THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2002

EXAMINERS’ REPORT

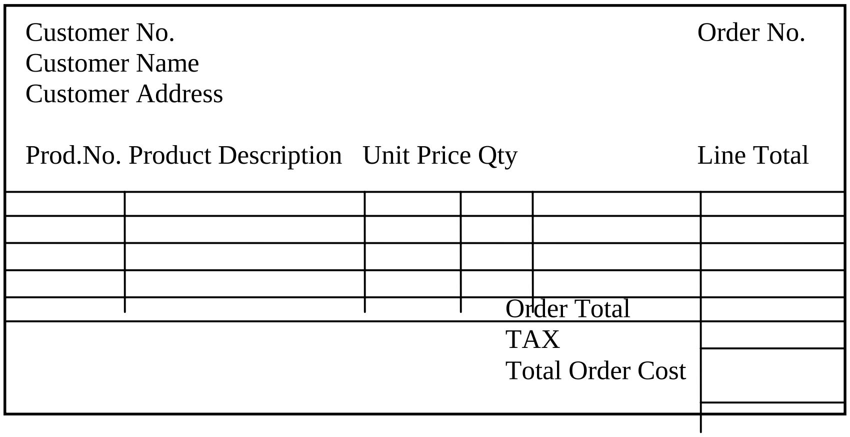
Systems Design

Question 1
(a)

Outline how modern screen-form design techniques can be used to provide interactive validation of data entered by keyboard or other means, and to minimize the time-consuming entry of keyed data.

(4 marks)

(b)
An order entry system is based on the clerical document shown below:



Customer No. Order No.
Customer Name
Customer Address

Prod.No.	Product Description	Unit Price	Qty		Line Total
				Order Total	
				TAX	
				Total Order Cost	

(i) **Explain the types of data entry validation needed for the fields in the above form.**

(ii) **State which fields would have data keyed in, and which would show as retrieved data dependent on other keyed fields.**

(iii) **Identify which key fields would need to be linked in order for the screen form to be used satisfactorily by operators to view orders from different customers.**

(15 marks)

(c)
Name and outline the content of the database tables from which the screen form would be derived.

(6 marks)

Answer Pointers

(a)

- Look –up of descriptive data from keyed codes – minimising potentially inaccurate keying, increasing entry speed, and providing validation of keyed codes.
- Numeric data can be checked for valid data ranges
- Alphanumeric data can be checked for valid patterns
- Pull-down lists can be used for entry, reducing keying effort
- On-screen assistance can be an effective aid to the user, and reduce training requirements
- Use of buttons, etc. (4 marks)

(b) (i)

Order number – (if keyed) – must be a valid number, range and pattern check
Customer number – range and pattern check

- customer name and address, for initial keying, likely to be structured, when retrieved would be checked with customer

Product number – initially must be range and pattern checked, when input for order must be valid (NOTE: referential integrity applies for any items which are to be retrieved – an item which has been used in an order cannot be deleted, otherwise all reference to the product is lost.
Similar checks would be done for other fields, particularly order quantity checks for both stock availability and reasonableness. The difference between optional and obligatory fields should be indicated.
(5 marks)

(ii)

- Order number – may be keyed, or more likely autogenerated to ensure no duplicated order numbers
- Input customer number – customer details appear on screen, to be checked with customer/customer order form.
- Key in product number, price and description appear on screen, providing validation of product identity
- Key in quantity, line total generated, also running grand total.
- Quantity range limit check, may vary with each product, guarding against accidental mis-keying. (Variations on this aspect can also be used on customer buying profiles)
- NOTE: keying is minimized and accuracy improved by the use of calculated fields, such as line and order totals.

(5 marks)

(iii)

The order number and product number must be linked so that when operator scrolls through orders, the products ordered appear appropriately.
Order number customer number must be linked to retrieve customer details, when scrolling through orders.
NOTE: The order form shown would be a main form for the customer with a sub-form for the products ordered.
(5 marks)

(c)

The four normalized tables (files) likely to be used are:-

Customer file:	Customer number with customer details
Product file:	Product number, with product details such as price, description.

Transaction file:	Order number, product number, quantity
Order file:	Order number, customer number

(6 marks)

Examiners' Comments
This is a standard relational database normalization to the third degree. Note also that the files updated in the ordering process are order file and transaction file. The other two files would be updated via separate screen forms.
It should be remembered that only ONE file can be updated at one time, hence the TWO forms, one for updating the order number/ customer file, the other the order number/product number /quantity file.
This question was not generally well answered, and reflects on the quality of the tuition given to the candidates. Understanding of basic normalization of a universal situation is essential to effective systems design.

Candidates must have some practical experience of designing and implementing such applications – even in an educational environment.

It is also essential that candidates understand the difference between reference files and transaction files. A typical basic system would have the form/sub-form for the ordering process, and separate forms for product and customer file maintenance.

Question 2

(a) Describe, with supporting examples, the components of a fourth generation language (4GL) and how they combine to form an effective prototyping tool.

(10 marks)

(b) Discuss possible limitations, when prototyping with such tools in order to design systems, and explain how you would attempt to minimise any difficulties.

(15 marks)

Answer Pointers
(a)

- Relational table building system
- Relational query generator
- Screen form builder
- Report generator

All the above points with explanations – the order form from Question One could have been used as an example.
These combine to facilitate prototyping of applications in a 'bottom up' manner, and can assist both designer and builder by allowing rapid implementation of changes, and ease of demonstration to users.

(10 marks)
(b)

- Scaleability problems
- Integration problems due to 'bottom up' approach
- Danger of uncontrolled design changes to suit user

- Cost of time spent can get out of control, i.e. project control problems
- Ease of use can raise false expectations in users

Minimised by:

- Strict project control
- Changes approved and documented
- Overall agreement on dialogues, etc.
- Incorporation into CASE tool set.

(15 marks)

Examiners' Comments

Some candidates used the rubric of Question One to provide the example for this question – this showed a good understanding of the subject.
NOTE: Most modern systems provide web page systems to allow on-line updating of database from the internet in e-commerce applications. (ASP web pages)
Candidates did not display the basic understanding of the difference between 'top-down' and 'bottom-up' design, and the necessary compromise needed to implement effective systems.

Question 3

Identify and briefly describe the main features you would expect to find in the following:

(a) Web page generators (5 marks)
(b) HTML pages (5 marks)
(c) Virus detection/scanning software (5 marks)
(d) File server (5 marks)
(e) Firewall software. (5 marks)

Answer Pointers

(a) Web page generators

- Ability to view both HTML source code and WYSIWYG (easy preview)
- Ease of modification of font style, colour, size , etc.
- Ease of adding links
- ASP facility (particularly for on-line updating,etc.)
- FTP built-in

(5 marks)

(b) HTML pages

- Candidates should emphasise that it is a Page description language NOT programming language

Should include

- Meta tag section
- Ease of loading
- Good presentation
- Good navigation

NOTE: It should be emphasized that only code is stored in the web page file, and that all images, etc are separate files. Local and URL link differences and problems are important. HTML code includes details such as Meta tag details which do not show when viewed by a browser.

(5 marks)

(c) Virus scanners

- Updateable (preferably on-line)
- Detect all current viruses including Trojan horse, etc.
- Full scanning, in real-time, not retrospective.
- Disinfect option.
- No conflict with running applications
- Quarantine options

(5 marks)

Candidates should have emphasized that NO virus scanning and detection software is foolproof, and that while constant updating is essential, complementary techniques and constant vigilance are also essential

(d) File server (and software)

- Secure multi-user operating system
- Web server, e.g Apache.
- DNS server
- Firewall
- Database server
- Network management – user management, password control, etc.
- Administrator mode secure

It is NOT enough to say that a modern file server 'serves client computers'

(5 marks)

(e) Firewall software

- Warning of potential port access – NOTE: Designers should be aware of the multiplicity of ports on the modern PC or server, and the need to restrict access to these ports.
- Option to allow access at user discretion
- Fast
- Non-conflicting/safe background operation

(5 marks)

Examiners' Comments

Candidates should have appreciated that parts (a) & (b) are largely complementary. Too many candidates did not understand how firewall software operates, and the need to restrict access to a multiplicity of ports, but that, like virus scanning/detecting software, no firewall is 100% safe.

Part (a) NOTE: Importance of different views of page, to see both the Internet view and the codes/tags only shown in source code view.

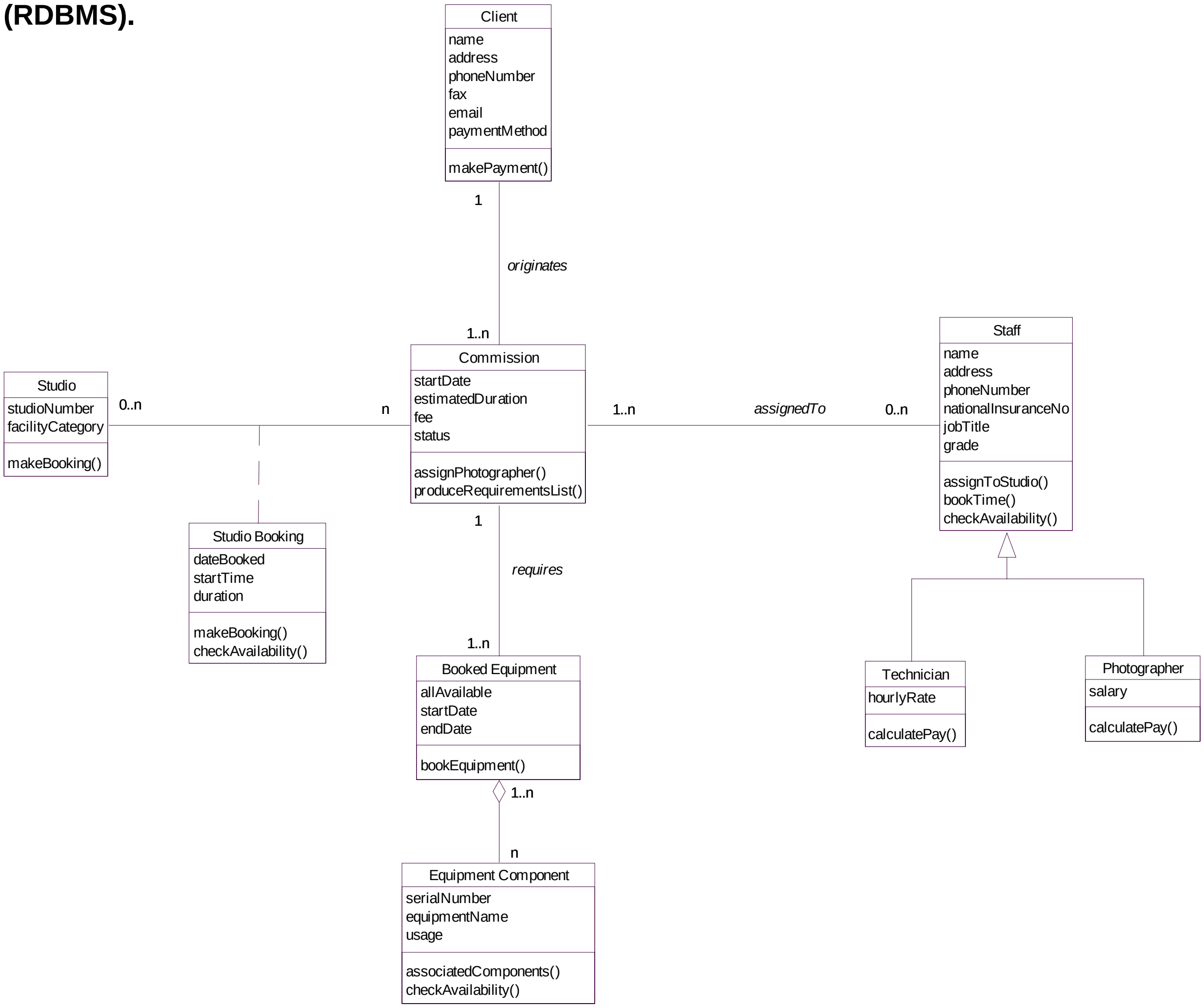
Candidates with experience of web page design appreciated that web page generators insert the html code, while showing the WYSIWYG view to the page designer.

Part (d) Candidates did not understand the universality of file servers for networks of all types and sizes, including the Internet, and the features that would be included in such file servers. The knowledge displayed, was in most cases, outdated, and did not reflect current practice.

Part (e) Firewalls are now an essential part of Internet access, and understanding of the multiplicity of ports that make PCs and servers so vulnerable, plus the process of limiting access to these ports by such firewall software is essential to all system designers. The question was not generally well answered.

Question 4

The following diagram is a fragment from a class diagram for a photographic agency system. It is to be mapped to a Relational Database Management System (RDBMS).



(a) Briefly explain how each of the following elements of the class diagram may be mapped:

- **(i) class**
- **(ii) one to many association**
- **(iii) many to many association**
- **(iv) association (link) class**
- **(v) aggregation**

(10 marks)

(b) Describe three strategies that may be used to implement the 'Staff' inheritance hierarchy when undertaking this mapping.

6 marks)

(c) Produce a suitable relational schema (set of normalised tables) for the above class diagram.

(9 marks)

Answer Pointers

Candidates are expected to base their answer around the following (simplified) principles:

(i) class

For each class create a relation, determine the primary key (if there are no suitable attributes for this create a primary key attribute).

(ii) one to many association

Post the primary key from the relation created for the uni-part class (one end), into the relation created for the multi-part class (many end) to become a foreign key.

(iii) many to many association

Create a new relation for the association, create the primary key as the composite of the primary keys of the relations crated for the classes participating in the association.

(iv) association (link) class

Create a relation as for a n..m association and add the attributes of the link class. Make sure PK is not affected by the addition of these attributes.

(v) aggregation

This aggregation is a n..m multiplicity, therefore deal with this as per the many to many association above.

5 x 2 marks
Total 10 marks

(d) Describe three strategies that may be used to implement the 'Staff' inheritance hierarchy when undertaking this mapping.

(6 marks)

Three strategies are:

1. implement all classes as tables posting staff PK into child class tables;
2. only implement the superclass; subclasses collapse into attributes (with null values where not used);
3. only implement subclasses; superclass is redundantly copied in each subclass.

3 x 2 marks for these points explained or other equivalent explanations
Total 6 marks

(e) Produce a suitable relational schema (set of normalised tables) for the above class diagram.

(9 marks)

This part should be fairly mechanistic as the candidate has already outlined the principles involved in the mapping. Suggested schema:-
Client(client#, name, address, phone_number, fax, email, payment_method)
Commission(commission#, start_date, estimated_duration, fee, status, client#)
Studio(studio#, facility_category)
StudioBooking (studio#,commission#,date_booked, start_time, duration)
BookedEquipment (equipment_booking#, all_available, start_date, end_date, commission#)
EquipmentComponent(serial#, equipment_name, usage)
BookedComponent(equipment_booking#,serial#)
StaffAssignmnet (staff#, commission#)

The staff inhertiance hierarchy should be mapped according to one of the strategies outlined in (b) above. Here the first strategy is shown. (Tstaff# & Pstaff# will be same as the employees staff#)

Staff (staff#, name, address, phone#, national_ins#, job_title, grade)
StaffTechnician(Tstaff#, hourly_rate)
StaffPhotographer (Pstaff#, salary)
The NI number could be used as the PK but here a staff# has been introduced.

Marks breakdown:	Reasonable set of relations	6 marks
	Sensible strategy for dealing with hierarchy	2 marks
	Identified need for date_booked As part of PK in StudioBooking	1 mark

Total 9 marks

Examiners' Comments

The important feature of this question is that object attribute values in an OO system are to be stored within a RDBMS and therefore a mapping of objects to entities is required.

Part (a) – some candidates answered this part well, however, many candidates scored badly on this part due to the fact that they did not explain how the mapping from objects to entities may be achieved. Instead many merely described, for example, what a class was and how it was represented on the class diagram rather than how the mapping may be made to relations, as per the above answer pointers. The same was true for the 1..m, n..m, link class and aggregation.

Part (b) – similarly, although some candidates answered this part well, many candidates merely described what aggregation was and did not explain the three strategies (as above) for mapping the aggregation structure.

Part (c) – this part was generally answered much better, with most candidates at least providing a basic set of tables representing each class and its attributes, however, many did not map the associations, aggregation and generalisation. The better candidates produced a set of tables by following the principles outlined in the answer pointers for parts (a) and (b).

Question 5

(a) With reference to the design of software architecture, draw an annotated diagram and briefly explain its role for each of the following Unified Modelling Language (UML) models:

(i) package diagram

(ii) deployment diagram

(10 marks)

(b) For each of the following UML models explain which aspects of systems design are being modelled:

(i) **class diagram**

(ii) **statechart (state diagram)**

(iii) **sequence diagram**

(9 marks)

(c) Sequence diagrams and collaboration diagrams are both used to model object interaction. Briefly outline the different approach each of these diagrams uses to achieve this.

(6 marks)

Answer Pointers

(a)

(i) Package Diagram: Shows packages (groupings) of classes and the dependencies among them.

(ii) Deployment Diagram: Shows the physical relationships among software & hardware components.

2 x 2 marks for suitable explanations

2 x 3 marks for suitable annotated diagrams

Total 10 marks.

(b) Candidate's explanation should include basic overview of what each model depicts, e.g.

(i) Class Diagram

- class diagram is a static view of the system
- shows all classes to support use case functionality
- class diagrams support the view of data as attributes & processes as operations.

(ii) Statechart Diagram

- dynamic view of the system
- shows changes to condition of an object as a result of an event happening
- Changes of state as a result of events (time) are shown

(iii) Sequence Diagram

- dynamic view of the system
- shows interactions between objects in a time sequence
- Passage of time represented in sequence diagrams linked with processes (message passing)

3 x 3 marks for suitable explanations

Total 9 marks

(c)

- Sequence diagrams emphasise the sequence with the passage of time from top to bottom of the diagram.
- On collaboration diagrams the sequence of messages is indicated by numbering and is not as clear.
- A collaboration diagram may be drawn for each route through a use case.
- Sequence diagrams are drawn for an entire single use case.
- Collaboration diagrams use icons and a similar layout to class diagrams, and:
- therefore are useful in checking the structure of the class diagram (particularly associations)

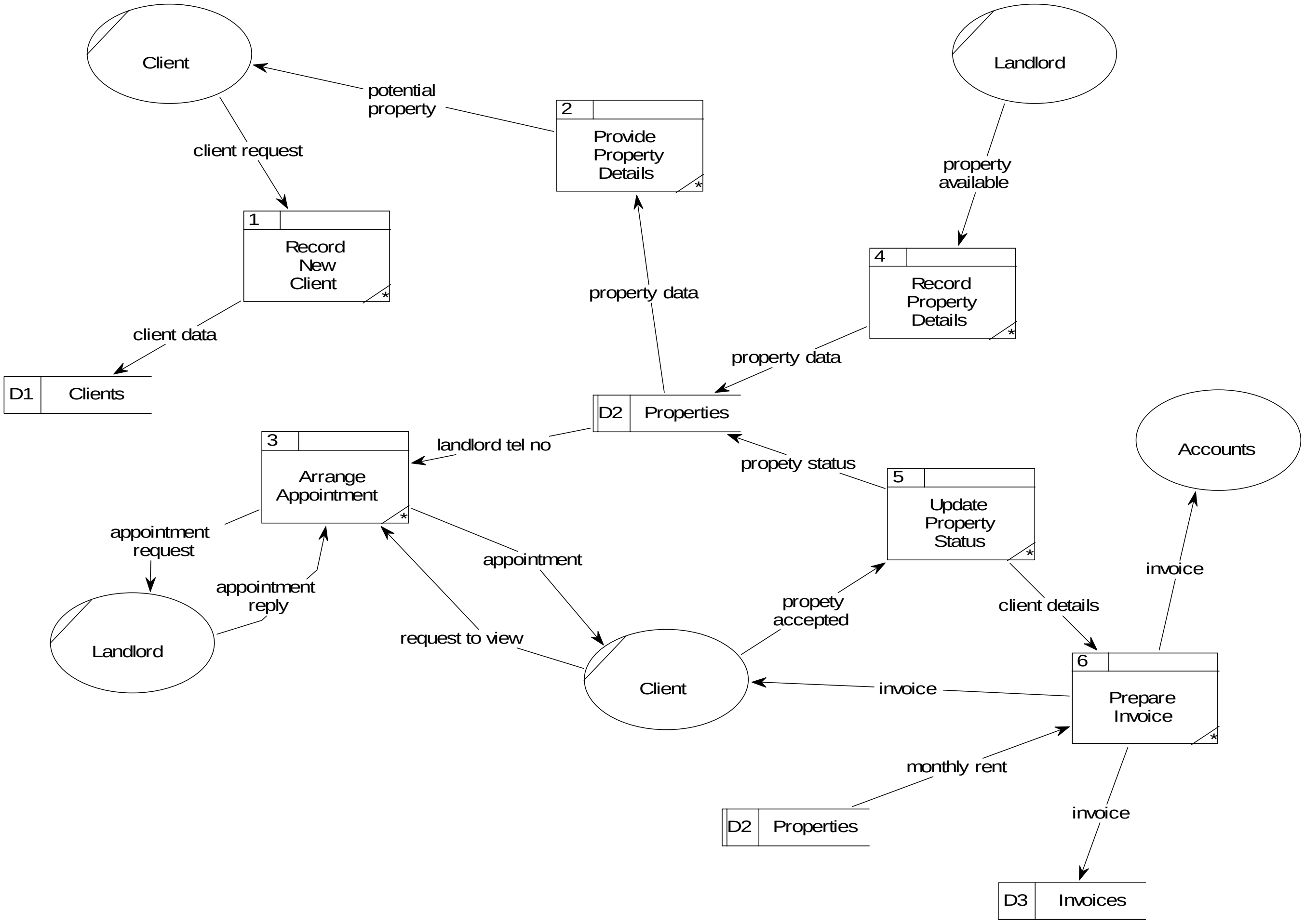
6 x 1 mark (for above or similar)
Total 6 marks

Examiners' Comments

Part (a) – Generally this part was not well answered. Many candidates were not familiar with the UML Package and Deployment diagrams and their role in design. Where candidates were familiar with these models good marks were achieved for this part.

Part (b) – Candidates generally answered this part well with most demonstrating an understanding of the syntax of the models and the aspects of the system being modelled by each. Although not asked for in the question, many candidates illustrated their answer with diagrams, which were generally drawn with accurate syntax and complemented their answers well.

Question 6
The following Data Flow Diagram (DFD) models part of the business processes of an accommodation agency that offers a service to landlords by finding them suitable tenants.



(a) **With reference to transform analysis and transaction analysis, discuss the main principles of how a structure chart may be constructed from the information contained in a data flow diagram.**

(12 Marks)

Answer pointers

(a) Candidates should base their answers around the following (credit will be given for suitable alternative points):

General principles:

Create an overall control module

- Each DFD process becomes a module on the structure chart
- Each flow becomes a coupling

- Control flags added as appropriate to convey condition information
- Group I/O modules under higher level module (transform centres)
- Group modules that work towards same identifiable functions (transaction centres)
- Decompose complex processes & flows

Transform Analysis – uses DFD to identify transform centres (central process with well-defined I/O streams) and data couplings. Processes that provide I/O are converted to modules with additional modules added to allow get/put operations.
Transaction Analysis – uses DFD to identify transaction centres (process splitting input and calling different transactions) and also repetition and decision.

Transform analysis & transaction analysis are the first steps in constructing a structure chart, iterative refinement will be necessary.
(Candidates may support their answer with a diagram to show the structure & syntax).
6 marks for suitable general principles
2 x 3 marks for suitable explanation of transform & transaction analysis
Total 12 marks.

(b) A suggested structure chart can be found on the next page: (suitable alternatives will be marked on merit)
4 marks for processes to modules
4 marks for module groupings & structure
5 marks for appropriate data couplings
Total 13 marks

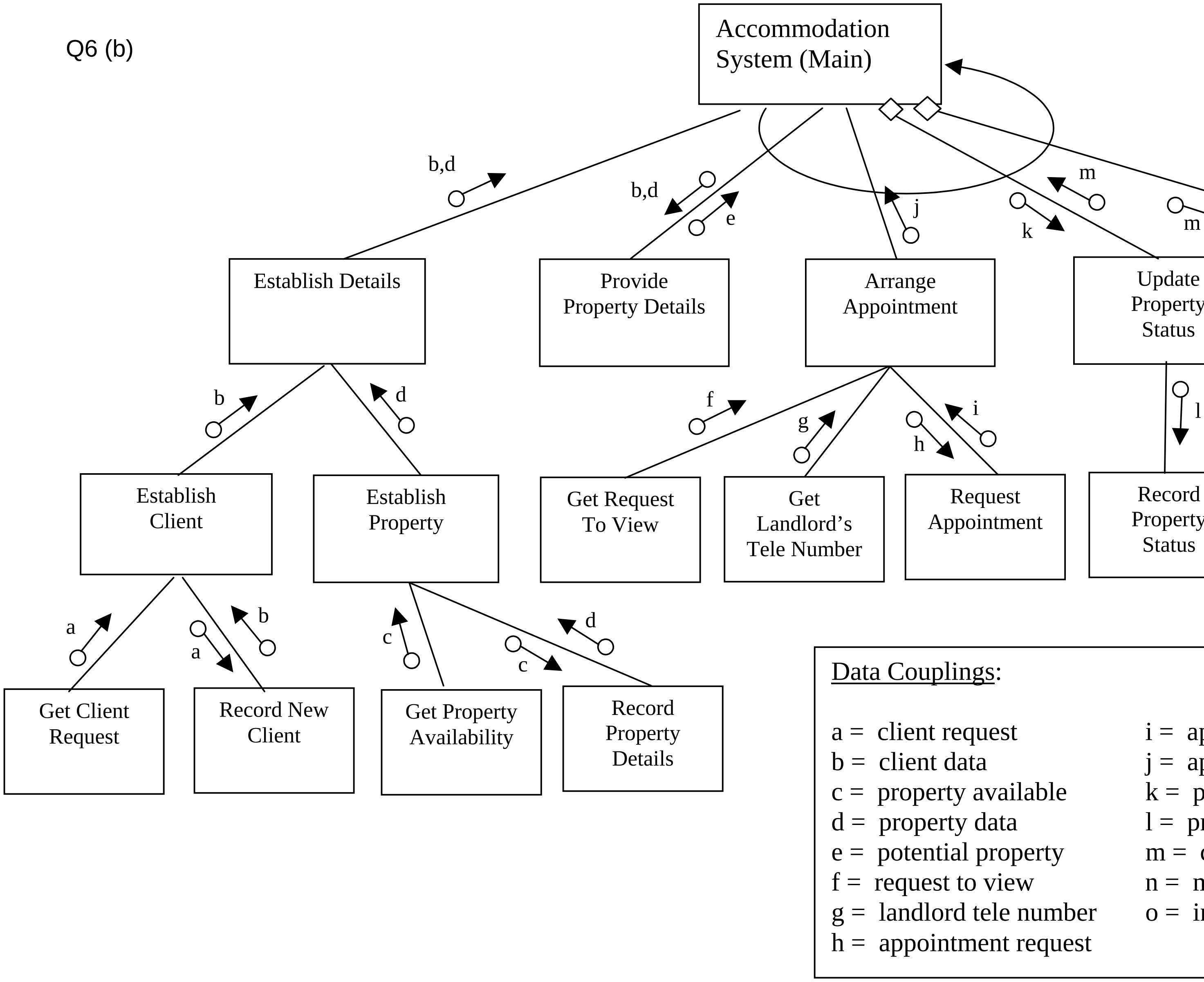
Examiners' Comments

This question was generally well answered by most students.

Part (a) – Most candidates were able to identify the general principles of creating a structure chart. Several candidates were able to discuss transform analysis and transaction analysis in some detail. However, others merely stated, for example, that transform analysis involves the identification of a transform centre without explaining what a transform centre is.

Part (b) – Structure charts produced for this answer were variable. Most candidates were able to map processes to modules and scored well for this, but did less well for the module groupings and structure. Several students did not include data couplings at all and consequently lost marks. It was noticeable that some students outlined the general principles of creating a structure chart in part (a), but then did not follow these principles when answering part (b).

Q6 (b)



Data Couplings:

a =	client request	i =	ap
b =	client data	j =	ap
c =	property available	k =	p
d =	property data	l =	pr
e =	potential property	m =	c
f =	request to view	n =	n
g =	landlord tele number	o =	ir
h =	appointment request		