BCS HIGHER EDUCATION QUALIFCATIONS
BCS level 5 Diploma in IT

Systems Analysis and Design MARCH 2018

Examiners' Report

Section A

Case Study for both Sections A and B – ABC Coaches

ABC Coaches is a coach company based near London. They specialise in organising day trips to various destinations in England. Customers of ABC Coaches include institutions such as schools, nursing homes etc. They hire coaches with drivers for trips which are organised/arranged especially for them.

The manager of ABC is responsible for allocation of coaches and drivers to trips. Trip records are created when trips are arranged. If a customer (for whom a trip is being arranged) is 'new' then the customer's details are recorded. Otherwise, the customer's record is updated.

Customers will typically request that a day trip be organised for them on a specific date. The number of coaches allocated to a trip depends on the number of seats required. In response to this request ABC will check to see if the required coaches can be made available on that date. If the coaches are available ABC will allocate one or more drivers and create a trip record for the customer. Customers are allowed to cancel a trip before a deposit is paid. The deposit should be paid within 7 days of the booking for the trip being taken. If a trip is cancelled after that, the deposit is kept by ABC Coaches. If a trip is cancelled the trip record will record this. ABC will request full payment for a trip in the week before it takes place.

A cancelled trip is deleted from the system 6 months after cancellation. Other trip records are deleted 12 months after the corresponding trips were completed.

Section A

General Comments

A reasonable range of marks for this section with most candidates obtaining a pass. None of the three questions attracted consistently low or high marks. Question A1 and A2 were equally popular being attempted by 53 and 51 candidates respectively. Question A3 was less popular being attempted by 41 candidates.

Question A1 Syllabus References

1.3 The characteristics and purpose of systems analysis and design methods and methodologies…

2.7 Business activity modelling, including the use of data flow diagrams (DFDs).

3.1 Use cases and scenarios

Question A1

a) Using information from the case study description above produce a logical top level data flow diagram for a Booking System for ABC Coaches system.

(12 marks)

b) Using information from the case study above create a Use Case diagram for ABC Coaches.

(7 marks)

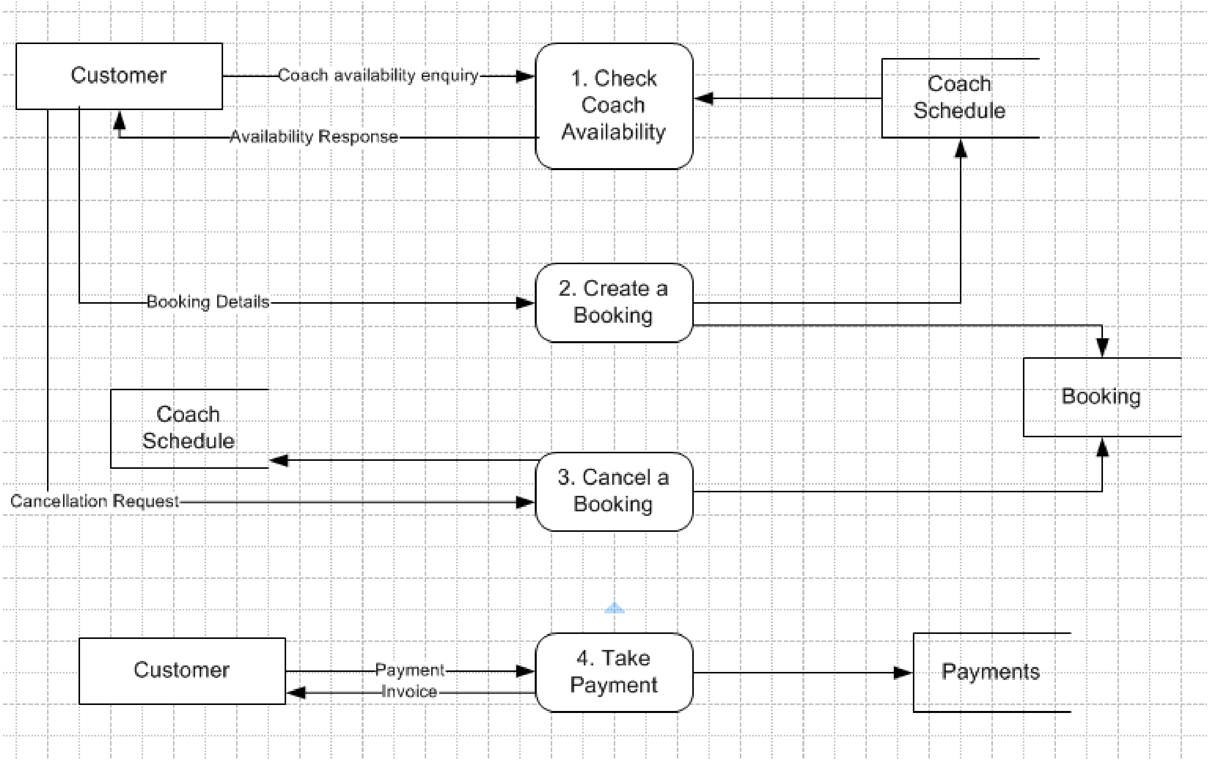
c) Briefly explain the difference between a high-level DFD and a Use Case diagram.

(6 marks)

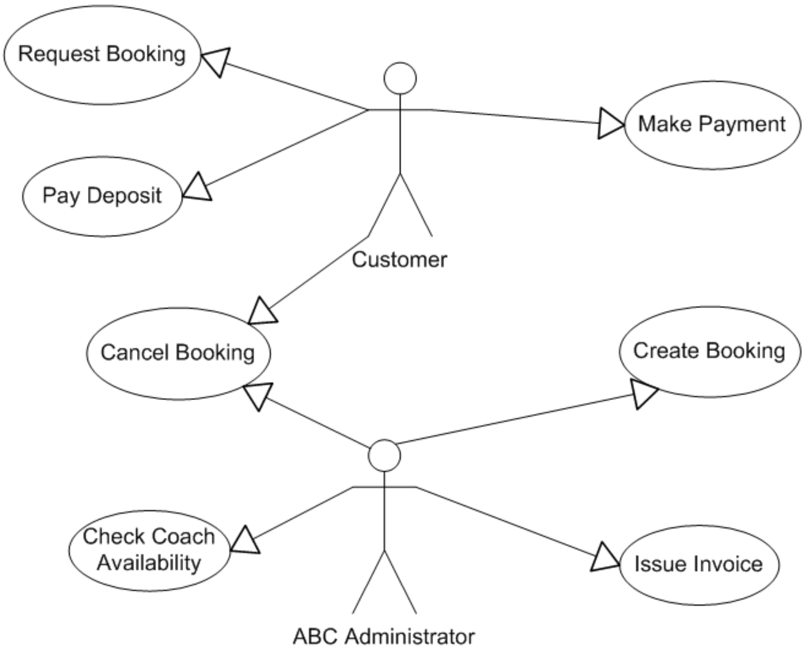
Answer Pointers

a) Up to 6 marks for reasonable identification of top-level processes. Up to 4 marks for dataflows. 2 marks for datastores.

An example DFD for the case study is below. Total 12 marks



b) Up to 6 marks for Use Cases, 1 mark for actors and relationships.



c) DFDs are concerned with the flow of information/data and processes in the system working with this information/data. It helps in identifying any business processes or existing business processes. The emphasis is on data flow and how it will be used in system including where the data will be saved. Something could be said about the hierarchical structure of these diagrams. (3 marks)

Use cases are concerned with defining what the system has to offer to its users. They map to goals the user has that the software system will help them to achieve. Something could be said about Use Case elaboration where the details of a Use Case are represented in a textual description of the interactions between the User and the software. 3 marks. (Total 6 marks).

Examiners' Guidance Notes

This question was attempted by 66% of candidates and attracted an average mark of 15/25. There is evidence that most answers for part (a) were good although a minority of candidates made the mistake of representing some dataflows as processes by connecting process boxes that look more like data stores. Answers to Part (b) and (c) were generally good.

Question A2 Syllabus References

1.1 Systems development lifecycle and position of SAD within it;

1.3 The characteristics and purpose of systems analysis and design methods and methodologies (including agile approaches such as Atern/DSDM Dynamic Systems Design Method, and XP eXtreme Programming);

Question A2

a) The following are four phases in the Systems Development Life Cycle (SDLC). For each phase identify TWO deliverables, briefly describe each of these deliverables and identify the techniques used to produce it.

i) Requirements identification **(5 marks)**

ii) Analysis **(5 marks)**

iii) Design **(5 marks)**

iv) Implementation **(5 marks)**

b) Briefly describe the waterfall method of systems development and briefly explain why this method is less popular now than it used to be. **(5 marks)**

Answer Pointers

a)

i) Requirements identification

The requirements phase is concerned with deciding why an information system should be built and determining how the project team will go about building it (1 mark). The main deliverables for this are:

- a system request (providing a brief summary of a business need and how a system addressing that need would provide business value) (2 marks),
- a feasibility analysis which examines key aspects of the proposed project (e.g. can we build it? Will it provide business value? Will it be used?) (2 marks).

These are the two main deliverables examinees are likely to identify. Alternatively they might suggest a work plan and/or a project plan. These would also be acceptable. Total 5 marks.

ii) Analysis

The analysis phase answers the questions of who will use the system, what the system will do and where and when it will be used (1 marks). The key deliverables for this stage are an analysis based on requirements gathering (2 marks) and a high-level initial design – or systems proposal - for the new system (2 marks)

In addition to naming these deliverables the examinee needs to say a little about ways in which requirements might be gathered and documented and how the high-level design will be communicated. Total 5 marks.

iii) Design

The design phase determines how the system will operate in terms of the hardware, software and network infrastructure that will be in place (1 mark). There are therefore a number of deliverables that might be associated with this phase. Acceptable answers would be: architecture design, interface design, database design or program design. For each of these the examinee would need to say a little about how they are developed – 2 marks for each of two identified and discussed. Total 5 marks.

iv) Implementation

This is the stage in which the software is actually built (1 mark). Again a number of deliverables could be discussed. Acceptable answers would include: the software system itself along with documentation, a training plan or a support plan. For each of these the examinee would need to say a little about how they are developed – 2 marks for each of two identified and discussed. Total 5 marks.

b) The key idea that must be communicated is that the waterfall method proceeds sequentially from one phase to the next (1 mark). Once the work in one phase is approved the phase ends and the next phase begins (1 mark). It is quite difficult to go back from one phase to its predecessor (1 mark). The approach has fallen out of favour as software tools have evolved and a greater importance has been given to prototyping within an iterative lifecycle (1 mark) – where earlier stages in the lifecycle are revisited in response to the user interacting with a partially implemented system (1 mark). Total 5 marks

Examiners' Guidance Notes

This question was attempted by 64% of candidates and attracted an average mark of 13/25. The evidence shows that several candidates did not focus on deliverables in the answer to part (a) and focussed instead on a description of the different stages of the SDLC – sometimes without mentioning any of the deliverables. Part (b) was generally well answered – most candidates were familiar with the waterfall model and its weaknesses.

Question A3 Syllabus References

1.3 The characteristics and purpose of systems analysis and design methods and methodologies….

1.4 Adaptation of methodologies to deal with the particular circumstances of a development or application environment, including adoption/adaptation of existing software solutions.

2.1 Stakeholder analysis

2.8 Use of prototyping, particularly as a method of requirements elicitation

Question A3

a) What is the difference between systems prototyping and throwaway prototyping methodologies? Give examples of the application of each approach **(10 marks)**

b) Identify THREE different ways of involving users in a development project. Comment on any problems that might arise with each of these. **(15 marks)**

Answer Pointers

a) The key point that must be made is that systems prototyping generally leads to a functional system (3 marks); while throwaway prototyping generally leads to understanding the user requirements (3 marks) and design considerations more quickly. 4 marks for examples of each type prototyping. Total 10 marks.

b) Users can be involved in the systems development process in the roles shown in the table below – (5 marks for each to a maximum of 15).

Role	Problems
Full team member	Can lose sight of the user perspective but this can be overcome by rotating the team membership.
Consultative and Review	No direct influence on the design of the new system.
Participant in fact finding as interviewee only	Lacks sense of ownership for the new system.

(15 marks)

Examiners' Guidance Notes

This was the least popular question in section A being attempted by 51% of candidates. The average mark obtained for this question was 13/25. There is evidence that most candidates made a good job of part (a). Several candidates interpreted Part (b) in an open-ended way with answers ranging from diagramming techniques that help users understand the architecture of the system to advice on how to run Application Development Workshops. This range of answers seemed reasonable for a question of this type which gave candidates an opportunity to focus on aspects of the course that they were most comfortable with.

Section B

General Comments

Questions B4 and B5 were much more popular than Question B6. The best results were achieved for Question B4. The Question B5 and Question B6 results were substantially worse.

Question B4 Syllabus References

4.1 Conversion of ERM to a relational schema
4.2 Normalisation and denormalisation
3.4 Entity relationship modelling (ERM)

Question B4

This question refers to the case study described above – ABC Coaches. The table below shows an example of a list of coaches, trips to which the coaches are allocated and the corresponding customers.

Coach No.: DV16PPS	No of seats: 50				
	Trip No: 23/17	Trip date: 6/5/17	Trip destination: Windsor	Cust No: 12	Cust name: Sunrise Home
	Trip No: 35/17	Trip date: 4/6/17	Trip destination: Brighton	Cust No: 19	Cust name: Ethelburga School
		.....	.....		
Coach No.: RS15VVX	No of seats: 56				
	Trip No: 28/17	Trip date: 18/5/17	Trip destination: Oxford	Cust No: 23	Cust name: Sutton High
		.....	....		
Coach No.: TD16BBD	No of seats: 62				
	Trip No: 11/17	Trip date: 2/3/17	Trip destination: Windsor	Cust No: 8	Cust name: Pines Home
	Trip No.: 36/17	Trip date: 5/6/17	Trip destination: Hastings	Cust No: 14	Cust name: Croydon High
		……	…….		

a) Normalise the table to produce a set of relations in the Third Normal Form. You must show all of your working explaining each step.

18 marks

b) Draw an entity relationship diagram (ERD) based on the relations produced in part a).

7 marks

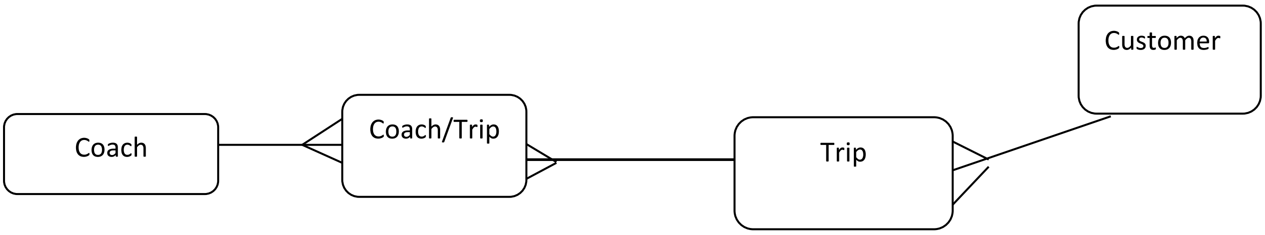
Answer Pointer

a) The steps of normalisation are shown below:

UNF	1NF	2NF	3NF	Relations
Coach No No of seats	Coach No No of seats	Coach No No of seats	Coach No No of seats	Coach
Trip No Trip date Trip destination Cust No Cust name	Coach No Trip No Trip date Trip destination Cust No Cust name	Coach No Trip No	Coach no Trip No	Coach/Trip
		Trip No Trip date Trip destination Cust No Cust name	Trip No Trip date Trip destination Cust No*	Trip
			Cust No Cust name	Customer

For correct 1NF with explanation (remove repeating groups)	5 marks
For correct 2NF with explanation (remove part key dependencies)	5 marks
For correct 3NF with explanation (remove non-key dependencies)	5 marks
For correct relations	3 marks
	Total 18 marks

b)



For correct relationships and their multiplicities	5 marks
For correct entities	2 marks
	Total 7 marks

Examiners' Guidance Notes

Nearly all candidates attempted this question and the majority of them achieved a pass mark for their answers. There is evidence that many answers for part (a) were very good and the majority of candidates were able to practically demonstrate the normalisation process. Some candidates however did not provide proper explanations. Part (b) was answered generally well. Some candidates had problems with relationships (with cardinalities of relationships in particular). Some candidates produced ERDs which were inconsistent with the normalised relations/tables.

Question Number B5 Syllabus References

5.1 OO concepts: classes and objects; encapsulation, interfaces, inheritance, polymorphism, message passing

5.2 Relating objects; associations and aggregations

Question B5

a) Consider the following extra information about the ABC Coaches system described above:

“There are two types of coach drivers: full time drivers and part time drivers. The following data should be stored about each full time driver*: Driver name, Date of birth, Contact details, Salary*. The attributes of each part time driver are*: Driver name, Date of birth, Contact details, Hourly rate, Hours worked.*
All drivers are required to submit their CVs. A CV consists of a header, a number of CV lines, a driver’s signature.”

Explain the following relationships between classes using examples from the Golden Racquet system to illustrate your answers:

(i) Association,
(ii) Aggregation or Composition, and
(iii) Generalisation/Inheritance.

The examples should show relevant fragments of a class diagram.

15 marks

b) There are many characteristics/attributes of a good software design. One of them is usability. Explain the meaning of usability and give five examples of poor software system usability. **10 marks**

Answer Pointers

a) Explanation of association 2 marks
Example of association (e.g. between Coach and Driver) 3 marks

Explanation of aggregation 2 marks
Example of aggregation (e.g. object of class CV ‘consists’ of Header, iteration of CV lines, Signature) 3 marks

Explanation of inheritance/generalization 2 marks
Example of inheritance/generalization (e.g. Driver – superclass with two subclasses: Full time and Part time) 3 marks

For each example a proper fragment of a class diagram should be produced.

Total 15 marks

b) One of the definitions of usability states: 'The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.' In software industry the main aspects of usable software are: ease of use, ease of learning, satisfying user requirements, efficiency, etc.

All main aspects of usability should be briefly explained 5 marks

Examples: poor interface design, inappropriate data entry, incomprehensible error messages, some functional requirements are not provided/implemented, poor response time. 5 marks

Total 10 marks

Examiners' Guidance Notes

This question was attempted by approximately 83% of candidates. The evidence shows that part (a) was answered reasonably well. Some candidates however were unable to give proper and correct examples of relationships between classes. A small number of candidates also had problems with definitions/explanations of relationships between classes (association and aggregation in particular). Also a small number of candidates did not draw relevant fragments of class diagrams.

Part (b): Some candidates provided reasonable answers, but there were also some irrelevant answers e.g. some candidates discussed different software quality attributes – such as reusability, flexibility, etc. Many candidates did not give proper examples of poor software usability.

Question B6 Syllabus references

5.3 Static modelling, including UML class diagrams;

5.4 Dynamic modelling: including UML interaction diagrams (e.g. sequence, communication/collaboration diagrams) UML statecharts

Question B6

a) UML statecharts/state machines and activity diagrams are based on a similar notation. They have however completely different meaning. Discuss the main differences between these diagrams.

(5 marks)

b) Produce a sequence diagram for the use case 'Allocate coaches to trip' in the ABC Coaches system described above. A brief description of this use case is given below.

" A manager enters the trip number and the system displays the trip details. Next the system displays a list of all available coaches. The manager selects one (or more) coach(es) and the system allocates this (these) coaches to the trip and displays the confirmation message".

(8 marks)

c) Produce a state machine/chart for the class Trip in the ABC Coaches system described above. **(12 marks)**

Answer Pointers

a) A brief explanation of both diagrams
Differences

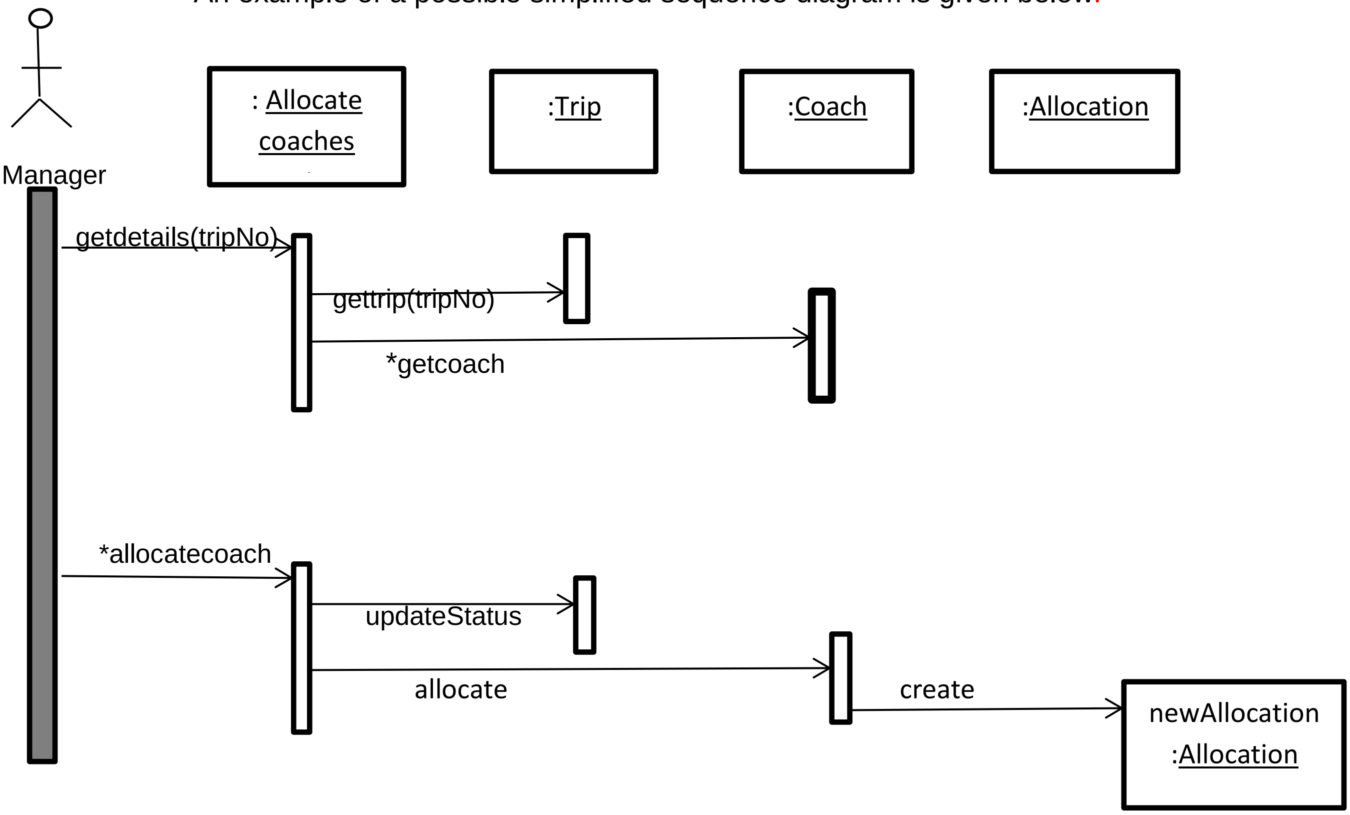
Total 5 marks

b) The sequence diagram should have the following elements:

Actor – Manager	1 mark
Classes/Objects: Trip, Coach, possibly Allocation.	3 marks
Right messages/operations	3 marks
Loops (to show lists displayed by the system)	1 mark

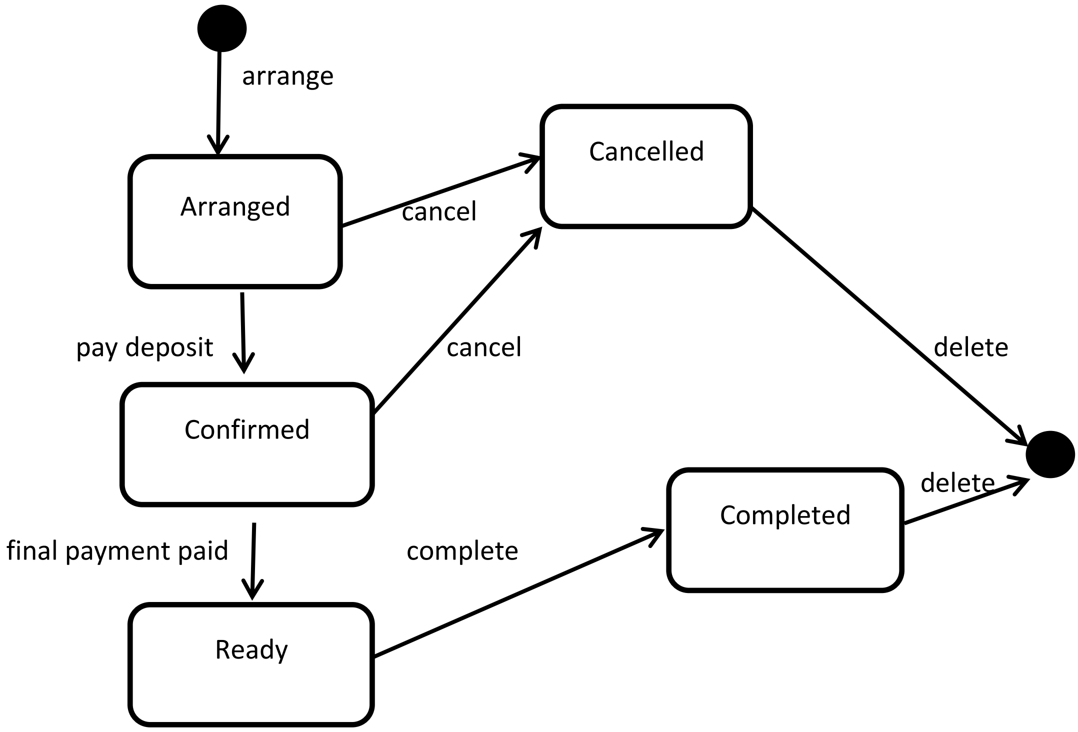
8 marks

An example of a possible simplified sequence diagram is given below.



N.B. Allocatecoaches object combines both the control object and the boundary object for this use case.
*getcoach and *allocatecoach represent iterations of messages/operations

c)



correct state nodes	5 marks
initial and final states	2 marks
correct transitions	5 marks

Total 12 marks

Examiners' Guidance Notes

Only a small number of candidates (33%) attempted this question. There is evidence that part (a) was answered reasonably well. Most candidates who answered part (b) identified the right actor, but only a few candidates identified the right classes/objects and messages. Part (c): some candidates produced reasonable state charts/state machines, but many candidates produced 'activity' diagrams instead.