Tutorial Letter 103/2/2018

Object-Oriented Analysis
ICT2622

Semester 2

School of Computing

IMPORTANT INFORMATION

Please register on myUnisa, activate your myLife e-mail address and make sure that you have regular access to the myUnisa module website, ICT2622-18-S2, as well as your group website where applicable.

Note: This is an online module and therefore it is available on myUnisa. However, in order to support you in your learning process, you will also receive some study material in printed format.

BARCODE

Define tomorrow.

UNISA | university of south africa

CONTENTS

ASSIGNMENT 2 QUESTIONS

Assignment Administration

Due Date	Refer to your Tutorial Letter 101
Submission Procedure	Electronically via *my*Unisa
Number of Questions	08
Total Marks	100
Contribution to Year Mark	40%
Unique Assignment Number	823980

Assignment Questions

Question 1 **[16]**

1.1 Using your own understanding, briefly discuss six core processes required in the development of any information system. (12)

1.2 What is systems analysis? Why is it important? (4)

Question 2 **[10]**

2.1 What are stakeholders in system development? (2)

2.2 Briefly discuss four types of stakeholders. Give an example for each stakeholder. (8)

Question 3 **[14]**

3.1 What are the seven steps of the event decomposition technique? (14)

Question 4 **[9]**

4.1 Briefly discuss the concept of inheritance. Using a diagram, give an example of inheritance. (5)

4.2 List at least four rules for developing state machine diagrams. (4)

Question 5 **[20]**

Consider the following narrative:

> *It's pay day! Bob, because he has been paid, has decided to go and withdraw some money from an automated teller machine (ATM). The withdrawal transaction starts by Bob inserting his bank card into the ATM. The ATM validates the card, and if it's valid prompts Bob for his personal identity number (PIN). If the card is invalid / unreadable, the ATM ejects the inserted card after displaying an appropriate message on the screen. Once the PIN is entered, the information is sent to Bob's bank to validate whether or not the entered PIN is valid. If the PIN is invalid, the ATM ejects the inserted card after displaying an appropriate message on the screen without prompting Bob for another try. If the PIN is valid, the ATM then prompts Bob for his desired amount. That information is again sent to Bob's bank to ascertain whether or not the requested amount is within his current balance. If the requested amount is more than the current balance, the ATM displays this information on the screen and ejects the inserted card without prompting Bob for another try. If the requested amount is within the current balance, the ATM dispenses the requested amount for Bob to take and the bank adjusts Bob's account accordingly. The latest account balance is then displayed on the ATM screen and the ATM then completes the transaction by ejecting Bob's card. Bob takes his card and runs off to go do some shopping.*

5.1 Develop an activity diagram for Bob's withdrawal transaction.

Question 6 [9]

Consider the following UML design class.

```
┌─────────────────────────────────────┐
│             <<entity>>               │
│                Boat                  │
├─────────────────────────────────────┤
│ - boatID: String {key}              │
│ - manufacturer: String              │
│ - model: String                     │
│ + numberOfBoats: Integer            │
├─────────────────────────────────────┤
│ + createNew(data...)                │
│ + updateManufacturer(data...)       │
│ + updateModel(data...)              │
│ + getBoatInformation(): String      │
│ + getProvince(): String             │
│ + getNumberOfBoats(): Integer       │
│                                      │
└─────────────────────────────────────┘
```
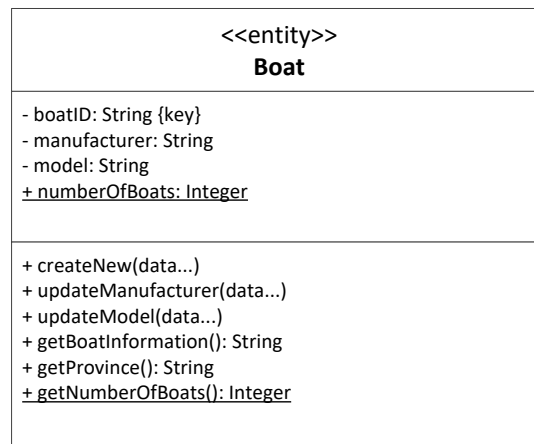
Figure 5.1

6.1 What is denoted by the "+" and "-" signs? (3)

6.2 What is meant by the **<<entity>>** stereotype? (2)

6.3 The *numberOfBoats* attribute is underlined because it's a special type of attribute.

Name and briefly describe this special type of attribute. (2)

6.4 The *getNumberOfBoats* method is underlined because it's a special type of method.

Name and briefly describe this special type of method. (2)

Question 7 [10]

Consider the following scenario:

> *At a university, courses are offered as course sections, and a student enrols in many course sections. Each course section contains many students. Typical attributes of a course are course code, course title, course credits and NQF level; a course section includes section number, start time and room number; typical attributes of a student are student number, student name and major.*

7.1 Draw a simple domain model class diagram for this scenario, using the camelback notation for class and attribute names where necessary.

Question 8 [12]

8.1 Consider the following diagrams: Figure 8.1 shows an abridged use case diagram of a basic address book application; Figure 8.2 shows a class diagram of a typical Person class belonging to this address book application; while Figure 8.3 shows an abridged class diagram of this address book application.
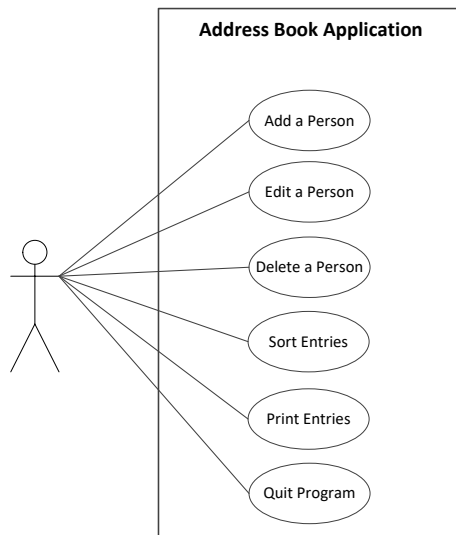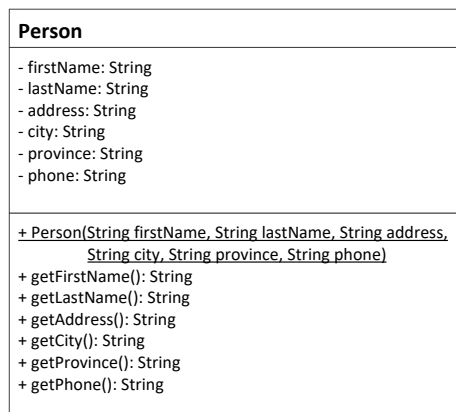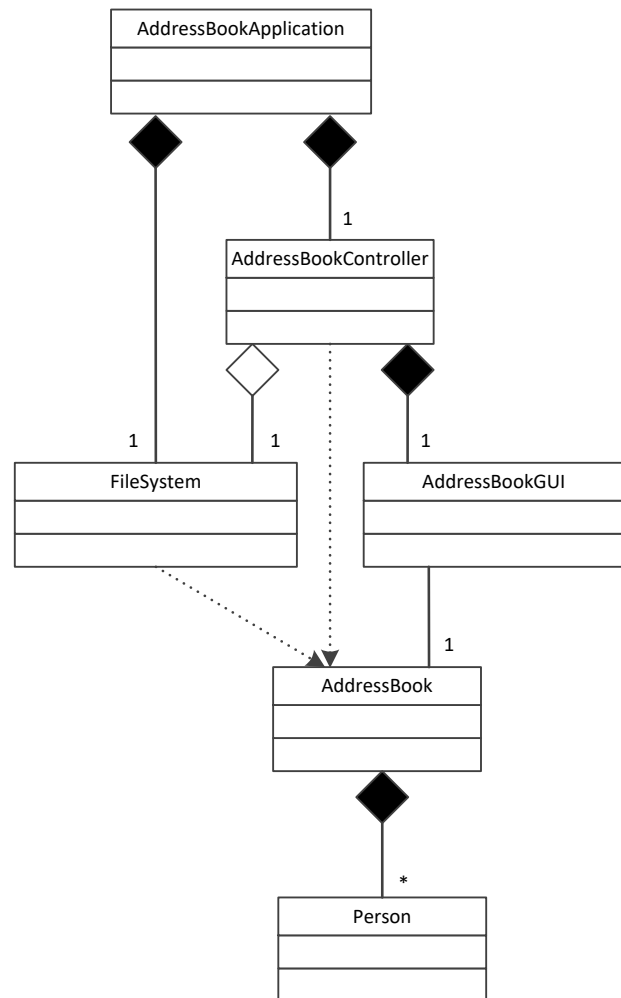
Figure 8.1

Figure 8.2

Figure 8.3

Consider also the following description for the *Print Entries* use case:

> *The Print Entries use case is initiated when a user chooses "Print" from the File menu of the Address Book Application. A Save File dialog is displayed and a user is allowed to choose a file to print the labels to (if the user cancels the Save File dialog, the Print operation is cancelled). The current contents of the address book are written out to the specified file (in their current order). No information maintained by the program is changed.*

Using any or all of the information given above, develop a sequence diagram that shows the *Print Entries* use case.