

THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING
BACHELOR OF SOFTWARE ENGINEERING



ECI4262 – OBJECT ORIENTED DESIGN AND PROGRAMMING

FINAL EXAMINATION - 2011/ 12

CLOSED BOOK

Date: March 10, 2012

Time: 1400 – 1700 hrs

<INSTRUCTIONS >

Answer Q1, compulsory question and any other **THREE** questions from Q2 – Q6.

[Q1] (Compulsory question)

EHelpDesk of E Marketing Solutions is a web-based IT support desk system designed to handle the requests for technical assistance coming from internal users. The system is designed to ensure efficient management of internal support operations, streamline task distribution, task monitoring and to maintain a centralized problem tracking system accessible by both end users and technicians at head office for documenting technical issues with resolutions.

The major functionalities of the support desk system include; web-based request submission, track progress of a reported issue, automatic email notifications, issue search, searchable knowledge base with resolutions for frequently encountered issues, allow different user levels and report generation.

The end users seek assistance and the technicians handle the reported issues. The administrator is responsible for ensuring the smooth management of the reported requests for technical assistance.

Given below are use case descriptions that elaborate how the request submissions and technician assignment happens.

Scenario 1: Request Submission

- Use case name: Submit request
- Description: When technical support required the User can place a support request over the World Wide Web. When a request is placed the user has to provide a complete description of the problem encountered.

Contd Page /2

- **Preconditions:** User is identified and authenticated.
- **Successful End Condition:** A new request is registered with the system and the user gets an email notification.
- **Failed End Condition:** The request is not submitted.
- **Basic Flow**
 - The use case begins when the user selects "Submit Request" option
 - The system displays the new request screen
 - The user enters the Subject, Description, Asset Details and the Request type.
 - The system submits the request.
 - The system generates a reference number and sends a notification Email to the user with the request reference number for further queries.
 - The system displays the request list of the current user.
 - The use case ends.

Scenario 2: Assign Technician

- **Use case name:** Assign Technician
- **Brief Description:** When a new request submitted to the system the administrator should be able to assign the job to one of the technicians. This use case facilitates assignment of support staff and notifying the technician of interest.
- **Preconditions:** The login use case for Administrator must be executed.
- **Successful End Condition:** The request is assigned to a technician and the technician is notified regarding the new assignment.
- **Failed End Condition:** The request remains unassigned.
- **Basic Flow**
 - The use case begins when the user selects "Assign Technician" option.
 - The system displays a list of available technicians.
 - The administrator selects a technician.
 - The system displays the allocation history for the selected technician.
 - The administrator confirms assignment.
 - The system assigns the job to the technician
 - The system sends a notification Email to the technician with allocation details.
 - The use case ends.

- a) Briefly describe the following concepts using examples from scenario 1. (10 marks)
- i. Actor
 - ii. Class
 - iii. Object
 - iv. Use case
 - v. Behaviour and attributes
- b) Draw the use case diagram identifying how the use cases are inter-related for the above two scenarios.(10 marks)
- c) Draw the sequence diagram for the use case for 'Assign Technician'.(5 marks)
- d) How does a sequence diagram differ from collaboration diagram?(5 marks)
- e) Develop the class diagram showing the relationships among the classes for the above two scenarios.(10 marks)

[40 Marks]

[Q2]

- a) A variable in a class can be declared with one of the "access modifiers" *public*, *private*, or *protected*, or it can be declared with no access modifier at all. Discuss the effect of each of these four possibilities. (4 marks)
- b) Briefly describe the following concepts.(4 marks)
- i. Instance method
 - ii. Class method
 - iii. Constructor
 - iv. Multiple inheritance
- c) In an object oriented programming language with which you are familiar, provide sample code that demonstrates the use of EACH of the concept given in part (b). (12 marks)

[20 Marks]

[Q3]

- a) List three reasons why it is useful to be able to inherit characteristics from parent classes.(3 marks)
- b) What are the different types of inheritance? Explain them.(8 marks)
- c) What are the similarities and differences between an Abstract class and an Interface?(4marks)
- d) Explain the difference between abstraction and encapsulation. (5 marks)

[20 Marks]

- a) Categorize the following design patterns in to the three design pattern categories namely creational design patterns, structural design patterns and behavioral design patterns: (6 marks)
- i. Abstract Factory
 - ii. Facade
 - iii. Proxy
 - iv. Singleton
 - v. Observer
 - vi. Prototype
- b) Give an example where the Singleton design pattern would be useful.(4 marks)
- c) Name the design pattern which matches the description given below.(10 marks)
- i. Defines an interface for creating an object, but let the subclasses decide which class to instantiate. It lets the instantiation differ to subclasses.
 - ii. Attaches additional responsibilities to an object dynamically. It provides a flexible alternative to sub classing for extending functionality.
 - iii. Defines a family of algorithms, encapsulate each one, and make them interchangeable. It lets the algorithm vary independently from clients that use it.
 - iv. Defines one-to-many dependencies between objects so that when one object changes state, all its dependents are notified and updated automatically.
 - v. Encapsulates a request as an object, there by letting you parameterize clients with different requests, queue or log requests, and support undoable operations.
 - vi. Ensures a class has only one instance, and provide a global access point to it.
 - vii. Defines the Skelton of an algorithm in an operation, deferring some steps to subclasses. It lets subclasses redefine certain steps of an algorithm without changing the algorithm structure.
 - viii. Provides unified interface to a set of interfaces in a subsystem. It defines a higher level interface that makes the subsystem easier to use.
 - ix. Provides a way to access the elements of an aggregate object sequentially without exposing its underlying representation.
 - x. Provide a surrogate or placeholder for another object to control access to it.

[20 Marks]

[Q5]

00059

a) Write the Java code segment to find the factorial of a given number.(7 marks)

Formula: $n! = 1 * 2 * 3 * \dots * n$

Where

$n!$ represents n factorial

n = Number of sets (Non negative integer)

b) Suppose that *color* is a variable of type *Color*. Write a code segment that will set *color* to one of the four standard colors *Color.black*, *Color.red*, *Color.blue*, or *Color.green*.

(Hint: The color should be chosen randomly with equal probability for each color. (Recall that `(int) (4*Math.random ())` is an integer in the range 0 to 3, inclusive.)(7 marks)

c) Write the Java code segment to reverse the elements of an array.(6 marks)

[20 Marks]

[Q6]

a) What are the advantages of using UML for modeling? (2 marks)

b) Briefly describe the following UML diagrams.(16 marks)

- i. Use case diagram
- ii. Sequence diagram
- iii. Collaboration diagram
- iv. State diagram
- v. Class diagram
- vi. Component diagram
- vii. Deployment diagram
- viii. Activity diagram

c) Name four UML modeling tools available.(2 marks)

[20 Marks]

---- End of Paper ----