Design and Analysis of Software Systems
Mid-Semester Exam – Spring 2024

This is a closed-book, closed-notes exam. It consists of 12 questions. You should answer all questions. You have 90 minutes to complete the exam for a total of 50 points.

1. **Multiple choice questions (7 points) - 1 point each**

a. Use-case actors are always people, never system devices.
   a. True
   b. False

b. The _____ for a project is the timetable that specifies when each activity should start and finish.
   a. objective
   b. schedule
   c. scope
   d. time scope

c. The need and requirements are usually written up by the customer in a document called:
   a. request for price
   b. request for proposal
   c. request for contracts
   d. request for bids

d. CPM stands for:
   a. critical part model
   b. critical path method
   c. criteria plan method
   d. criteria part model

e. What are the activities associated with requirements engineering?
   a. Elicitation, Verification & Validation, Code Inspections, Design Reviews
   b. Elicitation, Modeling, Domain Analysis, Specification
   c. Domain Analysis, Object-Oriented Design, Testing, Configuration Management
   d. Specification, Object-Oriented Design, Code Inspections, Domain Analysis

f. Acceptance tests are normally conducted by the
   a. End users
   b. Developers
   c. Test team
   d. Systems Engineers

g. The _____ process model is a risk-driven process model
   a. Scrum
   b. Prototype
   c. Spiral
   d. Waterfall

2. **(2 points)** At the end of each stage of some lifecycle models we perform "V & V." Each V is shorthand for a software project activity. What is the goal of each step?

3. **(2 points)** What does product backlog mean?

4. **(2 points)** Describe what is regression testing and why is it done?

5. **(3 points)** Making sure to use proper notation (symbols), draw a UML class diagram for:
   a) A Generalization between Phone and iPhone
   b) A Multiplicity of 1 to many for a Manager and Assistant
   c) A strong aggregation of Team and Member where only the aggregate knows about the other.

6. **(3 points)** Consider the following specification for a program:
   - The user shall respond **yes** or **no**.

   Based on this specification:
   Define *three distinct* equivalence classes that should be used to test the program, and For each class, tell whether the elements of the class should be treated as valid or erroneous

7. **(4 points)** Assuming you are a member of the Megathon (hackathon organized by IIITH), detail out the Initiatives, activity lists, high level effort estimates and priorities (high-medium-low). This should be in a form that can be used to track just the progress of the project as things get done.

8. **(4 points)** What characteristics of good requirements does the following requirements statement lack? Rewrite it to reflect the characteristics of good requirements.

   "The cash register program must show the correct prices for all items bought by the customer and print the total at the bottom."

9. **(4 points)** Based on the following table, calculate the ES (Early Start), EF (Early Finish), LS (Late Start), LF (Late Finish) times and SLACK for each task assuming a expected completion time of 16 weeks. Can the project still be completed on time? If necessary, identify the tasks you would concentrate on to get the project back on schedule.

Task	Est. Time	Act. Finish Time	Immediate Predecessor
A	3	4	-
B	3	6	-
C	4		A
D	5		A
E	6		B
F	7		C,D,E

10. (Total 8 points)

The two parts of this question refer to this system description.

Your team is designing a course registration system to be used by colleges and universities. Students need to be able to register for classes. A student can view course listings, add and drop classes, and view his or her schedule. The registrar needs to access statistics about student enrollment, and also needs information about all classes that have too few students to run. Department secretaries maintain the course listings and schedules. The secretary can add/modify/delete a program's course listings and course section offerings scheduled to run in

future terms. Instructors are allowed to view registration information for any course he or she is teaching. Secure access to only the information a person is allowed to see must be maintained. The institution's central database must be used for all persistent storage.

A. **(4 points)** List the features (use cases) for this system and draw a use case diagram.
B. **(4 points)** Choose one of your features and write the main flow and alternate flows for it.

11. (3 points) The following pseudo-code for a function that calculates the monthly repayment amount for a home loan. Identify at least three potential edge cases or error conditions and write test cases for them.

```
function calculateRepayment (principal, annualInterestRate, years) {
  monthlyInterestRate = annualInterestRate / 12
  numberOfPayments = years * 12
  return (principal * monthlyInterestRate) / (1 - (1 + monthlyInterestRate)^(-numberOfPayments))
}
```

12. (8 points) You are provided with a Python module **inventory_manager.py**. This module contains a class **InventoryManager** that manages an inventory of products. The class for InventoryManager is given below. The **InventoryManager** handles adding and removing of products to the inventory, querying product quantities, and calculating the total inventory value based on a price list.

```
class InventoryManager:
    def __init__(self):
        self.products = {}

    def add_product(self, product_id, quantity):
        if quantity <= 0:
            raise ValueError("Quantity must be positive")
        if product_id in self.products:
            self.products[product_id] += quantity
        else:
            self.products[product_id] = quantity

    def remove_product(self, product_id, quantity):
        if product_id not in self.products or self.products[product_id] < quantity:
            raise ValueError("Insufficient quantity or product not found")
        self.products[product_id] -= quantity
        if self.products[product_id] == 0:
            del self.products[product_id]

    def get_product_quantity(self, product_id):
        return self.products.get(product_id, 0)

    def total_inventory_value(self, price_list):
        total = 0
        for product_id, quantity in self.products.items():
            total += price_list.get(product_id, 0) * quantity
        return total
```

Write a suite of Unittests for the InventoryManager class.