THE BCS PROFESSIONAL EXAMINATIONS
Diploma

April 2006

EXAMINERS' REPORT

Systems Design

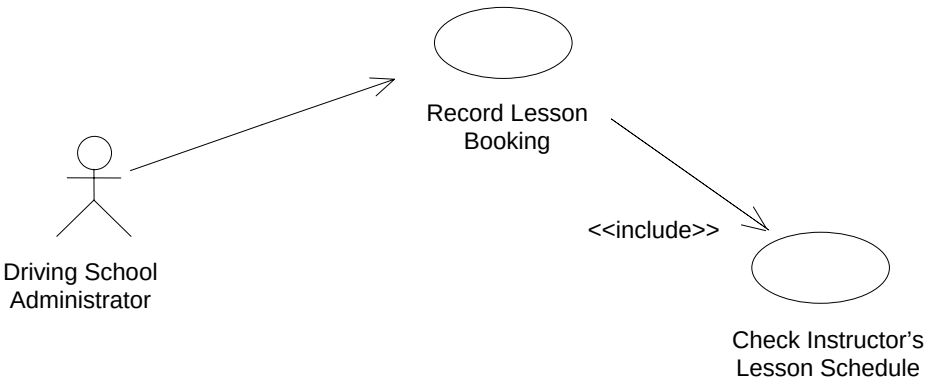
Question 1

1. *a)* Write a BRIEF explanation of the purpose of TWO of the following UML diagrams as used in Object-Oriented systems design. You should illustrate your answer with a labelled diagram:

i) ***design*** class diagram,
ii) sequence diagram,
iii) statechart (state diagram).

(2 x 5 marks)

b) The following fragment of a Use Case diagram and entry from the Use Case Catalogue **(Figure 1)** are taken from a driving school system. Draw a labelled Sequence Diagram for the Use Case *Record Lesson Booking* to illustrate how objects will interact during the execution of the Use Case. **(15 marks)**



Use Case: *Record Lesson Booking*
Primary Actor: *Driving School Administrator*
Goal: *To facilitate the recording of a lesson booking for an existing pupil with their allocated instructor*
Preconditions: *Instructor is available for lesson and pupil is already registered*
Success Condition: *A specific driving lesson is recorded for a specific registered pupil with their allocated instructor.*

Main Sequence:

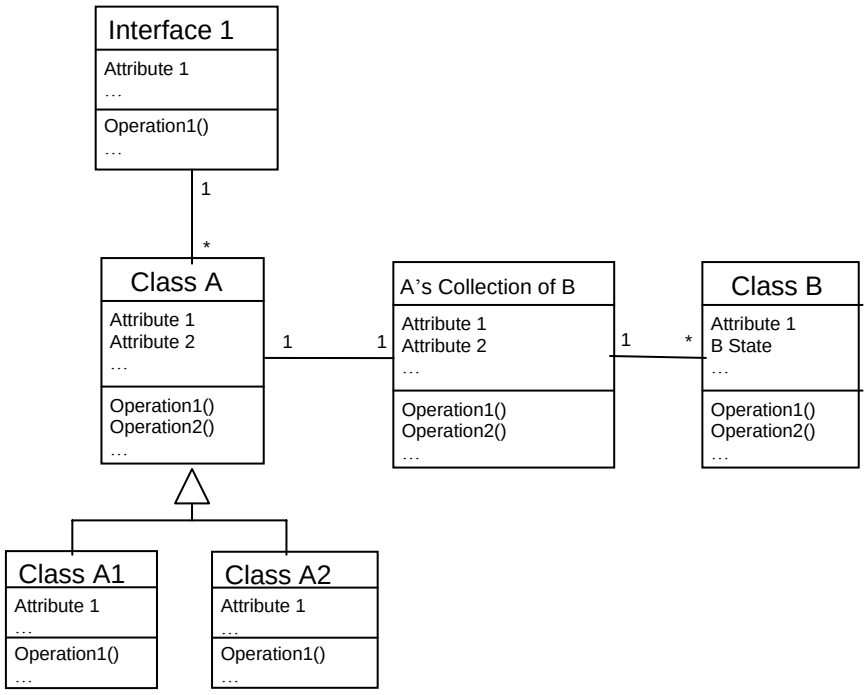
1. The driving school administrator inputs the pupil's driver number
2. The screen displays the pupil's details and presents the name of their instructor
3. The driving school administrator checks the instructor's lesson schedule for a suitable date and time
4. The system displays the lesson slot time and date and confirms instructor's availability for the lesson
5. The driving school administrator approves lesson booking
6. The system displays a confirmation that the booking is made and updates the instructor's schedule

Figure 1

Answer Pointers

(a) i) Design Class Diagram

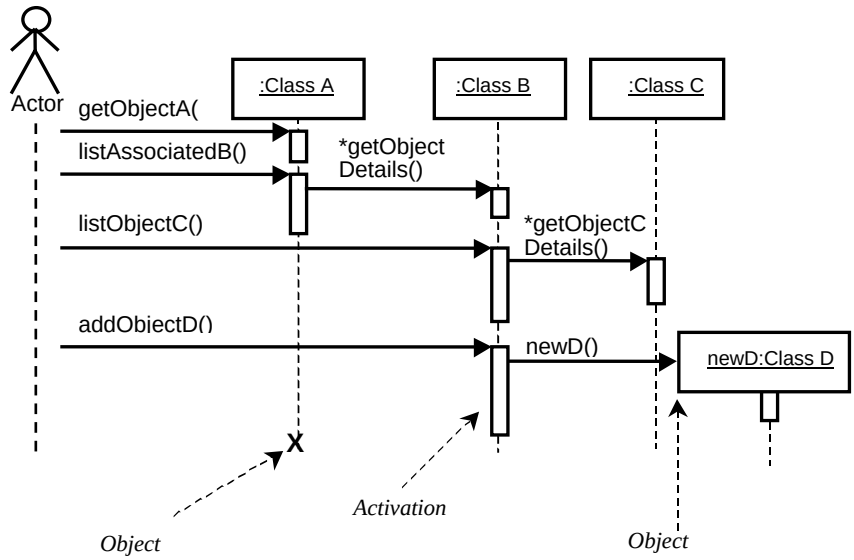
- static view of the system design
- contains operations that provide functionality
- may contain attributes to implement design e.g. object state
- contains business classes but also design classes
    - o interface classes, collection classes, persistence classes
- may contain any design patterns that have been applied



(marked on merit depending on points made and diagram provided - max 5 marks)

Sequence Diagram

- dynamic view of the system
- drawn for a single use case
- shows interactions (message passing) between objects

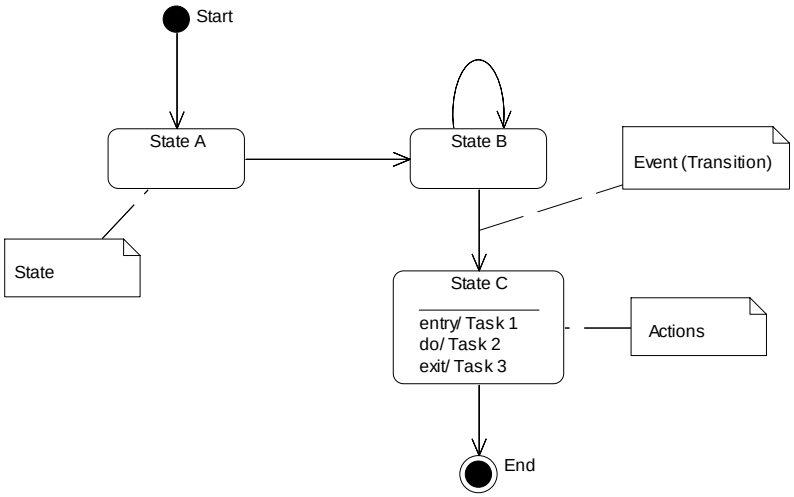


- passage of time represented in the order that messages are passed

(marked on merit depending on points made and diagram provided - max 5 marks)

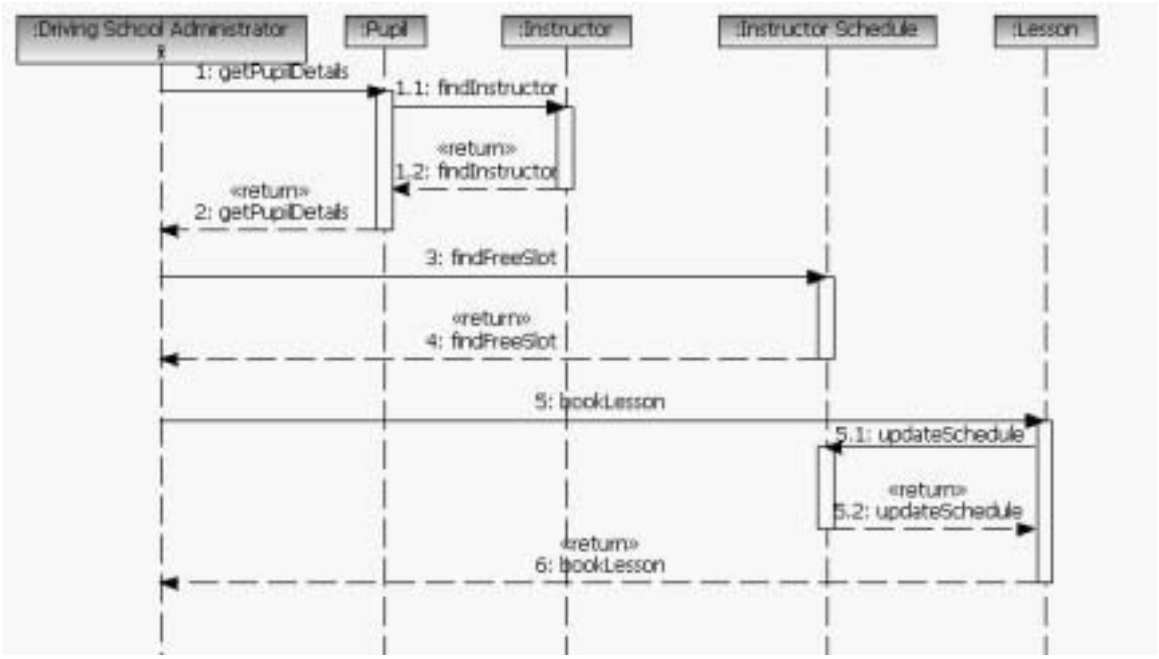
(i) Statechart Diagram

- dynamic view of the system
- is drawn for any single class of objects with non trivial behaviour
- shows all events affecting that class & all its possible states
- may show actions that take place when an object is in a particular state
- state is determined by the value of an objects attributes and its associations



(marked on merit depending on points made and diagram provided – max 5 marks)

(b)



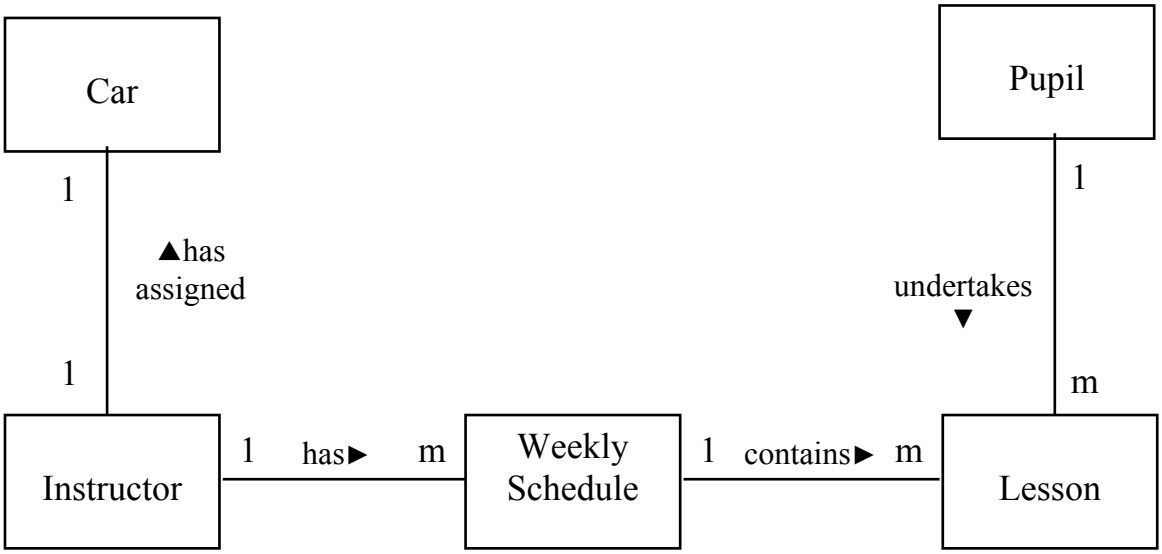
Identification of classes = 4 marks (suitable alternative class names accepted)
Identification of messages = 5 marks (suitable alternative message names accepted)
Logic of interactions= 3 marks
Labelling of diagram = 3 marks (total 15 marks)

Examiner's Comments
This question was very popular with 89% of candidates attempting it. Of those who did it, approximately 70% achieved a pass mark for it and there were some good answers.

Part (a) was a straight forward demonstration of knowledge. In general candidates did not answer the first part of the question well. Only a small number understood the difference between an ***analysis*** class diagram and a ***design*** class diagram. Many candidates described class diagrams in general and focused upon syntax. Few were able to distinguish design objects (e.g. interface, data management, collections) from the analysis (business) objects. Several candidates confused entities with objects and primary keys with object ids (i.e. memory address/reference). Candidates showed a better understanding of the sequence diagram and statechart diagram. However, in some answers there was confusion over a sequence diagram showing ***object*** interactions (as asked for in the question) and a ***system*** sequence diagram showing user interactions.

Part (b) required the demonstration of the candidates' ability to apply knowledge and understanding of Sequence Diagrams to a given Use Case. Several candidates produced a good solution to this demonstrating a good level of understanding. Some candidates, however, were not able to identify all of the required classes for the collaboration and to show suitable message passing among them. Nevertheless, most candidates were awarded marks for the classes and message sequencing that they had identified. In several answers the logic of the interactions was confused. Only a small number of candidates labelled their diagram, as asked to do in the question and subsequently lost marks.

Question 2



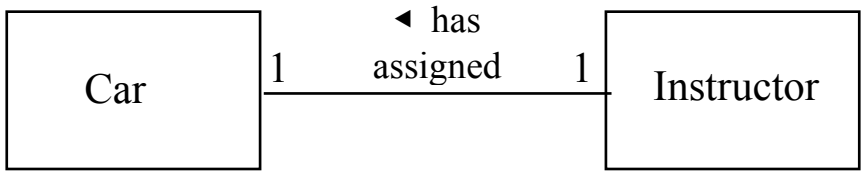
a) Using examples taken from the above ERD and tables, briefly explain the design decisions that should be considered during the physical design of the relational database for each of the following activities:

i) choosing data types
ii) dealing with derived (calculated) attributes
iii) controlling referential integrity
iv) dealing with null values. (4 x 3 marks, total 12 marks)

b) When designing the physical tables it may be necessary to denormalize the given normalized tables in order to optimise for efficient data processing.

i) Explain what is meant by the term denormalization and state two disadvantages of denormalizing tables. **(9 marks)**

ii) Produce the denormalized table(s) for the following one-to-one relationship. **(4 marks)**



CAR (Registration-Number, Make, Model, Mileage)
INSTRUCTOR (Instructor-ID, Instructor-Name, Grade, Registration-Number)

(13 marks)

Answer Pointers

(a) Physical Design decisions

- Choosing data types
  Candidates should discuss such things as: choice of data types that minimise storage space, can represent all possible field values, and allows the appropriate manipulation of the data, allows for business growth and can be validated. Any relevant examples accepted.

- Dealing with derived (calculated) attributes
  Explain what a derived attribute is – mathematically determined by value of other attributes; if stored then more storage space is used, but retrieval is quicker – must be updated when a change made to value of other attributes; if not stored but calculated slower to access but always accurate at that moment in time; required speed of access, therefore, important. E.g. How to deal with Total-Hours in WEEKLY SCHEDULE table.
- Controlling referential integrity
  Referential integrity – value of attribute in one relation must have matching value in another (PK/FK). E.g. may not want to delete a customer record when there are orders outstanding for them. May control this by cascading deletes or restricting deletes. Updates need to be similarly controlled.
- Dealing with null values
  Null values are 'missing' or 'unknown'. Some values cannot be allowed to be null (PKs & FKs must have a value to allow joins – otherwise get Cartesian products). Null values may also affect the results of calculations on certain fields. Any relevant examples accepted.

4 x 3 marks for explanation and suitable example
total 12 marks

(b)

(i)

- Denormalization is the splitting/combining of normalized relations in order to optimize performance of the database. Other suitable explanations accepted.

3 marks

- Can increase the chance of errors & inconsistencies that normalisation avoided
- Improves the performance of some queries at the expense of others
  Any other relevant disadvantages accepted.

2 x3 marks
total 9 marks

(ii) INSTRUCTOR (<u>Instructor-ID</u>, Instructor-Name, Grade, Registration-Number, Make, Model, Mileage)

Table could also be called SALES AREA

(4 marks)
(Total 13 marks)

Examiner's Comments

This question was popular with 66% of candidates attempting it. There was a good pass rate for this question with approximately 84% achieving a pass mark. Again there were some good answers.

Part (a) was generally well answered; however, there were some common mistakes made. Several candidates merely listed an appropriate data type for each of the attributes given. This is ***not*** what is asked for; what was required was the reasoning behind the design decisions considered (i.e. why choose a particular data type, c.f. answer pointers above). There was evidence of some confusion between derived attributes as in this question and the ***inheritance*** of attributes in an object-oriented system. This question is clearly about relational databases. There was also some confusion concerning the handling of null values. Most candidates rightly pointed out that a primary key may not be null, but far fewer spoke of handling legitimate null values (c.f. answer pointers above).

Although Part (b) was answered fairly well by some candidates there was common confusion between un-normalized data (i.e. not even in 1NF) and de-normalized data where flexing of the ERD has taken place in order to optimize the most common queries.

Question 3

3. *a)* Local Area Networks (LANs) are an important feature of today's computer systems within most organizations. With reference to the design of these distributed environments explain what is meant by the following terms:

i) File Server Architecture
ii) Client/Server Architecture.

(2 x 6 marks, total 12 marks)

b) Discuss the advantages and disadvantages of each of the above architectures when choosing between them for a LAN implementation. **(13 marks)**

Answer Pointers

Candidates should base their explanations/discussion around the following.
Other suitable points marked on merit.

(a) i) File Server Architecture

- File server manages file operations and is shared by each client PC attached to a LAN
- Data resides on the file server and the programs on the client
- All processing (data manipulation) takes place on the client
- File server may provide additional resources (printing, email etc.)

ii) Client/Server Architecture

- Application processing is shared between the client and server
- Central (database) server performs commands sent by data manipulation commands sent by client (only result of the request is sent to the client)
- Server controls shared access/recovery etc.
- Client is responsible for managing user interface including presentation of query results

(2 x 6 marks, total 12 marks)

(b) Advantages & Disadvantages depend upon the requirements of the business functions that are using the LAN, however, a comparison of the features upon which to base any choice between the two architectures is outlined below.

File Server	Client/Server
▪ Processing on client only	▪ Processing on both client and server
▪ High network traffic (large file/data transfer)	▪ Low network traffic (efficient data transfer)
▪ Concurrent access low and managed by clients	▪ Concurrent access high and managed by server
▪ Low data integrity/security managed by clients	▪ High data integrity/security managed by server
▪ Low software maintenance most changes just on server	▪ Higher software maintenance changes required on client and server

(13 marks)

Examiner's Comments

This question was less popular with 32% of candidates attempting it. The pass rate for this question was approximately 67%. Answers were variable with some candidates achieving a comfortable pass; however, other candidates scored poorly.

Not all answers distinguished between a file server and client server architectures, indeed some answers were based around the WWW and internet usage as opposed to the LAN hardware/software architectural issues outlined above. Most marks were, therefore, lost by candidates not answering the question asked. Additionally, there was a certain amount of repetition in section (b) of the points made in section (a) by some candidates.

Question 4

4. It is common to find a single computer application has been developed using several different languages.

a) Write notes describing the different features and roles of each of the following languages and their contribution to the development of a single computer application:

i) HTML
ii) C++
iii) XML
iv) SQL
v) Java

(5 x 4 marks, total 20 marks)

b) Explain why you might need a mixture of the above languages for the development of a single application.

(5 marks)

Answer Pointers

a)

i) HTML
Original language for web pages, derived from SGML
Interpreted by browsers (IE and Netscape) to display web pages.
Static definition; for dynamic content we need e.g. Java or Javascript.
Markup language, not programming language
Can be limited and cumbersome to write from scratch
Limitations overcome by HTML generators
(both design-time and run-time)

ii) C++
Full programming language; OO derivative of C
Powerful but not very user friendly, requires skilled programmers
Risky for large systems
Very widely used
Compiled, so platform dependent and many different compilers exist

iii) XML
Interface language for holding structured data in simple text file
to be passed between different applications.
Based on SGML, Originally designed for large-scale electronic publishing.
Easily read, edited and processed (hence should be easy to debug)

iv) SQL
Stands for Structured Query Language
Simple but powerful, english based, non-procedural language for data definition and manipulation in relational database systems.
Standard for many different databases (though varies slightly!)

v) Java
Modern object oriented language for web and distributed systems.
Reliable and secure; interpreted but high performance.
Can be used in Java applets etc. for web page animation.

b) e.g. HTML for a web page layout (static), Javascript to add dynamic content, Java or C++ for server side, SQL to query database, and XML to pass structured information between linked applications.

Examiner's Comments

This was a very popular question, attempted by 84% of those who sat the paper, most of whom managed quite a good answer. This has been a common style of question for some years, and candidates probably felt comfortable with it.

In the first part, most candidates were able to show a reasonable understanding of the major features of the languages given, and the differences between them. However there were still a number of candidates who gave excessively brief answers, not saying much more than "this is a programming language" which gains no marks at all.

The final part proved a bit more of a challenge, with only a few answers containing convincing explanations of why several different languages would be required to implement a single application.

Some candidates had difficulty understanding the significance of the phrase "single application", and merely stated that a single application could be implemented in such and such a language.

Question 5

5. *a)* Describe the main features of Rapid Application Development (RAD), with particular reference to the design phase. **(10 marks)**

b) List five advantages and five disadvantages of the RAD approach. **(10 marks)**

c) What design strategies could you employ to overcome the disadvantages you have described? **(5 marks)**

Answer Pointers

a) RAD **Features**

General strategy rather than methodology
Traditional phases combined and accelerated
Lifecycle consists of: Planning, Design, Development, Cutover
Analysis and Design merged
Iterative around Design/Development
Heavy user involvement throughout (incorporates JAD)
Use of prototyping
Use of tools to speed design/development

b) **Advantages**

Can give time and cost savings, higher quality, better performance
Tighter fit between user requirements and system spec.
Copes better with changing requirements
Users feel more involvement & ownership
Can deliver better usability

Disadvantages

Can reduce quality
Risk of losing alignment with business
Neglecting standards, reuse, scalability, systems admin

High cost of user involvement

c) Design Strategies

Standards (programming, naming etc) should be established early
Make provision for System Admin.
Ensure Human Factors and other non Functional requirements not forgotten
(e.g. consistency, extendability, scalability etc.)
Take care to focus and optimise user involvement, and limit number of iterations.

Examiner's Comments

This was a relatively unpopular question, only attempted by just over half the candidates (57%). The first part proved rather difficult for them. They knew about RAD but did not understand its main features and got it mixed up with other methods.

Interestingly enough, most managed a good answer to the advantages of RAD, though fewer could name many of the potential problems and disadvantages. This is a common occurrence and proves the limit of learning by rote and the lack of real understanding of RAD.

The final part on design strategies to overcome the disadvantages proved the hardest. Many answers suggested a strategy to overcome the disadvantages by using not RAD but a different method altogether.

Question 6

6. A company Help Desk operates for its computer users as follows:
A user has a need or a problem: for example they have forgotten their password, or their mouse has stopped working, or they want a new report produced by the company financial software. They ring the Help Desk, and describe what they need. The Help Desk operator assigns the request to an appropriate team of technicians, one of whom then deals with the request.

The company has decided to install a new Help Desk software system to automate and administer this process, receiving requests direct from the users.

The following information will be held once a request has been received by the help desk:
Category of request (from dropdown)
Title (one line) and Summary (a few lines describing what is being requested)
Name, Userid, Email address of user,
Date submitted, Date required, Priority or severity
Hardware ID or Software version,
Attachments (eg screen dump or exact specification of requirement),
Name of team and individual assigned to work on the problem.

List the functions required of the system by each of the following:

a) The End User **(5 marks)**
b) The Help Desk Manager **(5 marks)**
c) The Technician assigned the problem **(6 marks)**

AND discuss the interface style appropriate to each. **(3 x 3 marks)**

Answer Pointers

The aim here is to get the candidates thinking about different sorts of information to be held by a system involving different user roles. The measure of success is that they understand, and have thought about the situation, and come up with something that makes sense, rather than arrived at a particular answer.

a & d) The End User

Functions: Create new request, Submit request, List own requests, Check request status, Re-raise request (e.g. mouse still faulty) Add notes to a request, Abandon request (= no longer needed)

Interface: The end user will generally only use the system occasionally, so the interface should be simple and helpful, and guide the user through the process. There should not be too many options as this may be confusing.

b & d)The Help Desk Manager

Functions: View list of all outstanding requests, assign request to particular team or technician, generate statistics report, change priority.

Interface: The manager will want a high level view of things, to see a quick summary of workload and backlog, and how SLA (Service Level Agreement) is being met. A fairly simple interface would be best, but using the system is an integral part of the manager's job.

c & d)The Technician

Functions: View list of requests, Accept request, Change Request status Abandon request, Complete request, Pass to someone else (e.g.another team or another technician), Estimate work effort,

Interface: The technician will typically be a frequent user, and will want to get as much information as possible from the system. The interface should allow the technician a high level of control and flexibility.

Examiner's Comments

This question was attempted by nearly 70% of candidates, but with only limited success, less than half achieving a pass mark for the question. For many this was the last question attempted, so time constraints may have played a part in this.

Listing the functions of the system as required by the different classes of user proved rather a challenge, and many simply stated a list of fields, repeating the information given in the question. Some however did manage to see that the different users would have different reasons for accessing the system, and be doing so in different circumstances.

Discussing the interface style proved the hardest part. Some repeated the list of fields from earlier in the question, though a fair number had realised that the end user would be an infrequent user, and would need a more guiding interface