Alert operator of overdue fees for the currently selected member.
4. View video availability.
5. To efficiently update video loan records to reflect the return of videos.
6. Add, edit, delete and view video stock details.
7. Add, edit, delete and view video distributor details.

8. Print the loan details for a specified loan.
9. Print a Daily Overdue Loans report.
10. Print a Financial Summary report.

Reviewed: ________________

The Initial Draft

2. Initial Draft Worksheet

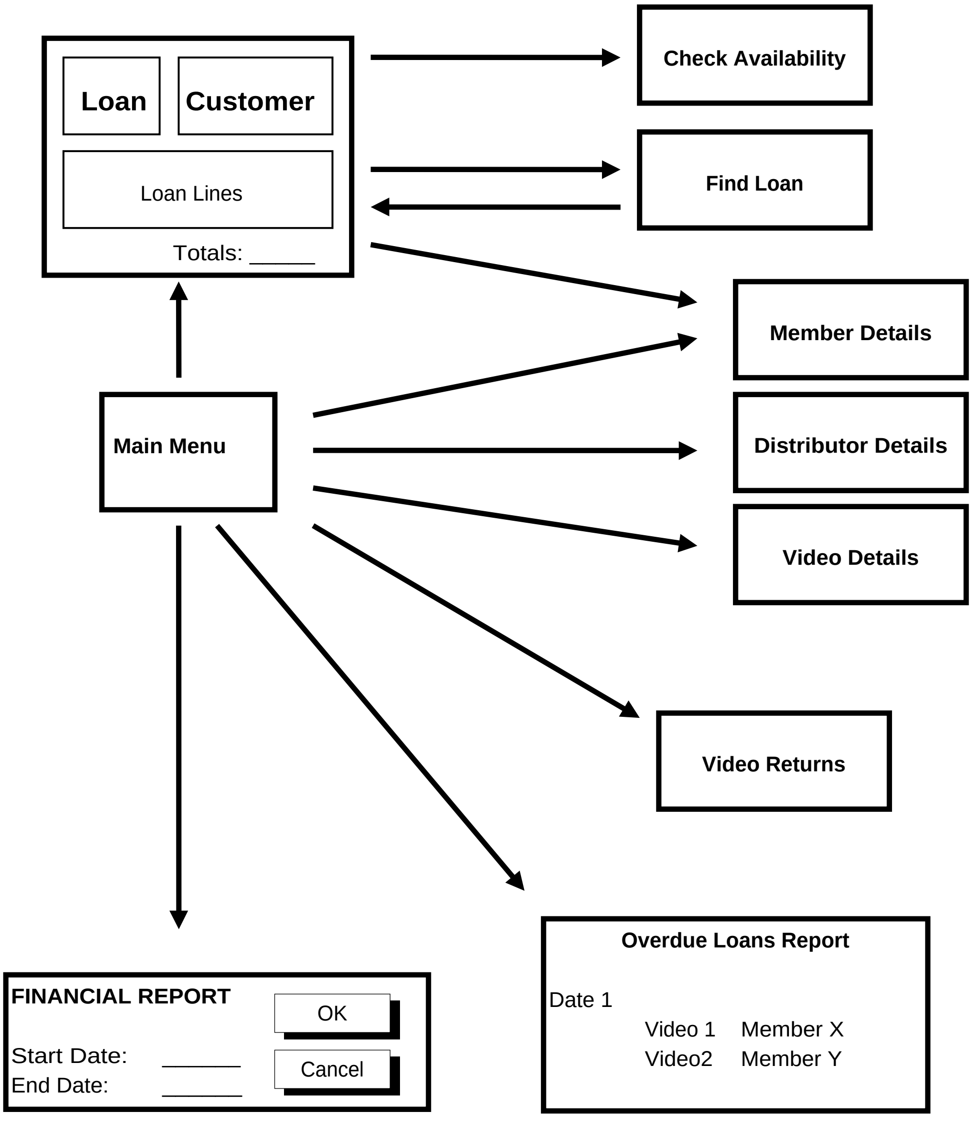
On additional paper, draw (in pencil) rough sketches of, or note down in words what might be on each screen and report. On this page, draw a rough sketch of the map that shows the links between the various screens and reports.

Project: Vincent's Videos

Completed By: ___________________________

Date: _______________________

Screen / Report Map:



Reviewed: ________________

The Interface Style

3. Interface Style Worksheet

Draw (in pencil) rough sketches of screen layout/styles, as well as report headers and footers.

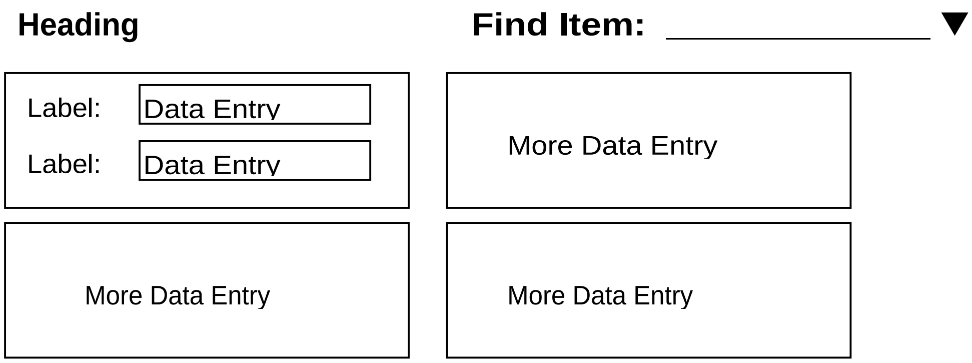
Project: Vincent's Videos

Completed By: ___________________________

Date: _______________________

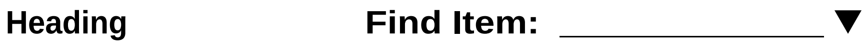
Screen/Report Layout(s):

The screens:



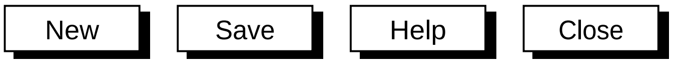
Reviewed: _________________

Screen/Report Header:



Reviewed: _________________

Screen/Report Footer:



Reviewed: _________________

3. Interface Style Worksheet

Draw (in pencil) rough sketches of screen layout/styles, as well as report headers and footers.

Project: Vincent's Videos

Completed By: ___________________________

Date: ______________________

Screen/Report Layout(s):

Group Heading

Heading 1	Heading 2	Heading 3	Heading 4
Data Item 1	Item 2	Item 3	Item 4
Data Item 1	Item 2	Item 3	Item 4
Data Item 1	Item 2	Item 3	Item 4

Reviewed: ________________

Screen/Report Header:

Logo

Business Address
Contact Details

Report Heading

Printed On ...
Vincent's Videos
Between Date 1 and Date 2

Reviewed: ________________

Screen/Report Footer:

Page Number (on right hand side)

Reviewed: ________________

Screens and Reports

4. Screens and Reports Worksheet

On additional paper, draw sketches of screen and report layouts / styles. Fill in the details of the screens or reports on separate copies of this worksheet. Only fill in details relevant to each type of screen or report.

Project: Vincent's Videos

Completed By: ___________________________

Date: _______________________

Objective	To manage members, video stock and loans for Vincent's Videos Business.
Task	Add, edit and view loan details.
Window Title	Video Loans - Vincent's Videos
Screen/Report Title	Vincent's Videos - Video Loans
Screen/Report Name	frmLoansMF
Sub-Screen/Report Names	frmLoansSF
Underlying Tables and Queries (possibly SQL)	frmLoansMF: All fields from tblLoan, tblMember and tblSuburb frmLoansSF: All fields from tblLoanLine, tblVideo and tblRating Reviewed: ________________

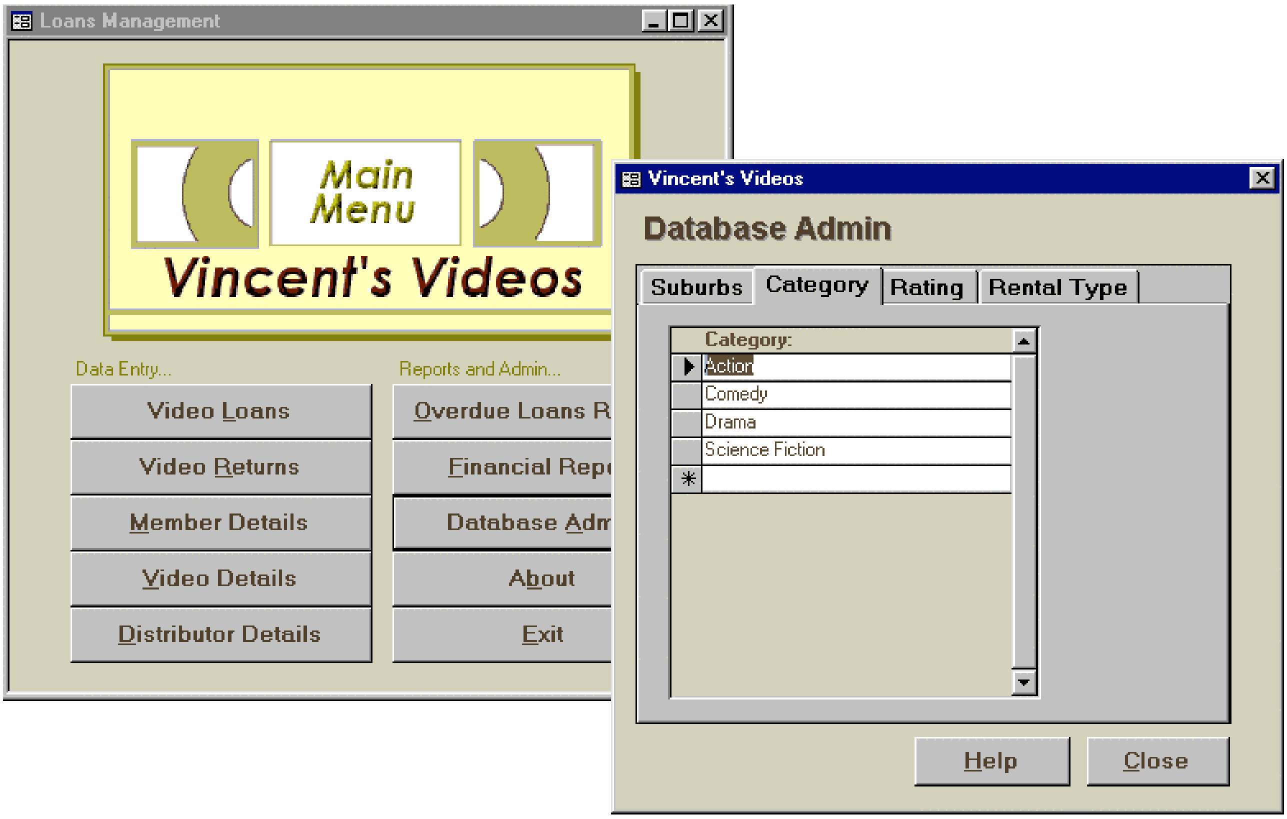
Field Descriptions

Field Descriptor	Mandit-ory	Justify (L/C/R)	Field Type	Size	Conditions / Validation / Format / Other
frmLoanMF					
LoanID	Yes	L	Number	Long	
LoanDate	Yes	L	Date		
MemberNo	Yes	L	Number	Long	
Member	Yes	L	Text	40	
MemberPhone	No	L	Text	15	
Street	Yes	L	Text	100	
Suburb/Province	Yes	L	Text	20	
State	Yes	L	Text	4	
PostZipcode	Yes	L	Text	8	
LateFeesDue	No	R	Currency		Calculated against a new loan
LateFeesPaid	No	R	Currency		
TotalDue	Yes	R	Currency		Sum of video loan charges
TransactionTotal	Yes	R	Currency		Video loan + late fees
TotalSalesTax	Yes	R	Currency		Transaction Total / 11
frmLoansSF					
VideoNumber	Yes	L	Number	Long	
Amount Due	Yes	R	Currency		
DateDue	Yes	R	Date		
Rating	Yes	L	Text	20	
					Reviewed: ________________

Automation

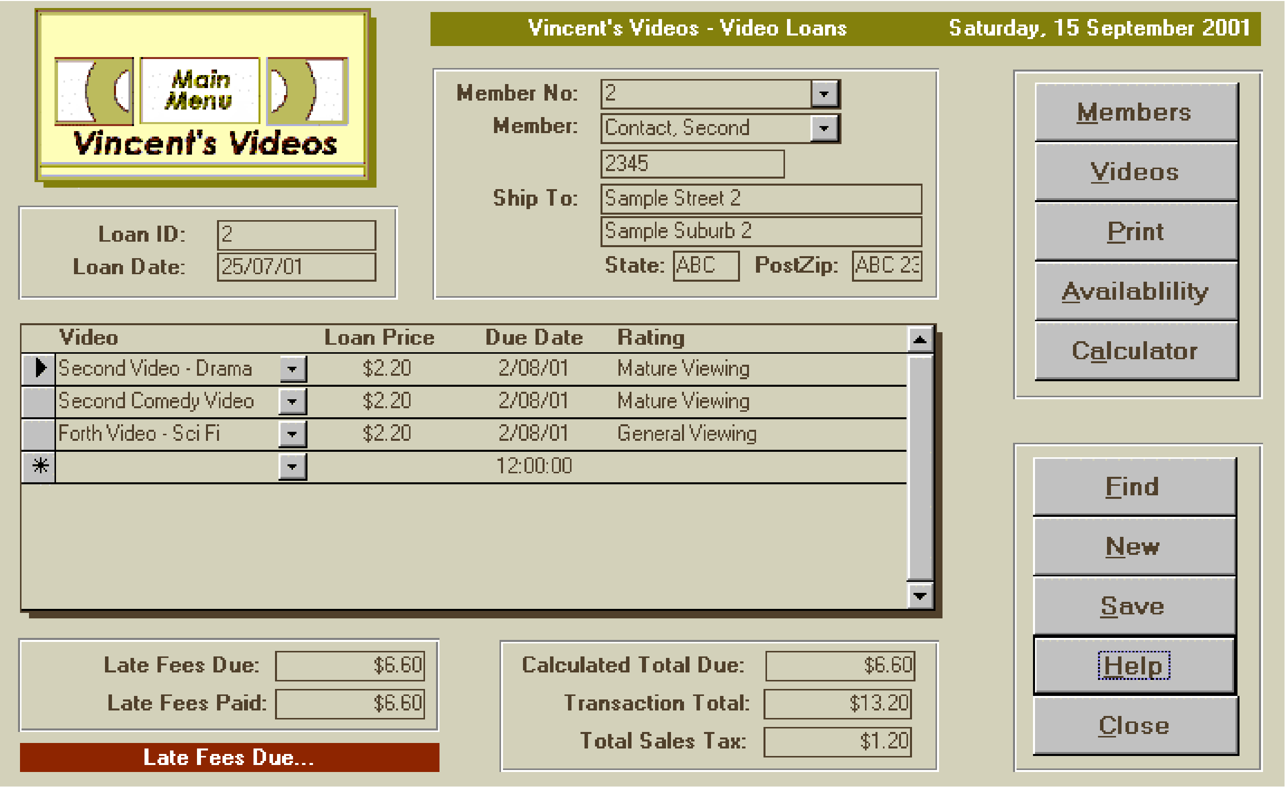
Object	Event Trigger	Function
Form	Load	Maximise form on screen
Select Member combo box	After Update	Identify, sum and display any overdue fees for the selected member
Select Video combo box	After Update	Locate and transfer the Loan Price for the selected video
Members Button	Click	Display the Member data entry screen
Videos Button	Click	Display the Video data entry screen
Print Button	Click	Save the current loan detail, and display the Loan Detail report for the current loan
Availability Button	Click	Display the Video Availability dialog box
Calculator Button	Click	Display calculator
Find Button	Click	Display the Find Loan dialog box
New Button	Click	Save the current record, clear the screen ready for a new loan record
Save Button	Click	Save the current loan record
Help Button	Click	Display Help associated with the current screen
Close Button	Click	Close the form
		Reviewed: ________________

The Prototype



Database Admin

Category:
Action
Comedy
Drama
Science Fiction



Vincent's Videos - Video Loans Saturday, 15 September 2001

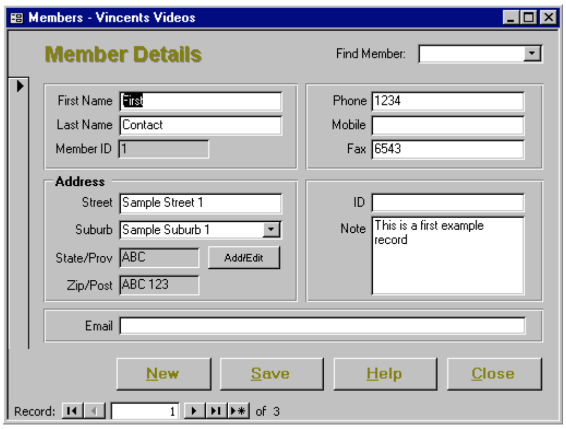
Loan ID: 2
Loan Date: 25/07/01

Member No: 2
Member: Contact, Second
2345
Ship To: Sample Street 2
Sample Suburb 2
State: ABC PostZip: ABC 23

Video	Loan Price	Due Date	Rating
Second Video - Drama	$2.20	2/08/01	Mature Viewing
Second Comedy Video	$2.20	2/08/01	Mature Viewing
Forth Video - Sci Fi	$2.20	2/08/01	General Viewing
		12:00:00	

Late Fees Due: $6.60
Late Fees Paid: $6.60

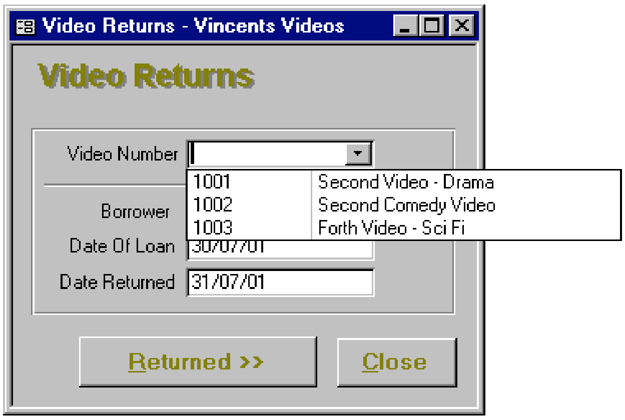
Calculated Total Due: $6.60
Transaction Total: $13.20
Total Sales Tax: $1.20



First Name: First
Last Name: Contact
Member ID: 1
Phone: 1234
Mobile:
Fax: 6543

Address
Street: Sample Street 1
Suburb: Sample Suburb 1
State/Prov: ABC
Zip/Post: ABC 123
ID:
Note: This is a first example record
Email:

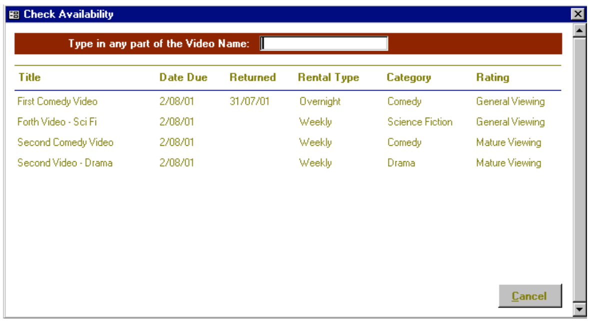
Record: 1 of 3



Video Number:

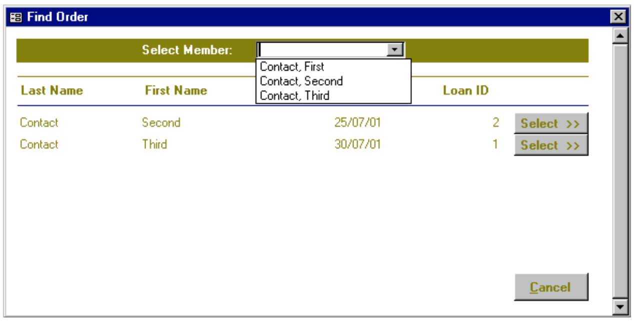
1001	Second Video - Drama
1002	Second Comedy Video
1003	Forth Video - Sci Fi

Borrower:
Date Of Loan: 30/07/01
Date Returned: 31/07/01



Type in any part of the Video Name:

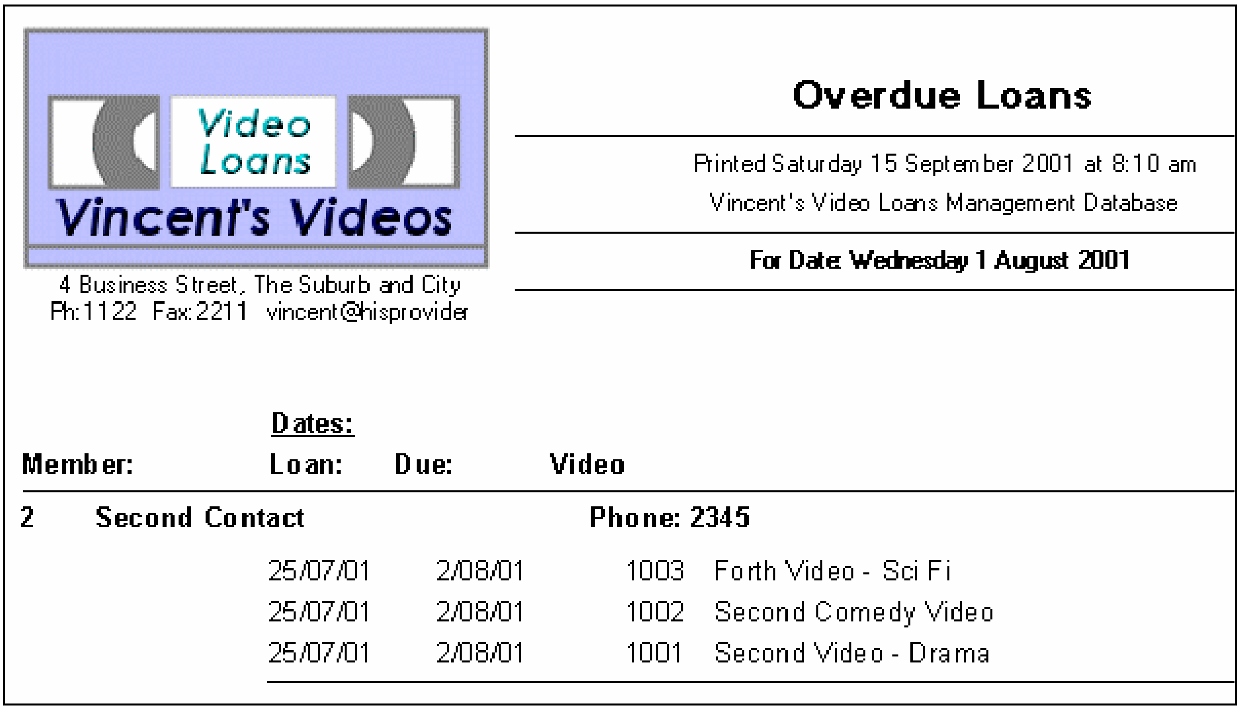
Title	Date Due	Returned	Rental Type	Category	Rating
First Comedy Video	2/08/01	31/07/01	Overnight	Comedy	General Viewing
Forth Video - Sci Fi	2/08/01		Weekly	Science Fiction	General Viewing
Second Comedy Video	2/08/01		Weekly	Comedy	Mature Viewing
Second Video - Drama	2/08/01		Weekly	Drama	Mature Viewing



Select Member:

Contact, First
Contact, Second
Contact, Third

Last Name	First Name		Loan ID
Contact	Second	25/07/01	2
Contact	Third	30/07/01	1


Video Loans
Vincent's Videos

4 Business Street, The Suburb and City
Ph:1122 Fax:2211 vincent@hisprovider

Overdue Loans

Printed Saturday 15 September 2001 at 8:10 am
Vincent's Video Loans Management Database

For Date Wednesday 1 August 2001

	Dates:			
Member:	Loan:	Due:	Video	
2 Second Contact			Phone: 2345	
	25/07/01	2/08/01	1003	Forth Video - Sci Fi
	25/07/01	2/08/01	1002	Second Comedy Video
	25/07/01	2/08/01	1001	Second Video - Drama

Assignment Activities

Erin's Electronics Engineering

Erin owns and runs an electronics engineering firm which primarily manufactures and supplies specialised electronic equipment for various trade and industry groups. Some of the electronic equipment is custom designed and developed, while other items are sold as part of a general product line.

A client may ring Erin's Electronics Engineering firm and specify their requirement for a specialised piece of equipment. The client and request details are recorded in a request log book. At other times, Erin or other members of the firm may consider a new piece of equipment to add to the product line. The details of such equipment is also added to the request log. The originator of the new product line item is recorded, though client details are not relevant.

On a biweekly basis, a development meeting is held and the new request log entries are discussed by the engineers. A project manager is selected for each development project and their name is recorded against the request in the log. The project manager is responsible for the development of the specified product.

After analysis of each project is completed, a list of electronic components is generated and a copy is placed in Erin's in-tray. Erin is responsible for creating weekly orders for the firm's various component suppliers. These orders do not need to be tracked by the database.

Erin has asked your development team to consider the development of a database to assist in the management of development requests. Although the system could assist in the management of other business tasks, such as orders and invoicing, Erin wishes to restrict the initial database system to the management of specialised client requests, new product line items and component lists.

Data Storage Requirements

In relation to the storage of data, Erin indicated that she would like to record:

- Client details, including business name, a contact person, one address, phone, fax, email and a section for general notes.
- Specialised request details, including name of item, equipment type, request date, scheduled completion date, a notes area for the item's function or use, and a notes area for preliminary technical detail.
- Product item details including much the same detail as that recorded for the specialised requests.
- Component list and development notes, including lists of components required for each development, and the procedure for the item's development.
- Suppliers, including address, phone and email details, and a simple listing of the supplier's components and their approximate costs.

Data Entry Requirements

In relation to the entry of data into the database, Erin indicated that she needs to:

- Enter all the firm's current suppliers, and their respective component listings.
- Add the current product line items, listing their respective components and development procedure steps.
- Add the preliminary details of development projects as the various client requests and suggestions come in.
- Add the component and development procedure for the various projects as the details become available.

On-Screen Review Requirements

In relation to the information to be displayed on screen, Erin suggested that she would like to search and view:

- Client details
- Supplier details
- Specialised client request details
- Product line items details
- Supplier components, by supplier and/or item type.

Reporting Requirements

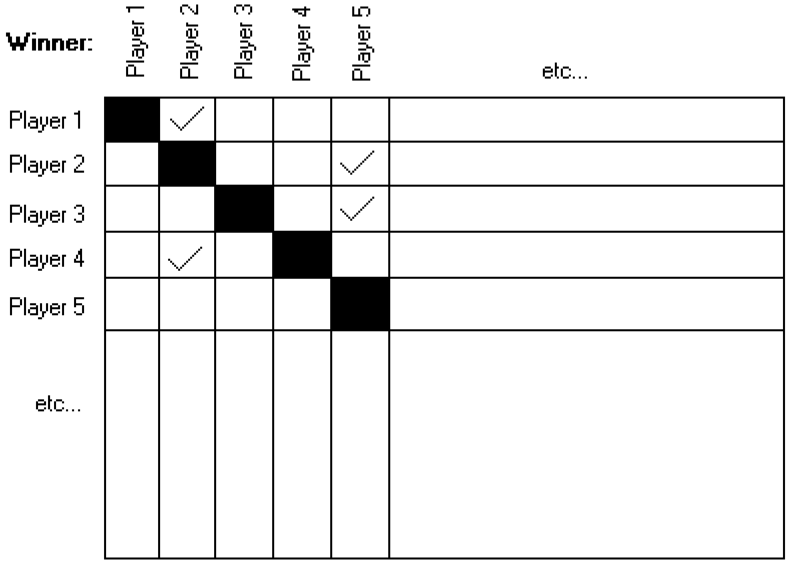
In relation to printed output, Erin identified the following database reports:

- A list of new projects, both new specialised client requests and new product line suggestions. (To be used as part of the agenda for the biweekly development meetings). This report is to be grouped and sorted by equipment type and then by request date.
- A single supplier's component listing, grouped by component type.
- The full details of any given development, including client details (if relevant), the product and/or request details, the development component list and procedure.

Chris' Chess Competition

Chris organises regular chess tournaments every 3 months. Each tournament is played over a weekend and is usually restricted to 10 competitors. Each competitor must play every other competitor once, and the competitor who wins the largest number of matches is declared the winner. One or more tiebreaker games may have to be played.

Chris currently uses a paper-based 10 by 10 grid with players names down one side and across the top, and games results are recorded within the grid. Total wins are determined after all games have been played.



Winner:	Player 1	Player 2	Player 3	Player 4	Player 5	etc...
Player 1	■	✓				
Player 2		■			✓	
Player 3			■		✓	
Player 4		✓		■		
Player 5					■	
etc...						

Chris is responsible for alerting a marketing officer that a new tournament is being organised. Subsequently, potential players contact Chris directly with their name and contact details. Chris accepts the first 10 callers and confirms their status as a competitor in the next tournament. Any further callers are placed on a waiting list queue and may be confirmed as competitors if existing competitors indicate their inability to play.

Once the tournament is complete, Chris sends a summary of the results to a local paper for publication.

Chris has asked your development team to consider the development of a database to manage the tournament details including player details and game results. The system should allow Chris to print summary results for the local paper as well as game summary slips for each competitor. The system should also be able to generate annual tournament statistics, including most improved player(s) and the highest scoring chess player(s).

Data Storage Requirements

In relation to the storage of data, Chris indicated that he would like to record:

- Player details including an address, phone and email details.
- Game details including dates and venues, competitor pairs for each round within the given tournament, and game results.

Data Entry Requirements

In relation to the entry of data into the database, Chris indicated that he needs to:

- Enter the details of the next tournament including the competitors and the waiting lists. These lists must identify the order that the various contacts were made.
- Generate all the games (including competitor names) for a given tournament with a click of a button.
- Enter the results of each round within a tournament at the end of each round. (NOTE: if a tournament has 10 players, then each player will play 9 other competitors. This would constitute nine rounds)
- Add the details and results of any necessary tiebreaker games.

On-Screen Review Requirements

In relation to the information to be displayed on screen, Chris suggested that he would require:

- The ability to view the progressive tournament results for all players at any time.
- The database to display the winner and runner up at the end of the tournament.

Reporting Requirements

In relation to printed output, Chris identified the following database reports:

- A copy of the game grid, similar to that used currently, with players names listed across the top and down the left side of the grid.
- A list of competitors and their total scores for a given tournament, sorted by score.
- A summary of each player’s scores for a given tournament, sorted by player name.
- A complete summary of all players scores for tournaments between two specified dates, grouped by player and tournament within that. Any relevant totals should be included.

Appendix A - Design Worksheets

The worksheets included in this appendix are:

1. Objectives and Tasks Worksheet
2. Initial Draft Worksheet
3. Interface Style Worksheet
4. Screens and Reports Worksheet

1. *Objectives and Tasks Worksheet*

Print (in pencil) one objective on a fresh copy of this page, and list all associated tasks. This page is to be used as a cover sheet for all pages of functional design associated with this objective.

Project: ________________________________

Completed By: __________________________

Date: ______________________

Objective:

Reviewed: _________________

Associated Tasks:

Reviewed: _________________

2. Initial Draft Worksheet

On additional paper, draw (in pencil) rough sketches of, or note down in words what might be on each screen and report. On this page, draw a rough sketch of the map that shows the links between the various screens and reports.

Project: _________________________________

Completed By: __________________________

Date: ______________________

Screen / Report Map:

Reviewed: _________________

3. *Interface Style Worksheet*

Draw (in pencil) rough sketches of screen layout/styles, as well as report headers and footers.

Project: ________________________________

Completed By: _________________________

Date: ______________________

Screen/Report Layout(s):

Reviewed: ________________

Screen/Report Header:

Reviewed: ________________

Screen/Report Footer:

Reviewed: ________________

4. Screens and Reports Worksheet

On additional paper, draw sketches of screen and report layouts / styles. Fill in the details of the screens or reports on separate copies of this worksheet. Only fill in details relevant to each type of screen or report.

Project: ________________________________

Completed By: __________________________

Date: ______________________

Objective	
Task	
Window Title	
Screen/Report Title	
Screen/Report Name	
Sub-Screen/Report Names	
Underlying Tables and Queries (possibly SQL)	

Reviewed: _________________

Field Descriptions

Field Descriptor	Manditory	Justify (L/C/R)	Field Type	Size	Conditions / Validation / Format / Other

Reviewed: ________________

Automation

Object	Event Trigger	Function

Reviewed: ________________

Additional References and Resources

NOTE: *A number of the texts listed will have updates and new editions.*

D'Orazio & Happel	**Practical Data Modelling for Database Design** Wiley, 1996
Garner, Stuart	**Systems Analysis and Design Using Ascent** Knowledge Base (www.knowledgebase.com.au) (Supplier of the Ascent CASE Tool)
Getz, Litwin & Reddick	**Microsoft Access 2 Developer's Handbook** Sybex, San Francisco, 1994
Hawryszkiewycz, I.	**Systems Analysis and Design** (various editions) Prentice Hall
Jennings, Roger	**Using Access 97 (Platinum Edition)** QUE, 1997
Kendall & Kendall	**Systems Analysis and Design** (various editions) Prentice Hall
Litwin, Getz & Gilbert	**Access 97 Developer's Handbook** Sybex, San Francisco, 1997
Norman, Ronald	**Object Oriented Systems Analysis and Design** Prentice Hall
Shelly Cashman & Adamski	**Systems Analysis and Design** (various editions) Thomas Nelson
Weinschenk & Yeo	**Guidelines for Enterprise-Wide GUI Design** Wiley 1995