THE UNIVERSITY OF MELBOURNE
DEPARTMENT OF COMPUTER SCIENCE

Examination – Semester 1, 2013

SWEN30006 Software Modelling and Design

Exam Duration: 2 hours
Total marks for this paper: 120

Reading Time: 15 minutes

This paper has 7 pages including this page.

Authorised materials:
No materials are permitted

Instructions to Invigilators:
Each student should initially receive a script book.
Students may keep the exam paper after the examination.

The exam paper is worth 60% of your final mark Instructions to Students:

- Answer *all* questions in Part A.
- Answer *all* questions from Part B.
- Answer questions on the lined pages in a script book.
- Start your answer to each question on a new page.
- Be sure to clearly number your answers.

The exam consists of 10 questions grouped into two parts. The marks for each question are listed at the beginning of the question. The marks for each question are split to indicate the number of marks for each part. For example a question that has marks (5 + 10 + 5) assigns 5 marks to part (1), 10 marks to part (2) and 5 marks to part (3).
Please use the marks as an indication of the amount of time to spend on each question and as a guide to the detail required in your answers while keeping your answers concise and relevant. Point form is acceptable in answering descriptive questions.

This paper is to be reproduced and lodged with Baillieu Library.

Part A – Short Answer Questions

Students must answer *all* questions in this section.

A Point of Sale System (**POS**) is to be developed for a Hardware retail business. The business is located in a warehouse where customer service personnel sell many types of goods, for example, power tools, timber, paints, paint brushes, hardware, and garden tools. To help the customer service personnel (CSP) the POS stores information about products and allows CSP's to make sales directly from hand held mobile data terminals (MDT). The warehouse uses a wireless network for communication and maintains a database of products and customers. Two scenarios are particularly important.

Scenario 1 - The Enquiry Scenario: The *Customer* asks a CSP about a product item. The CSP may answer the question from their own knowledge and experience but if they cannot then the CSP may read the item's product serial number and use it to access the product database to answer the customer's question.

Scenario 2 - The Sales Scenario: When a customer wants to purchase an item they can go to a customer service person (CSP) instead of a checkout counter. To complete a sale the customer services person reads the product's serial number using the mobile data terminal and the MDT displays the price and any discounts that apply. The CSP can take cash or credit cards. In the case of a credit card the CSP can handles the transaction through the MDT. The customer makes the purchase, the CSP OK's the purchase via the MDT and the product item is *released*. When customer's leave the store with their products a scanner reads the product's serial number. If the items carried by the customer have been released then nothing happens, but if not then an alarm goes of at the store entrance.

Figure 1: A computerised **Point of Sale** (POS) system for a home and hardware store.

Question 1 [4+4 marks]

According to some authors the process of **Design** consists of **Product Design** and **Product Engineering**. Briefly state the *goal* and *activities* involved in:

(1) Product design; and

(2) Product engineering.

Question 2 [4+4 marks]

(1) Briefly explain the difference between a **Domain** model and a **Design** model.

(2) Give a brief descriptive example that shows the difference.

Question 3 [10 marks]

The second scenario described in the *Point of Sale system* in Figure 1 deals with the sale of a product by a *Customer Sales Person* (CSP) using a *Mobile Data Terminal* (MDT).

Give a **use-case description** for the sale of a product specifying at least the **Actors**, **Triggers**, **Pre-Conditions** and **Flow of Events**.

Question 4 [4+8 marks]

This question refers to the Java code in Figure 2

The owners want to extend the range of products to include lighting products, plumbing products, appliances and electrical goods and other types of products over time.

(1) The code in Figure 2 is not **closed** for modification. Explain why the class is not closed for modification.

(2) Using a class diagram, redesign the class so that it is **closed** for modification and **open** for the extensions listed above.

Ensure that you clearly specify attributes, methods and relations in your class diagram.

Question 5 [10 marks]

This question refers to the Point of Sale System described in Figure 1. The data-flow diagram shown in Figure 3 is a model for the two key scenarios in Figure 1. Write down the **Transform Centre** for the data-flow diagram and give the **structured design** resulting from the data-flow diagram.

Question 6 [5 + 5 marks]

This question refers to the Point of Sale system described in Figure 1.

```
public double totalPrice( Product[] purchaseList ) {

    double total = 0.0;

    for (int i=0; i < purchaseList.length; i++)
    {
       switch ( purchaseList[i].kind ) {
            TIMBER : total = total + 1.275 * purchaseList[i].getPrice();
            TOOLS : total = total + 1.45 * purchaseList[i].getPrice();
            PAINT : total = total + 1.275 * purchaseList[i].getPrice();
            HARDWARE : total = total + 1.52 * purchaseList[i].getPrice();
       }
    }
    return total;
}
```

Figure 2: The Java code for totalling a customer's purchases. The section of code is intended to total the customer's list of current purchases. The cost price of an item is retrieved using the product's 'getPrice' method. The total price method sums the purchases taking into account the markup and discounts according to current pricing policy.

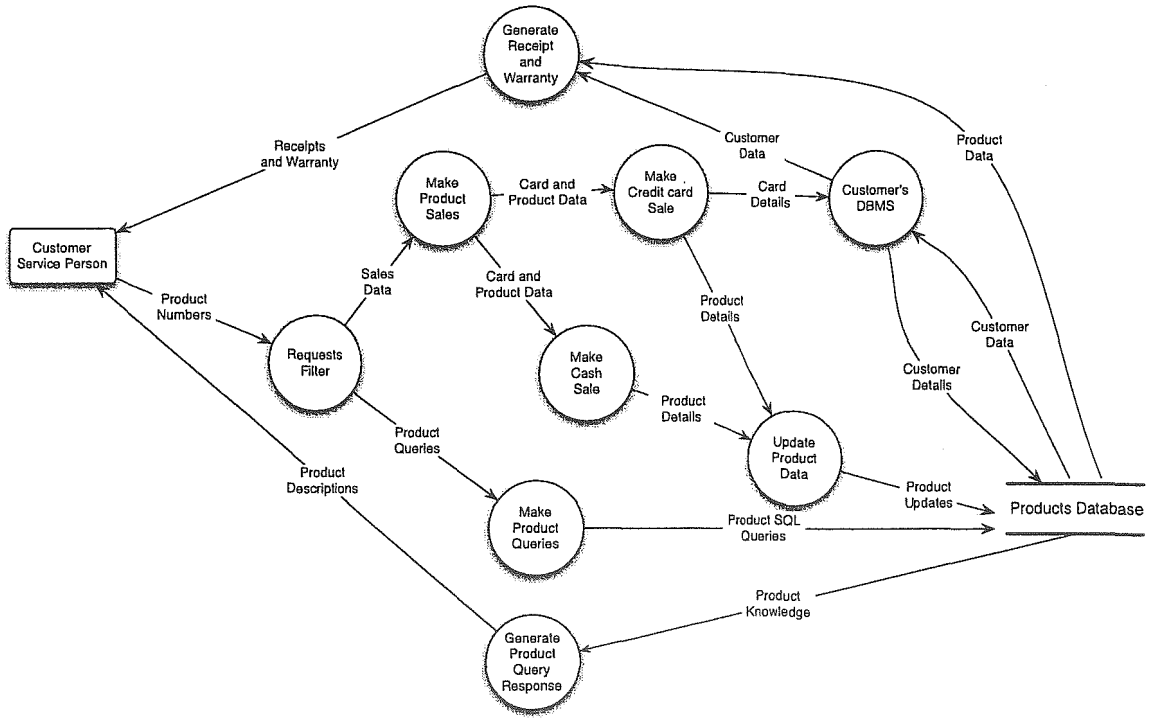

Figure 3: A Data-flow diagram for part of the POS system.

The system engineers wish to use a **Model-View-Controller** architectural pattern for the implementation of the POS system.

(1) Why is a model-view-controller pattern relevant or not relevant for the architecture of the POS system?

(2) Discuss the advantages or disadvantage of the model-view-controller by comparing it to a **Client-Server** architecture for the same problem.

Part B – Long Answer Modelling and Architecture Questions

Students must answer *all* questions in this section.

All questions in this section refer to the POS system in Figure 1

Question 7 **[12 + 8 marks]**

(1) Draw a domain class diagram for the POS system based on the description in Figure 1.

(2) Draw a sequence diagram for the classes participating in the **Sale** use case.

Question 8 **[4 + 16 marks]**

(1) In two or three sentences briefly describe what you could model using a state-chart in POS system and why.

(2) Write a Java class implementing the state diagram for the **Purchasing** behaviour shown Figure 4

Question 9 **[4 + 12 + 6 marks]**

Develop a design for the POS system by:

(1) Selecting a design style or pattern for an overall software architecture. Justify your choice.

(2) Developing the design package or component diagrams, and classes diagrams for your architecture. You need to show the relationships between your classes, but you do not need to show methods or attributes.

(3) What interfaces must exist between your modules. Give one example of one service in one of the interfaces.

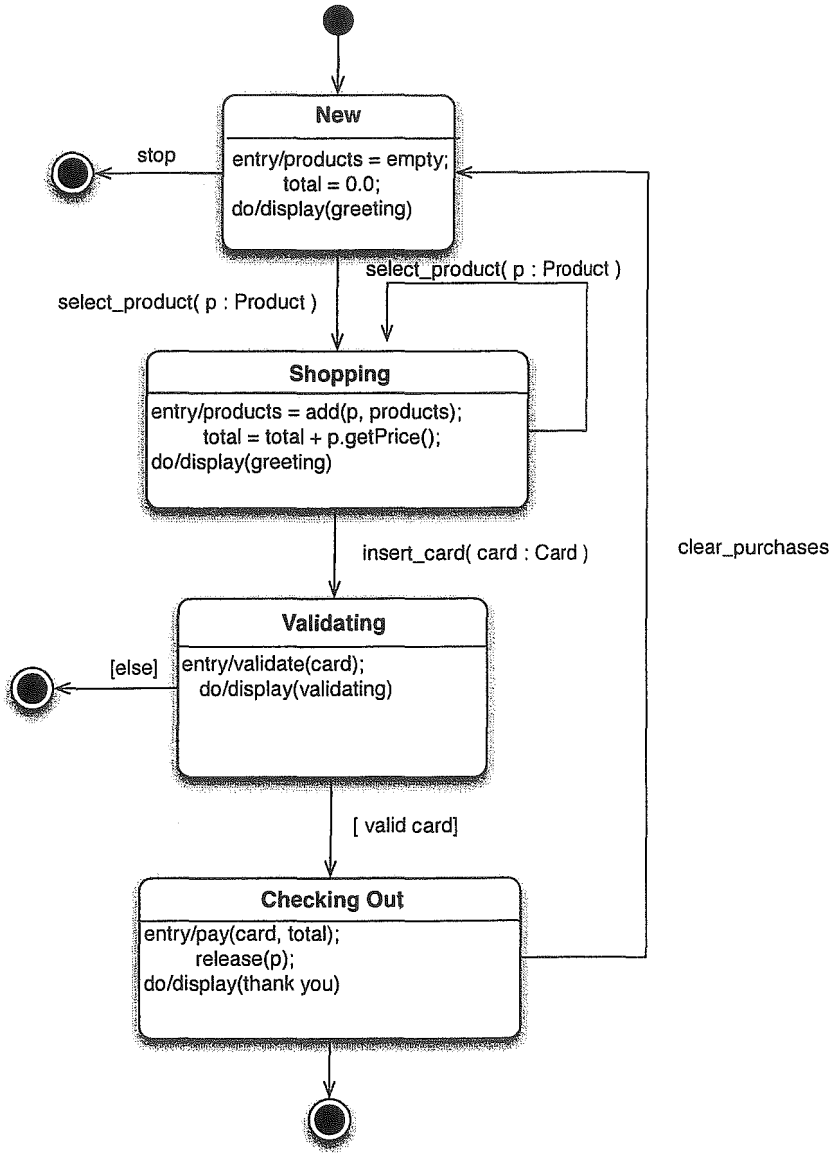

Figure 4: A state-chart for purchasing a list of products in the Point of Sale system described in Figure 1.