NANYANG NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2016-2017

CE2006/CZ2006 – SOFTWARE ENGINEERING

Apr/May 2017 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 5 for the project description which is needed to answer all questions.

---

1. In Appendix A there is an initial description of a basic voicemail system.

(a) Identify the actors and the features of the voicemail system by drawing a UML Use Case Diagram. Use <<include>> and <<extend>> relationships where appropriate.

(12 marks)

(b) Write the Use Case Description(s) for the voicemail system when a registered user leaves a message. Your Use Case Description(s) must include any pre-conditions, post-conditions and the flow of events.

(13 marks)

2. (a) From your Use Case diagram, identify the main Classes and their associations.

(8 marks)

Note: Question No. 2 continues on Page 2

(b) Design a Dialog Map to represent the functionality of the voicemail system. The Dialog Map should illustrate the main user functionalities identified in Q1(a).

(12 marks)

(c) During the planning of the implementation of the voicemail system, you need to break down the project into a Work Breakdown Structure (WBS) and sequence the tasks and their dependencies. List THREE different methods you would use to estimate the time to complete each task. State the name of the diagram and the project management technique that will enable you to determine the shortest and longest time your project will take to complete.

(5 marks)

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

(i) Apply Layered Architecture to model these classes. Indicate the boundary class, control class, and entity class on the layered architecture model.

(6 marks)

(ii) Indicate clearly the dependencies between the classes in the layered architecture model you have applied in Q3(a)(i).

(1 mark)

(iii) List THREE strengths of layered architecture. List ONE weakness of layered architecture.

(4 marks)

(b) Refer to the Voicemail System description in Appendix A on pages 5, the owners can access and administer their mailboxes with three options: local access, remote access, or web or email access (via subscription).

(i) Identify and explain the design problems to implement this requirement.

(4 marks)

(ii) Propose an appropriate design pattern to address the design problems identified in your answers to Q3(b)(i).

(2 marks)

Note: Question No. 3 continues on Page 3

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

(8 marks)

4. (a) The voicemail system allows the mailbox owners to access and administer their mailboxes through three access options: local access, remote access, and web or email access. For local access, owners access their mailbox from their own (local) phone devices via a hotkey on the phone device. The hotkey contains integer values from 100 – 699. An example of a hotkey is 411.

(i) Determine the equivalence classes for local access.

(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.

(3 marks)

(iii) You intend to perform **defensive testing** of the local access option. Design a set of test cases to test the local access option based on the equivalence classes and boundary values identified in your answers to Q4(a)(i) and Q4(a)(ii).

(5 marks)

(b) The voicemail system responds to users' selection from the message menu via phone key pressed by the users. The *messageMenu(String key)* method (as shown in the Java code in Figure Q4(b)) implements this application logic.

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

(6 marks)

(ii) Calculate the Cyclomatic Complexity of the *messageMenu(String key)* method.

(2 marks)

Note: Question No. 4 continues on Page 4

(iii) List the basis set of linearly independent paths for performing basic path testing of the *messageMenu(String key)* method. Design a test case (including the input parameters to the *messageMenu(String key)* method and the expected outcome for each of the basic paths.

(6 marks)

```
/**
   Respond to the user's selection from message menu
   @param key the phone key pressed by the user
      private static final String messageMenu =
            "Enter 1 to listen to the current message\n"
            + "Enter 2 to save the current message\n"
            + "Enter 3 to delete the current message\n"
            + "Enter 4 to return to the main menu";
*/

   private void messageMenu(String key){
      if (key.equals("1")){
         String output = "";
         Message m = currentMailbox.getCurrentMessage();
         if (m == null){
            output += "No messages." + "\n";
         }else{
            output += m.getText() + "\n";
         }
         output += messageMenu;
         thePhone.speak(output);
      }else if(key.equals("2")){
            currentMailbox.saveCurrentMessage();
            thePhone.speak(messageMenu);
      }else if(key.equals("3")){
            currentMailbox.removeCurrentMessage();
            thePhone.speak(messageMenu);
      }else if(key.equals("4")){
            state = MAILBOX_MENU;
            thePhone.speak(mailboxMenu);
      }
      return;
   }
```

Figure Q4(b)

Appendix A

Description of the Voicemail System

The system is a basic **voicemail** system. The voicemail system is an independent system that interfaces with an already implemented telephone system (i.e., assume that the telephone related hardware, software, and API are available and ready to use). The main goal of the system is to act as a messaging service for an organization-wide telephone system. The organization may have several telephone numbers and multiple calls active at a given time.

The voicemail system gets activated after a caller's dialed number to the telephone system goes unanswered for a given number of rings (e.g., five rings). The telephone numbers are uniquely mapped to mailbox numbers. Note that some numbers may not be assigned valid or active mailboxes (or none at all). Assume that the telephone system is responsible for functionalities such as establishing and managing the connection, managing the call (e.g., pickup, hang up), and providing information from and to the caller (e.g., the pressed phone keys, and obtain and play the actual voice message). An example of the interface to the phone system is given below (you may assume that all interface calls are blocking and as such the message system will wait until the interface call has completed execution and a value returned):

```
PhoneSystem

      pickup()  //Initiates one end of a connection

      hangup() //Ends a connection

      dial(char)//Allows dialing of ‘0’..’9’,’#’, ‘*’

      play(message)// Play message

      connection(): Boolean // True if a connection exists

      record(int): message // Record a message until hangup() or time limit is reached

      getRingCount(): int // Returns the number of rings

      getDial(): char//Returns the character dialed (0 if buffer is empty else 1)

      getExtension(): string // Returns the current extension
```

The voicemail system must support the callers to leave voicemails, and the mailbox owners to access and retrieve their voicemails. The owners can access and administer their mailbox with the following options:

(1) Local access: from their own (local) phone devices via a hotkey on the phone device

(2) Remote access: from other (remote) phones via dialing phone and/or extension number

(3) Web or email access: via subscription

The caller can record a message verbally. The message is ended with a call hang up or a timeout in number of seconds. There is also an administration mailbox with extended functionality to manage the email system. This mailbox can access and manage other mailboxes (e.g., add or remove a mailbox and set a user's password).

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.