

**NANYANG TECHNOLOGICAL UNIVERSITY**  
**SEMESTER 2 EXAMINATION 2020-2021**  
**CE2006/CZ2006 – SOFTWARE ENGINEERING**

Apr/May 2021

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 5 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 some of the questions.

- 
1. Based on the project description given in Appendix A,
    - (a) Identify the actors and use cases of the project and draw the use case diagram. Use <<include>> and <<extend>> relationships where appropriate.  
(10 marks)
    - (b) Write the use case description for the functionality of leaving the carpark. Your use case description must include description of the user case, pre-condition, flow of events and any alternative flow if it exists.  
(8 marks)
    - (c) From your use case description in Q1(b), identify the main classes and their associations, and draw a conceptual class diagram including their stereotypes and associations between them. You do not need to identify any attribute or operation within the classes.  
(7 marks)

2. (a) Draw a state machine diagram to show the states of the “leaving the carpark” component of the system.  
(7 marks)
- (b) Draw an activity machine diagram using the swimlane style to show the activities of the leaving car, the carpark leaving component, and the gate of the carpark when a car is leaving the carpark.  
(8 marks)
- (c) Assuming we have the following tasks for the carpark system project: Task A takes 4 weeks and can start after the project starts. Task B takes 2 weeks and should start after Task A finishes. Task C must happen after Task A and takes 3 weeks. Task D must happen after Tasks B and C and takes 3 weeks. Task E must take place after Tasks B and C and takes 5 weeks. Task F takes place after Tasks D and E and takes 2 weeks. Task G happens after Tasks D and E and takes 1 week. Task H is the last task after Tasks F and G and takes 2 weeks.
- (i) Draw the activity/task diagram including task names and durations based on the above description.  
(2 marks)
- (ii) Work out the critical path of the project based on the activity/task diagram derived from Q2(c)(i) with workings using the forward path analysis and the backward path analysis.  
(6 marks)
- (iii) Based on the workings in doing Q2(c)(ii), decide the shortest duration of the project and the slack time of Task D.  
(2 marks)
3. (a) Refer to the classes identified in your answer to Q1(c):
- (i) Propose an appropriate architecture for the carpark system, and draw a detailed Class diagram with key attributes and methods in each class to reflect the architecture design.  
(8 marks)
- (ii) Propose an alternative architecture design using a different design pattern and discuss the advantages of the alternative architecture.  
(6 marks)

Note: Question No. 3 continues on Page 3

- (b) Answer the following questions related to software design:
- (i) Propose two additional scenarios where you can apply observer pattern besides event framework like Java Swing framework.  
(4 marks)
  - (ii) Propose a scenario in the carpark system where façade pattern can be applied. And list as least three APIs in the façade pattern.  
(5 marks)
  - (iii) Discuss disadvantage of dynamic loading technique.  
(2 marks)
4. (a) When a vehicle enters the carpark, the following information will be sent to the carpark system:
- 1. Vehicle Number: the car plate number, which must be in ASCII format.
  - 2. Vehicle Type: one of “Car”, “Bus”, “Motorcycle”, “Lorry” and “Unknown”.
  - 3. Entry Time: the entering time stamp, which is a date between 2000 and 2021.
  - 4. ID Number: the unique ID of the vehicle in Singapore, which is a ten-digit integer value.
- (i) Determine the equivalence classes for the above FOUR inputs. (*Hint: for Entry Time, you can convert it to some data type we have learned in the class*)  
(4 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 vehicle information input interface. Design a set of test cases to test the FOUR inputs based on the equivalence classes and boundary values identified in your answers to Q4(a)(i) and Q4(a)(ii).  
(5 marks)

Note: Question No. 4 continues on Page 4

- (b) When a vehicle is arriving to the carpark, the carpark system will first check whether there is an available slot for the new arriving vehicle. The *checkAvailableSlot()* method (as shown in the Java code snippets in Figure Q4(b)) implements this interface.
- (i) Draw the control flow graph for the *checkAvailableSlot()* method.  
[Use the line numbers for clarity]  
(5 marks)
  - (ii) Calculate the Cyclomatic Complexity of the *checkAvailableSlot()* method.  
(2 marks)
  - (iii) List the basis set of linearly independent paths for performing basic path testing of the *checkAvailableSlot()* method. Design a test case (including the input parameters to the *checkAvailableSlot()* method and expected outcome) for each of the basic paths.  
(5 marks)

```

1.  bool checkAvailableSlot(int arrivingVehicleID) {
2.      bool available = true;
3.      Int existingVehicle = 0;
4.      //Vehicle class contains three attributes: (String number, int ID, Date entryTime)
5.      Vehicle[] vehicles = loadCurrentVehicles();
6.      for(int i = 0; i < vehicles.length; i++) {
7.          if(vehicles[i].entryTime < getCurTime()) { //if vehicle's entry time is valid
8.              if(arrivingVehicleID == videoRecords [i].ID) {
9.                  System.out.println("The car is already inside the carpark!");
10.                 available = false;
11.             } else {
12.                 existingVehicle++;
13.             }
14.         }
15.     }
16.     if(existingVehicle >= CARPARK_SIZE)
17.         available = false;
18. }
19. return available;
20. }
```

**Figure Q4(b)**

Note: Appendix A is on Page 5

**Appendix A****Carpark System (CS) Description**

The Carpark System (CS) keeps track of cars entering and leaving the carpark every day and provides both customers and administrators functionalities for carpark services. The initial requirements are given below.

A carpark system administrator shall maintain (insert/update/delete) the system data including the number of parking lots available in the carpark, charging rate at different time ranges of a day, etc. after login to the system. He/she shall be able to generate reports showing the carpark usage, parking charges, and fault operations for a specific duration respectively. When a car enters the carpark, the system shall record the car ID and the time entering the carpark. When a car leaves the carpark, the system shall record the car ID and the time leaving the carpark. The system shall calculate the charges for this car and deduct the amount of money from the cash card of the car/driver. If the balance in the cash card is not enough, an error message shall display and the gate of the carpark shall keep closed. If the parking charge is deducted successfully, the gate will open and the car can leave the carpark. On each parking lot of the carpark, there is a sensor to detect whether the parking lot is occupied. The system uses the sensor data to monitor the status of a parking lot. If an empty parking lot is occupied, the vacant parking lot number shall be decreased by 1. If a parking lot becomes empty, the vacant parking lot number shall be increased by 1. The vacant parking lot number is displayed at electronic boards installed at different places of the carpark.

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.