NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2018-2019

CE2006/CZ2006 – SOFTWARE ENGINEERING

Apr/May 2019 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 6 pages.
2. Answer **ALL** questions.
3. This is an open-book examination.
4. All questions carry equal marks.
5. Refer to **Appendix A** on page 6 for the project description which is needed to answer the 4 questions.

---

1. (a) During planning for the implementation of the CSMS application project described in Appendix A, the Work Breakdown Structure (WBS) was broken down into eight tasks, Task A to Task H, and dependencies as summarized in Table Q1.

(i) Draw a Network Diagram to represent these tasks and their dependencies.

(3 marks)

(ii) Using forward analysis, identify the critical path and calculate the shortest possible time to complete the CSMS project.
[*Show your workings on how you obtain the critical path and the shortest possible time.*]

(3 marks)

(iii) Using backward analysis, calculate the Slack Time value for Task C.
[*Show your workings on how you obtain the slack time value.*]

(2 marks)

Note: Question No. 1 continues on Page 2

Table Q1

Task	Duration (weeks)	Dependencies (Tasks which must be completed before a task can commence.)
A	3	none
B	10	A
C	8	A
D	11	A
E	1	B, C
F	2	C
G	2	E, F
H	1	D, G

(b) Based on the initial project description given in Appendix A, identify the actors and the main use cases of the CSMS application by drawing a UML Use Case Diagram. Apply generalization, <<include>>, and <<extend>> relationships where appropriate. For <<extend>> relationship, extension points should be clearly indicated.

(10 marks)

(c) Based on the initial project description given in Appendix A, write the use case description for the *compute estimated service waiting time* functionality. Your use case description must include the following elements:

- Actor(s)
- Flow of events
- Alternative flows

[*Provide one alternative flow.*]

(7 marks)

2. (a) From your use case description in Q1(c) for the *compute estimated service waiting time* functionality, identify the main classes and their associations by drawing a conceptual model Class Diagram. Your conceptual model Class Diagram should clearly depict each analysis class as a **Boundary**, **Control**, or **Entity** class. Use analysis class icons to represent Boundary, Control, and Entity classes on the conceptual model Class Diagram. You do not need to identify any attributes or operations within the classes.

(12 marks)

Note: Question No. 2 continues on Page 3

(b) From the main classes identified in Q2(a), draw a UML Communication Diagram to depict how these classes may interact to provide the *compute estimated service waiting time* functionality. Clearly indicate the message flows on the diagram, i.e., the message names, the direction of message flows, and the message number in which the message is sent.

(13 marks)

3. (a) Refer to the classes identified in your answer to Q2(a).

(i) Propose an appropriate architecture for the CSMS application project, and draw a high level architecture diagram.

(8 marks)

(ii) Justify your architecture proposed in Q3(a)(i) by listing THREE (3) benefits of the proposed architecture. State ONE (1) drawback of your proposed choice of architecture.

(4 marks)

(b) Answer the following questions related to design patterns.

(i) What is the difference between push and pull update in observer pattern?

(4 marks)

(ii) Identify one possible design problem in the CSMS system where you can apply observer pattern.

(3 marks)

(iii) Depict the application of the observer pattern proposed in Q3(b)(ii) in a class diagram. Explain briefly the roles each class plays in the proposed design pattern.

(6 marks)

4. (a) The CSMS system calculates the estimated waiting time when a customer comes for servicing based on the following information:

1. Customer number: It is a five-digit integer ranging from 10001 to 89999.
2. Customer priority: One of "Priority 1", "Priority 2", and "Priority 3".
3. Average time to complete a job: The time (in minutes) is an integer value between 10 and 60.

Note: Question No. 4 continues on Page 4

(i) Determine the equivalence classes for the above THREE inputs.

(3 marks)

(ii) Determine the boundaries of the equivalence classes identified in your answers to Q4(a)(i). For each boundary, identify a value on the boundary, a value just below the boundary, and a value just above the boundary.

(4 marks)

(iii) You intend to perform **defensive testing** of the waiting time input interface. Design a set of test cases to test the THREE inputs based on the equivalence classes and boundary values identified in your answers to Q4(a)(i) and Q4(a)(ii).

(5 marks)

(b) The CSMS system automatically calculates the estimated service waiting time price for customers' service requests. The *calculateWaitingTime()* method (as shown in the Java code snippets in Figure Q4(b)) implements this interface.

(i) Draw the control flow graph for the *calculateWaitingTime()* method. [*Use the line numbers for clarity*]

(5 marks)

(ii) Calculate the Cyclomatic Complexity of the *calculateWaitingTime()* method.

(2 marks)

(iii) List the basis set of linearly independent paths for performing basic path testing of the *calculateWaitingTime()* method. Design a set of test cases (including the input parameters to the *calculateWaitingTime()* method, the return value of *getNewCustomerNo()* when necessary, and expected output) for each of the basic paths. Note that you need to provide the return value for the *findCustomerRecord()* method for each test case.

(6 marks)

Note: Question No. 4 continues on Page 5

```
1.  WaitResult calculateWaitingTime(int customerNo)   {
2.      // WaitResult has fields (WaitTime, waitLength, averageJobTime)
3.      WaitResult result = null;
4.      // Customer has fields (number, priority, jobTime)
5.      Customer[] allRequests = getAllRequests();
6.      Customer[] outstandingRequests = getOutstandingRequests();
7.      Customer customer = findCustomerRecord(allRequests, customerNo);
8.      While (customer == null) {
9.          System.out.println(customerNo + " not found; retrieve new
    customerNo.");
10.         customerNo = getNewCustomerNo();
11.         if(customerNo < 0)
12.             throw new Exception("invalid customer!");
13.         else
14.             customer = findCustomerRecord(allRequests, customerNo);
15.     }
16.     int priority = customer.priority;
17.     int waitTime = getWaitingTime(allRequests, outstandingRequests, priority);
18.     int waitLength = outstandingRequests.length;
19.     int averageJobTime = getAverageJobTime(allRequests);
20.     result = new WaitResult(waitTime, waitLength, averageJobTime);
21.     return result;
22. }
```

Figure Q4(b)

Appendix A

Computer Services Management System (CSMS) Description

A software system, Computer Services Management System (CSMS) uses a priority-based scheme (rather than a first-come-first-served scheme) to help a company's staff to estimate the service waiting time for customers' service requests and assign customers' service requests to technicians.

Initial CSMS Project Description

- Each customer has a five-digit customer number and is assigned a priority number (highest priority "3" to lowest priority "1"):

  Priority 3: A customer who has had three or more previous computer services rendered.
  Priority 2: A customer who has had one or two previous computer services rendered.
  Priority 1: A customer requests for computer service for the first time (i.e., new customer).
- Works are first assigned to the technicians for services requested by customers with priority 3, followed by priority 2, and then priority 1.
- The CSMS enables staff to determine the **estimated service waiting time** (i.e., how soon service will be available) for a new service request from a customer as follows:
  - CSMS allows staff to select *compute estimated service waiting time* menu function.
  - CSMS obtains all service requests information from the service request file.
  - The staff inputs the customer's number, upon request from CSMS.
  - CSMS searches for the customer's number from the customer file. If the customer's number is found, CSMS retrieves the customer's priority number from the customer file. If the customer's number is not found, the system displays an error message and asks the staff to re-input the customer number.
  - CSMS computes the estimated service waiting time according to mathematical formulas based on all service requests information, the new request, and the customer's priority number. The *compute estimated service waiting time* function also includes computing additional two functions: *length of current waiting list* (i.e., total number of outstanding requests) and *average time to complete a job*.
  - CSMS displays the *estimated service waiting time*, the *length of current waiting list*, and the *average time to complete a job* information on the screen.
- Once the customer has agreed on the estimated service waiting time information, the CSMS enables staff to add the customer's service request to the waiting list. The customer may cancel the service request so as to not to be put on the waiting list.
- Customers on the waiting list can be sorted by priority number, and by date and time of the request within priority. The priority number of a customer on the waiting list may be altered at any time by the manager, under special circumstances.
- The CSMS enables manager to view a number of specialised reports, namely Waiting List report, Outstanding Job report, and Service Request Assignments report. The manager may request the CSMS to print a selected report.

END OF PAPER

CE2006 SOFTWARE ENGINEERING
CZ2006 SOFTWARE ENGINEERING

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.