

Questions SE

👤 Created by	ⓑ Borhan
🕒 Last edited time	@June 15, 2025 2:04 PM
⋮ Tag	

<https://www.studocu.com/en-us/document/fairleigh-dickinson-university/digital-marketing-strategies/dfd-case-study-with-solutions-updated/77995749>

<https://sadamutmspace.blogspot.com/2015/06/meet-4-exercise-on-dfd-read-casebelow.html>

PAGE 1

Course Title: Software Engineering and Information System Design

Department: Computer Science and Telecommunication Engineering, NSTU

Year: 3rd Year, Second Term, Final Examination, Session 2012-'13

Course Code: CSTE 3209

Full Marks: 70

Time: Four hours

Instructions: Answer any SEVEN of the following questions. Parts of each question should be answered consecutively.

1.

a) What is software engineering? What is the difference between software engineering and system engineering? (5)

Software Engineering is an engineering branch associated with development of software product using well-defined scientific principles method and procedure.

Aspects	Software Engineering	System Engineering
Definition	Software Engineering is an engineering branch associated	System engineering isa an interdisciplinary field that focuses

Aspects	Software Engineering	System Engineering
	with development of software product using well-defined scientific principles method and procedure.	on integrating and managing complex system.
Scope	Focuses on software components	Holistic approach to entire system
Objective	To develop, maintain and improve software product	To ensure system functionality and integrating various parts
Method	Scrum, Agile, Devops	MBSE
Deliverable	Software Application	Fully functional system
Application	IT and Software Industry	Broad application in aerospace, automobile
Stakeholder Interaction	Software users, developers and IT Personals	A broader range of stakeholders from various fields

b) Describe the generic process framework activities. (5)

1. Communication: It is the first and foremost thing for the development of software development. Communication is necessary to know the actual demands of the client. It includes customer and stakeholders
2. Planning : It basically means drawing a map for reduced the complication of development. It includes work plan. describe technical risk, list of resource requirements etc.
3. Modelling: In this process a model is created according to the client for better understanding. The software model is prepared by
 - a. Analysis of requirements
 - b. Design
4. Construction: It includes the coding and testing of the problem. It also includes fixing bugs and confirming that all criteria are met.
5. Deployment: It includes the delivery of software to the client for evaluation and feedback. On the basis of their feedback, we modify the product for the supply of better product.

c) "Prototype can be problematic for some reasons" - What are the reasons? (5)

1. This model is costly
2. It has poor documentation because of continuously changing customer requirements
3. There are may be too much variation in requirements
4. Customers sometimes demand the actual product to be delivered soon after seen an early prototype,
5. There are suboptimal solutions because of developers in a hurry to build prototypes.
6. Customers may not be satisfied or interested after seeing the initial prototype,
7. There is certainly in determining the number of iterations.
8. There are may be incomplete or inadequate problem analysis
9. There may increase the Complexity of the system.

2.

a) Which development process model would you employ for the following projects? Why? (5)

- A data entry system for office staff who has never used computers. The user interface and user-friendliness are extremely important.

Prototyping.

This model involves creating early versions (prototypes) of the software **to gather user feedback and make iterative improvements**. It allows users to interact with and provide feedback on the interface, which helps in designing a more user-friendly system. It is particularly useful when the requirements are not well-defined initially or when usability is a critical concern.

- A Website for an online store which has a long list of desired features to be added, and it wants a new release with new features to be done very

frequently.

Agile Model.

The Agile model supports

frequent releases, incremental delivery, and continuous feedback. It is ideal for projects where the requirements are expected to change or evolve, as in an online store that wants to add new features regularly. Agile enables **quick adaptation to changes**, ensures **customer involvement**, and **delivers working software in short iterations.**

b) What are the steps of the software development life cycle? If you were a lead developer of a software company and you are asked to submit a project/product within a stipulated time-frame with no cost barriers, which model would you select? Why? (5)

Steps of the software development life cycle

1. Project Planning
2. Gathering Requirements and Analysis
3. Design
4. Coding or Implementation
5. Testing
6. Deployment
7. Maintenance

If I were the lead developer of a software company and asked to submit a **project/product within a fixed time-frame** but with **no cost barriers**, I would select the **Rapid Application Development (RAD) Model.**

- Focus on **rapid prototyping** over costly planning.
- Frequent **user involvement** and feedback.
- Use of **component-based development.**
- Suitable for **projects with well-defined modules** and **tight deadlines.**
- Requires **skilled developers** and **plentiful resources**

[Since **cost is not a constraint**, but **time is fixed**, the **Spiral Model** is the most appropriate among your syllabus options. It allows **iterative releases**, handles

risk and complexity, and supports **frequent revisions** with high quality — ensuring **on-time delivery with flexibility**.]

3.

a) Discuss the following four points of effective software management: people, product, process, and project. (5)

(The management Spectrum)

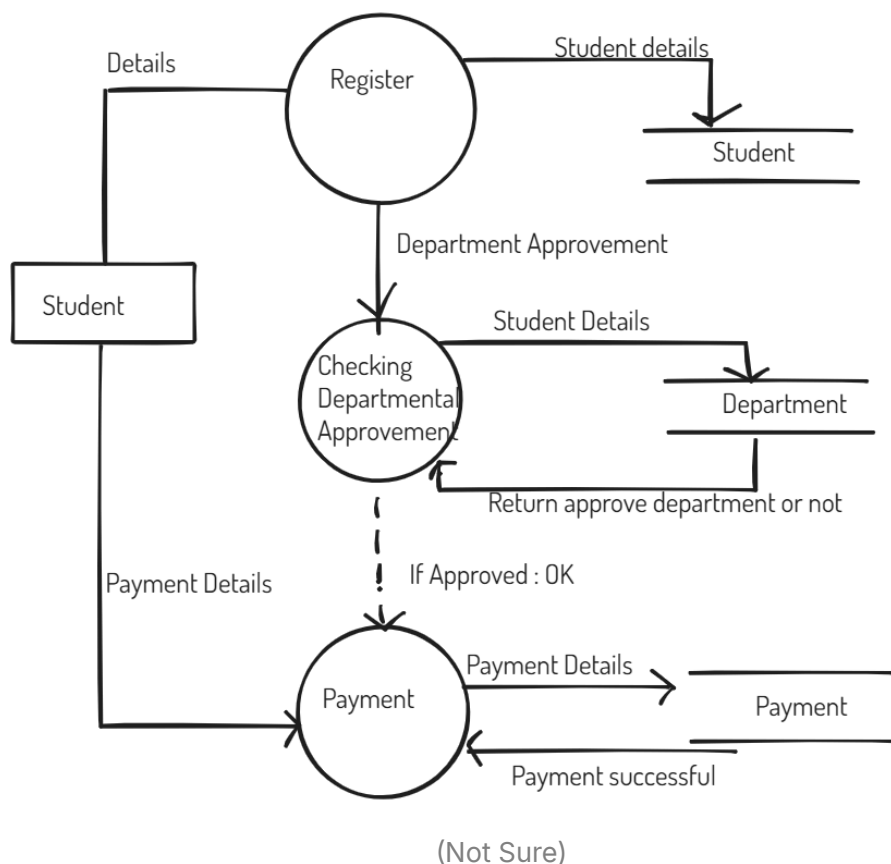
1. People : The most important contribution in software project is not made by system or tool. It is done by people, human resources. The success of project depend on selecting right kind of people with right talent.

Roles:

- a. Senior Manager : Define business issues
 - b. Project manager : Plan, motivate, organize and control project work
 - c. Software Engineer : who deliver technical skills
 - d. Customer : Who specify the requirements
 - e. End user: who interact with the software product
2. Product: The product is the ultimate goal of the project. This is any types of software product that has to be developed and deliverable. Before a product can be planned, its objectives and scope should be established, alternate solutions should also considered. Lack of these information, it is impossible to define reasonable and accurate estimation.
 3. Process: Project manager and team members should have a methodology and plan that complete project as per customer requirements. Without a clearly defined process, team members will not know what to do and when to carry out project activities. Using the right process will increase the project execution success rate that meet its original goals and objectives. Process has several steps documentation, designing, implementation, deployment etc.
 4. Project: The project is the complete software project that includes requirements analysis, development, delivery maintenance and updates. The project manager plays a vital role for completing project. They are responsible to guide the team members to achieve the project target and

objectives. They helping and assisting with the issues, checking on cost and budget. They mange complete project activities to avoid project failure.

b) Consider a program which registers students for different programs. The students fill a form and submit it. This is sent to the departments for confirmation. Once it is confirmed, the form and the fees are sent to the account section. Draw a data flow diagram for this purpose. (5)



4.

a) Explain how both the waterfall model of the software process and the prototyping model can be accommodated in the spiral process model. (5)

the spiral process model accommodates the waterfall model by breaking it into iterative phases and accommodates the prototyping model by allowing for the creation and refinement of prototypes in each iteration.

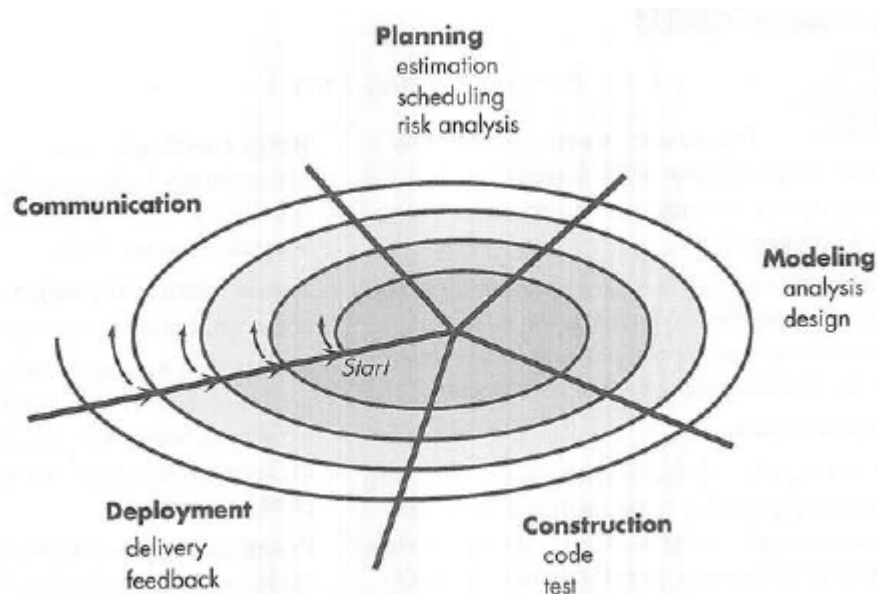
The waterfall model can be accommodated in the spiral process model by having each iteration represent a phase of the waterfall model. For example, the

first iteration could focus on requirements gathering and analysis, the second on design, the third on implementation, and so on.

Similarly, the prototyping model can be accommodated by using the spiral model's iterative nature to create and refine prototypes in each iteration, incorporating user feedback and refining the prototype until it evolves into the final product.

Steps

- i. The first loop may result in the development of a basic prototype of the final product.
- ii. The subsequent loops may result in the gradual development of more mature versions of the product.
- iii. This spiral continues until an acceptable software product is built and delivered to the client.



b) Identify the objects and their attributes, and processes from the following statement of a problem domain by employing Natural Language Approach. A salary system which computes employee salaries and deductions. The input is a list of employee numbers who are to be paid that month. The system maintains tables holding tax rates and the annual salary for each

employee. The output is a salary slip for each employee plus a list of automated payments to be made by the company's bank. (5)

Analysis Using Natural Language Approach

Category	Identified Elements	Explanation
Objects	- Salary System- Employee- Employee Number- Salary Slip- Tax Rate- Annual Salary- Payment List- Bank	All are nouns in the domain and represent entities involved in the system.
Attributes	- Employee Number- Annual Salary- Tax Rate- Deductions- Net Salary- Payment Amount	These are descriptive features of the objects, mostly related to Employee , Salary Slip , or Payments .
Processes	- Compute Salaries- Compute Deductions- Generate Salary Slip- Generate Payment List- Send to Bank	These are verbs/actions representing what the system does.

Structured Summary

Object	Attributes	Processes involving object
Employee	Employee Number, Annual Salary, Tax Rate	Identified from the list, used in salary computation
Salary Slip	Deductions, Net Salary, Tax Amount	Generated per employee after computation
Salary System	—	Computes salary and deductions
Bank	—	Receives list of automated payments
Payment List	Payment Amounts per Employee	Created for bank transfer

5.

a) What is the fundamental reason that software cannot be considered to be over-engineered? (5)

Software cannot be considered to be over-engineered because software is intangible, ever-evolving, and highly dependent on changing requirements, technologies, and user expectations.

Formal Justification:

Reason	Explanation
1. Changing Requirements	Software requirements often evolve over time due to market demands, user feedback, or business goals. What may seem "extra" today could become essential tomorrow. Therefore, preparing the software to adapt is not over-engineering but foresight.
2. Scalability and Maintainability	Adding robustness, modularity, or extensibility may increase initial effort but greatly reduces long-term cost and maintenance effort. This makes the system more future-proof.
3. Intangible Nature	Unlike physical products, software has no manufacturing cost per unit and no material constraints. Hence, extra design or abstraction doesn't add physical waste — making "over-engineering" less meaningful.
4. Quality Attributes	Enhancing software with better performance, security, usability, and reliability is always beneficial, even if it requires more effort. These are not wasteful but quality improvements .
5. Risk Management	Building for flexibility, validation, and error handling early may seem excessive but helps avoid failures, especially in critical systems. This is strategic engineering, not over-engineering.

b) The ATM system accepts customer requests and delivers cash, account information, database updates, and so on. Customers may send standard messages to their bank branch requesting an account statement, cheque book, and so forth. Facilities for customers to initiate an electronic funds transfer may be available. The machines provided by a particular bank may allow customers of other banks to use a subset (typically cash withdrawal and account balance querying) of their facilities. Give a preferred description of the exception handling process in ATM and write the exception handling process in program description language. (5)

6.

a) Discuss the various design strategies. Which design strategy is the most popular and practical? (5)

b) What is estimation for software projects? Describe the COCOMO II model for software project estimation. (5)

Estimation in software engineering is a procedure that predicts the time and budget that required for completing a project. It required complex tools and mathematical and planning knowledge.

COCOMO 2 : Intermediate Cocomo Model

Extension of basic model which enhance more accuracy to cost estimation model result. It includes cost drivers.

$$E = a * (KLOC)^b * EAF$$

$$D = c * E^d$$

EAF is Effort adjustment factor which is calculated by multiplying the parameter values of different cost driver parameters.

c) Define metrics, measures, and indicators. (5)

d) What are the characteristics of a good design? (5)

1. Functionality
2. Usability
3. Reliability
4. Performance
5. Security
6. Maintainability
7. Reusability
8. Scalability
9. Testability

e) What is Forward Engineering? Describe the activities of Forward Engineering for Client-Server Architecture. (5)

f) Write the generic steps to perform Object-Oriented analysis. (5)

g) Describe the difference between static and dynamic views of an Object-Oriented system. (5)

h) What role does cardinality play in the development of an object-relationship model? (5)

i) How do Object-Oriented design and structured design differ? What aspects of these two design methods are the same? (5)

PAGE 2

1.

a) What is regression testing? Give three situations where the testing of all independent paths through a program may not detect program errors. (5)

Regression testing is like a software quality checkup after any changes are made. It involves running tests to make sure that everything still works as it should,

b) What is cyclometric complexity? What is dynamic testing? (5)

c) Let's firstly consider an example program to test:

- Suppose we want to have a program to classify triangles from the lengths of their sides.
- The program prints out whether the program is Scalene, Isosceles, Right Angled, Equilateral, or Not a triangle.
Now, using the dynamic testing procedure, how can you test the above program? (5)

1. The main program below creates a simple invoice and causes it to be printed out. Draw an object diagram showing the objects that are created when the method main runs and the messages that are passed between these objects when the invoice is printed:

```
java
```

```
CollapseWrap
```

Copy

```
public class main
{
    public static void main(String[] args)
    {
        Client jewel = new Client("John Jewel");
        Invoice inv = new Invoice(jewel);
        inv.add(new goods("new flours", 100.0));
        inv.add(new staff("staff", 50, 2));
        inv.print();
    }
}
... (5)
```

2.

- a) Define Reverse Engineering? Write a snippet on Java class for the following simple class diagram. (5)
- b) What are the tools used for web testing? Describe different types of Maintenance that are used after completion of the software system. (5)
- c) Using the binary search techniques, we find the hidden number within a moment. Though we've to accomplish the sorting first, then we can apply the searching procedure. A Java program that does binary search has 10 modules where 5 modules are used to take information from the user and 5 others are used to show the corresponding output upon the value from the inner modules. Now calculate the complexities of the program using Henry Kafura's Metric. (5)

PAGE 3

Course Title: Software Engineering and Information System Design

Department: Computer Science and Telecommunication Engineering

Year: 3rd Year, Term 2, Final Examination (December 2021), Session 2017-18

Course Code: CSTE 3209

Full Marks: 70

Time: Four hours

Instructions: Answer any SEVEN of the following questions.

1. a) When you know programming, what is the need to learn software engineering concepts? Describe the essential attributes of good software. (5)
- b) What do you mean by software project management? Why do we need

software project management? What are the software management activities? (5)

2. a) Explain the requirement engineering process. (4)
b) What are software metrics and how can they be used to manage a software project and the software process? (4)
c) What are the characteristics of a successful team? (2)
3. a) What do you mean by Agile Process? How do planning, design, coding, and testing activities of Extreme Programming (XP) differ from other process models? (4)
b) What are the new practices that Industrial XP incorporates to ensure that an XP project works successfully within a large organization? (4)
4. a) How does Scrum differ from Extreme Programming? (4)
b) How does agility impact the cost of change? Draw and explain the cost of change curve of conventional software and agile processes. (4)
c) "Pareto principle: 80 percent of an application can be delivered in 20 percent of the time it would take to deliver the complete (100 percent) application." - How does DSDM follow this philosophy? (2)
5. a) What do you mean by Refactoring? Why is Low Coupling & High Cohesion Required? (4)
b) Write the generic steps to perform object-oriented analysis. (4)
c) Design a class diagram for the below Order System:
 - i. First of all, Order and Customer are identified as the two elements of the system. They have a one-to-many relationship because a customer can have multiple orders.
 - ii. Order class is an abstract class and it has two concrete classes (inheritance relationship) SpecialOrder and NormalOrder. The two inherited classes have all the properties as the Order class. In addition, they have additional functions like dispatch() and receive(). (4)
6. a) What are the coding principles that should be followed in the software process? (4)
b) What do you mean by data flow diagram? Describe different levels of data flow diagram. Draw DFD for hospital management system. (4)
7. a) What is the difference between cardinality and modality? Explain with an example. (4)
b) Explain the test case design for OO software. (4)

- c) Discuss the key factors that distinguish development and maintenance.
What do you understand by inheritance and polymorphism? (4)
-

PAGE 4

Date: 7th February 2022

1. a) Explain briefly the taxonomy of software architecture styles. (4)
b) What is the impact of OO programming on software testing? Explain. (4)
c) Write short notes on: Trademarks, Copyrights, Patents. (4)
 2. a) Describe MVC Web Architecture. (4)
b) Why does software need maintenance? Define corrective, adaptive, perfective, and preventive maintenance. (4)
c) What do you mean by Software Re-engineering and Reverse engineering? (4)
-

PAGE 5

Course Title: Software Engineering and Information System Design

Department: Computer Science and Telecommunication Engineering

Year: 3rd Year, Term 2, Final Examination (January 2024), Session 2019-20

Course Code: CSTE 3209

Full Marks: 70

Time: Four hours

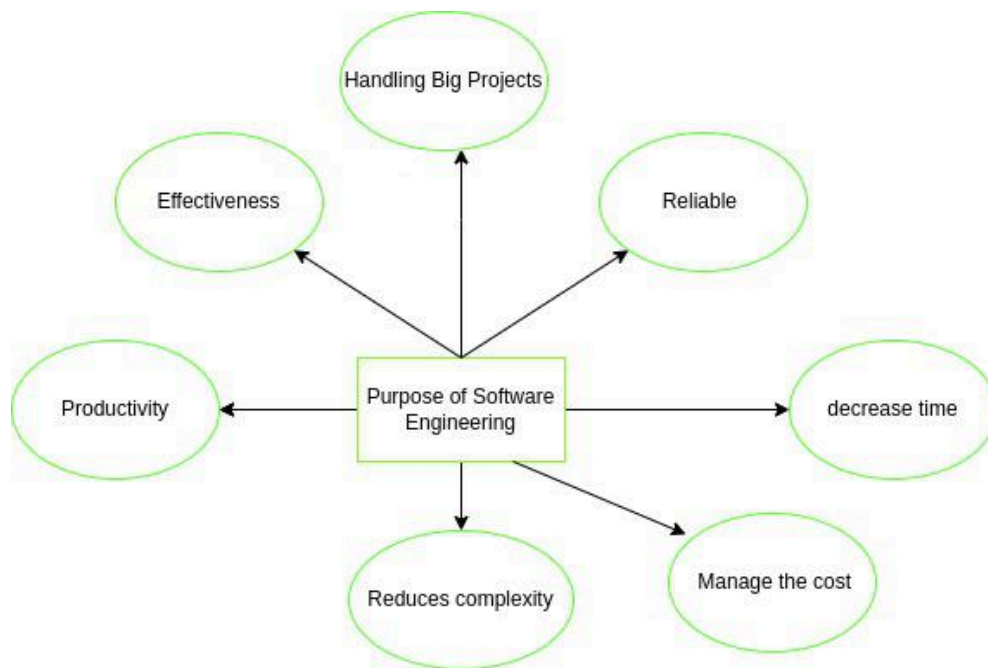
Instructions: Answer any SEVEN of the following questions.

1.

a) How do you differentiate between computer software and computer programs? When you know programming, what is the need to learn software engineering concepts? (2+3)

Aspect	Software	Program
Dependency	Software is mainly dependent on the <u>operating system</u> .	Programs are mainly dependent on the compiler.
Categories	Various software categories include application software, system software, computer programming tools, etc.	There are no such categories of programs.

Size	The size of software generally ranges from megabytes (Mb) to gigabytes (Gb).	The program size generally ranges from kilobytes (Kb) to megabytes (Mb).
Developer Expertise	Software is usually developed by people having expert knowledge and experience as well as are trained in developing software and are also referred to as software developers.	Programs are usually developed by the person who is a beginner and has no prior experience.
Nature	Software's can be a program that generally runs on computer.	Programs cannot be a software.
Necessity for Computer Functionality	If software's are not present in computers, then computer is useless.	If programs are not present in computer, then also computer can function well because of operating system.
Download	Software's can be downloaded on computer using internet without any need of program.	Program cannot run on computer without any software present in computer.
Features	Features of software includes security, safety, dependability, correctness, etc.	Features of program includes reliable, cost effectiveness, maintainability, profitability, etc.
Development Time	It requires more time to create software than program.	It requires less time to create program than software.
Examples	Examples of software includes Adobe Photoshop, Google Chrome, PowerPoint, Adobe Reader, etc.	Examples of program includes Web browsers, word processors, video games, etc.



b) What does a software project manager do? (2)

1. People Management :
2. Project Management

c) What is project estimation? How can we derive the size of a software product? (3)

Project estimation is a procedure that predicts the time and budget that required for completing a project.

1. Line of Code : Easiest and simpler metrics, calculating and comparing the productivity of programmers, not count comment or blank line in
2. Functional Point: function-oriented, what the system performs, counting the type and type of functions

1.

a) What is process flow? Explain different types of process flow with proper diagrams. (1+4)

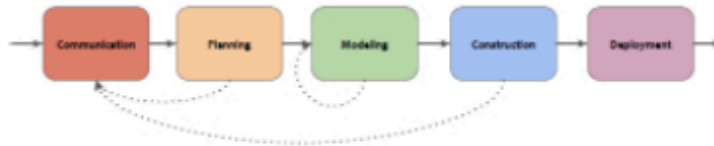
A process flow is a sequence of steps and activities followed to complete a task.

iii. 4 unique process flows

1. **Linear Process Flow:** Linear process flow executes each of the 5 (five) activities sequentially.



2. **Iterative Process Flow:** Iterative process flow repeats one or more activities before proceeding to the next activity.



3. **Evolutionary Process Flow:** Evolutionary process flow executes activities in a circular fashion. Each cycle leads to a more complete version of the software.



4. **Parallel Process Flow:** Parallel process flow executes one or more activities in parallel with other activities.



b) "The SEI capability maturity model contains five levels." - What are they? Explain shortly. (3)

the Software Engineering Institute (SEI)'s Capability Maturity Model (CMM) ..

1. Initial: process are initially uncontrolled
2. Managed/Repeated : product management procedure defined and used
3. Defined : process management procedure and strategies defined and used
4. Managed : quality management procedure and strategies defined and used
5. Optimizing : process improvement strategies defined and used

c) What are the requirements for coping with change? (2)

1.

a) What is process pattern? Explain different types of process patterns with examples. (3)

A process pattern describe a process-related problem is encountered during software development, identifies the environment and suggest one or more proven solutions.

1. Task Pattern : related to SE work/actions and relevant to successful SE practice. Ex: requirementGathering
2. Stage : problem associated to framework activity. example : establishing communication
3. Phase : define the sequence of framework activities that occur in a process. example : Spiral of prototyping

b) Differentiate between prototyping and spiral model. (3)

c) Discuss the life cycle of the unified process with an appropriate diagram. (3)

Aspect	Waterfall	Incremental	Prototyping	Spiral
Definition	Different activities are	Involves development	A prototype is built, tested	Development of the software

	executed in a sequential and systematic manner.	and deployment of a series of versions of the software product, known as increments.	and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed.	product through a series of versions of that product. Deals with the uncertainties in software project by incorporating different risk analysis techniques throughout the process.
Project size	Small			
Requirements	Well-defined and known before the starting of the project.	More or less fixed and clear	Vague and likely to change	Vague and likely to change
Phase 1	The process begins with communication, where requirements are collected from the client and documented.	A simple functioning system, known as core product, which handles basic requirements is first developed and delivered.	One or more prototypes of the software product are built with currently known client requirements before the development of final product.	The first loop may result in the development of a basic prototype of the final product.
Phase 2	In planning phase, the time and financial constraints of the project are estimated, resulting in a schedule and a budget.	Client feedback is collected after each incremental delivery to incorporate in the next increment	The client evaluates the prototype and provides feedback as well as additional requirements which get	The subsequent loops may result in the gradual development of more mature versions of the product.

			incorporated in next prototype.	
Phase 3	Then, a design of the software product is crafted in modeling phase based on gathered requirements and keeping project constraints in mind.	Multiple increments are delivered by adding more functionalities, as per the client requirements, until the final version of the product is released.	This workflow is repeated until the prototype evolves into a complete product, acceptable to the client.	This spiral continues until an acceptable software product is built and delivered to the client.
Phase 4	In construction phase, essential code is generated and tested to build the final product.			
Phase 5	Finally, in deployment phase, the product is delivered to the client and necessary maintenance is provided based on the client feedback.			
Advantage	Simple model to use and implement. Easily understandable workflow Easily manageable as requirements are known before the	Flexible to changing requirements. Promotes active involvement of the Modifications can be made throughout the	Client feedback helps to better understand the product and facilitates early detection of error in the product Detailed client requirements	Any type of changes can be incorporated even at a later stage of the process

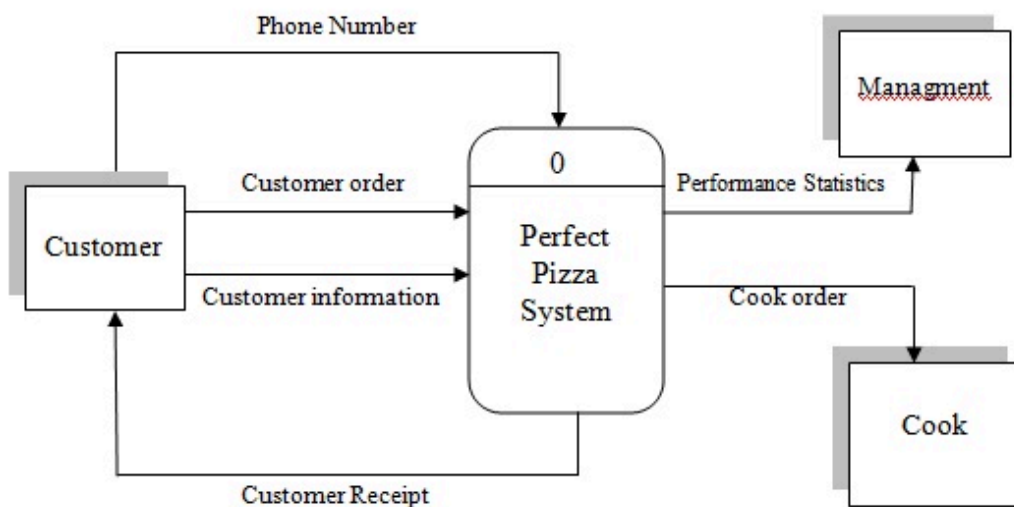
	starting of the project	process Errors are mitigated as the product is assessed by the client after each incremental delivery Functioning software product is available at the early stage of the process Product can easily be tested because of multiple iterations The initial project cost is lower	are not needed to start the project	
Disadvantage	It may become tough for the client to provide all the requirements beforehand. Testing and client evaluation are carried out in the last phases resulting in high risk Iteration of activities is not promoted which may be crucial for some projects.	Breaking the problem into increments is difficult Total project cost is high A complete planning of the project is required before committing Refining requirements in each iteration may affect the software architecture	Prototyping can slow down the process Frequent changes may increase complexity of the system Client dissatisfaction may lead to scrapping of multiple prototypes	Process gets costly and complicated Requires risk assessment expertise

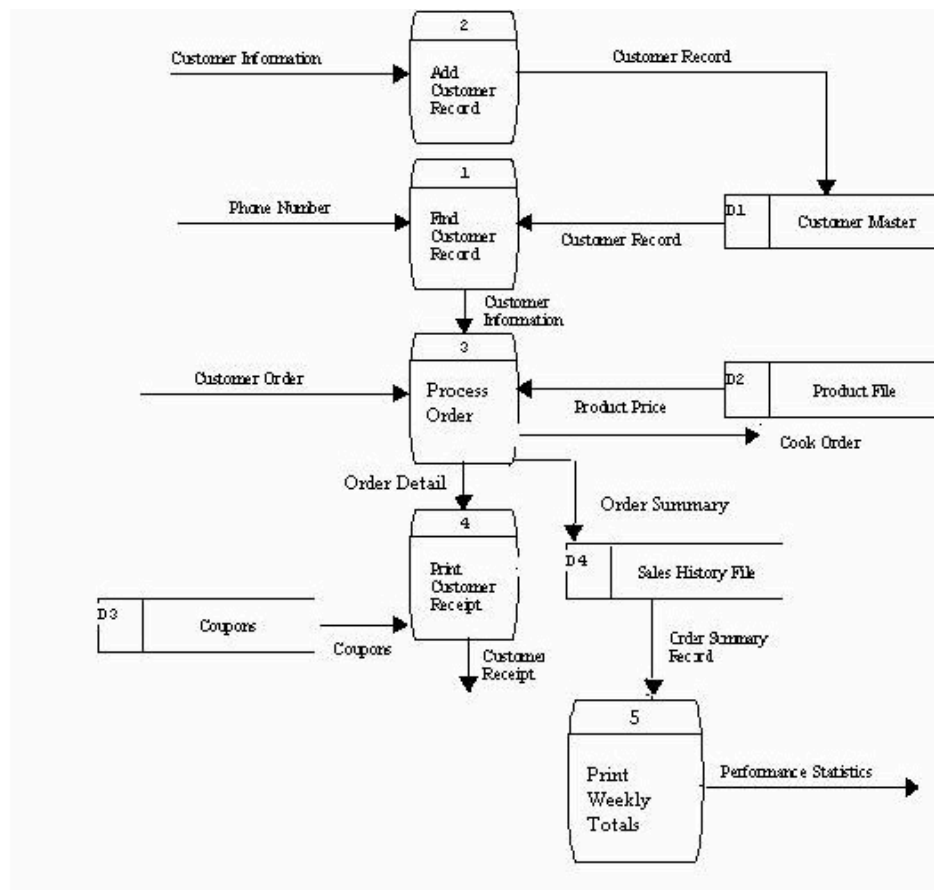
1. a) What do you mean by agility? Discuss the new practices of IXP ensuring that the XP project works successfully. (3)

- b) List the names of the most common agile process models. (3)
- c) Draw and describe the steps of extreme programming. (3)
2. a) What do you mean by quality function deployment (QFD)? Explain the types of QFD requirements. (2+3)

b) Consider the following perfect pizza ordering system and draw a context diagram (Level 0) and Level 1 DFD for it:

Perfect Pizza wants to install a system to record orders for pizza and chicken wings. When regular customers call Perfect Pizza on the phone, they are asked their phone number. When the number is typed into a computer, the name, address, and last order date is automatically brought up on the screen. Once the order is taken, the total, including tax and delivery, is calculated. Then the order is given to the cook. A receipt is printed. Occasionally, special offers (coupons) are printed so the customer can get a discount. Drivers who make deliveries give customers a copy of the receipt and a coupon (if any). Weekly totals are kept for comparison with last year's performance. (2+3)





1. a) What are the FURPS quality attributes? Explain. (3)
 b) Discuss modularity with an example. Why does high cohesion and low coupling generate good software design? Explain with an example. (3)
 c) Construct a use-case diagram for student registration system. (3)
2. a) What are the principal factors that affect the cost of system re-engineering? (3)
 b) What is Requirement Engineering? Define the following terms for requirement analysis:
 i) Closed interview.
 ii) Open interview. (2+2)
 c) Describe different types of non-functional requirements. What is the common problem of non-functional requirements? (2)

PAGE 6

CT #1 (SET-B)

Course Title: Software Engineering and Information System Design (CSTE-3109)

Date: 02/03/2025

Marks: 25, Time: 40 Minutes

1. What do you mean by legacy software? Why must legacy software be changed? (1+4=5)
2. What are the umbrella activities in a process framework? Explain shortly. (6)
3. How should a process pattern be described in software engineering? Explain with an example. (5)
4. Differentiate between the waterfall model and V-model. (3)
5. What do you mean by agility? Discuss the Scrum process model. (2+4=6)

CT #1 (SET-A)

Course Title: Software Engineering and Information System Design (CSTE-3205)

Time: 45 Minutes

1. What do you mean by software? "Software doesn't wear out" - do you agree with the statement? Justify your answer accordingly. (1+3)
2. Discuss process framework activities. (5)
3. Differentiate between linear and iterative process flow with appropriate diagrams. (4)
4. Explain the Extreme Programming (XP) agile process model in detail. (7)
5. What are the differences between the waterfall model and V-model? Explain. (5)

CT #1 (SET-A)

Course Title: Software Engineering and Information System Design (CSTE-3109)

Date: 02/03/2025

Marks: 25, Time: 40 Minutes

1. What do you mean by software? What are the rationales for studying software engineering? (1+4=5)
2. Discuss the failure curve for software. "Software engineering is a layered technology." - do you agree? Answer accordingly. (2+4=6)

3. What are the process improvement activities? Discuss capability maturity levels with an appropriate diagram. (3+3=6)
4. What is the unified process? "There are four phases in each cycle of a unified process." - What are they? (3)
5. Which key traits must exist among the people on an agile team and the team itself? Discuss the XP process model. (2+3=5)

CT #1 (SET-B)

Course Title: Software Engineering and Information System Design (CSTE-3205)

Time: 45 Minutes

1. What do you mean by legacy software? Why must it change? (2+2)
 2. Discuss process umbrella activities. (5)
 3. Differentiate between linear and iterative process flow with appropriate diagrams. (4)
 4. Explain the Scrum agile process model in detail. (7)
 5. What are the differences between prototyping and spiral model? Explain. (5)
-

PAGE 7

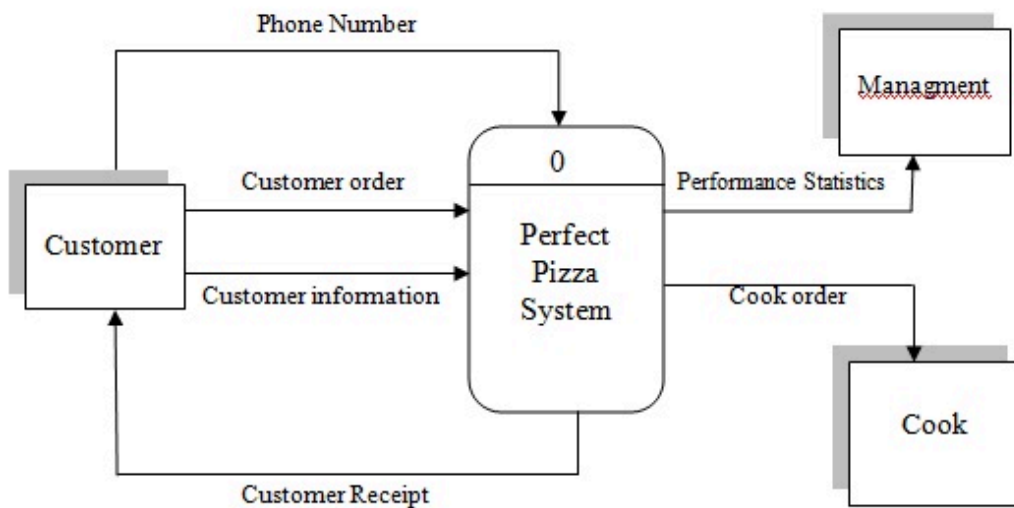
Class Test #4 (Set-A)

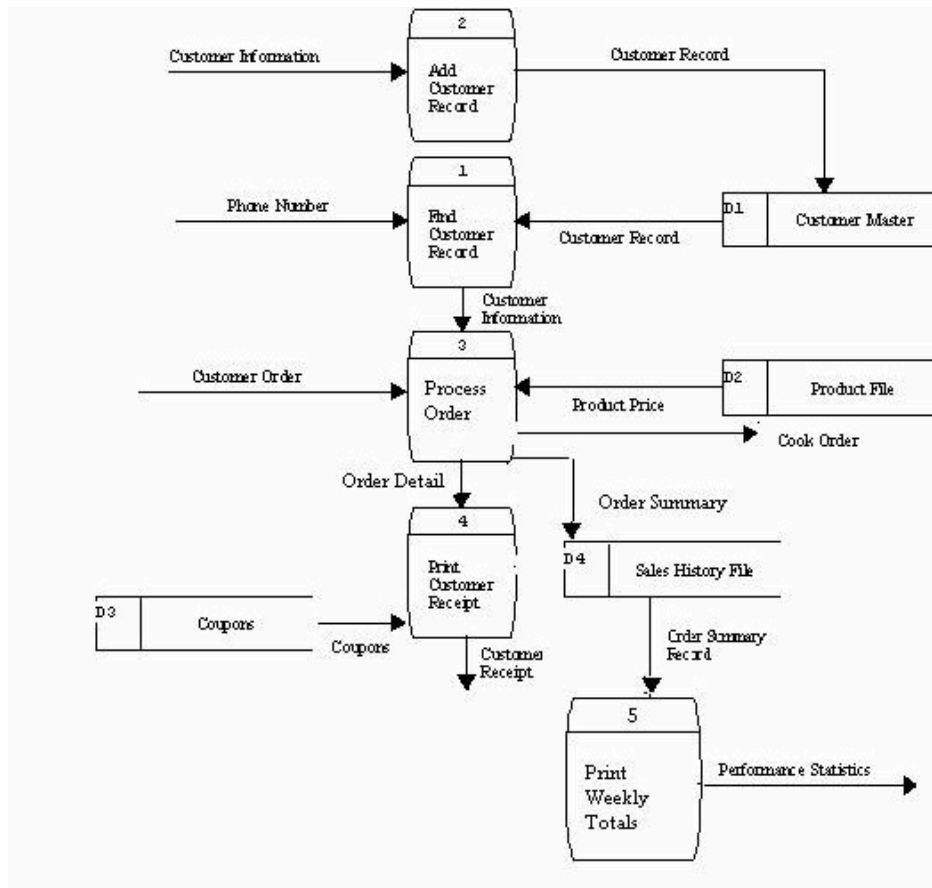
Course Title: Software Engineering and Information Systems (CSTE-3209)

Time: 45 Minutes, Marks: 25

1. What is eliciting requirement? Discuss the types of eliciting requirements. (1+3=4)
2. What do you mean by domain analysis? What are the elements of requirements analysis with a proper diagram? (1+3=4)
3. Show a use-case diagram for an ATM machine. (5)
4. **Consider the following perfect pizza ordering system and draw a context diagram and Level 1 DFD for it:**
Perfect Pizza wants to install a system to record orders for pizza and chicken wings. When regular customers call Perfect Pizza on the phone, they are asked their phone number. When the number is typed into a

computer, the name, address, and last order date is automatically brought up on the screen. Once the order is taken, the total, including tax and delivery, is calculated. Then the order is given to the cook. A receipt is printed. Occasionally, special offers (coupons) are printed so the customer can get a discount. Drivers who make deliveries give customers a copy of the receipt and a coupon (if any). Weekly totals are kept for comparison with last year's performance. (7)





1. Discuss modularity with an example. Why does high cohesion and low coupling generate good software design? Explain with an example. (2+3=5)

Class Test #2 (Set-A)

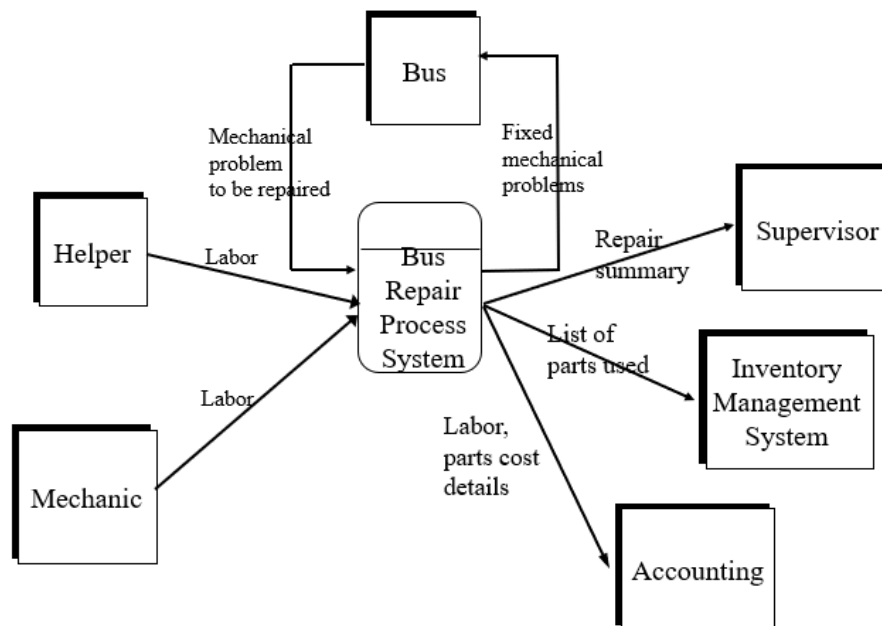
Course Title: Software Engineering and Information Systems (CSTE-3209)

Time: 45 Minutes, Marks: 25

1. What is quality functional requirement? Discuss the types of quality functional requirements. (1+3=4)
2. What do you mean by class-based modelling? How can you identify potential classes to develop class-based modelling? Explain. (1+3=4)
3. Show a use-case diagram for a student registration system. (5)
4. Consider the following bus repair system and draw a context diagram and Level 1 DFD for it:
Buses come to a garage for repairs. A mechanic and helper perform the repair, record the reason for the repair, and record the total cost of all parts

used on a Shop Repair Order. Information on labor, parts, and repair outcome is used for billing by the Accounting Department, parts monitoring by the inventory management computer system, and a performance review by the supervisor. (7)

Bus Garage Context Diagram



1. What are the FURPS quality attributes? Explain. (5)
2. a) What are different types of time under the condition Guidelines of an organization?
 b) What do you mean by: Unit Testing, Data Testing, Regression Testing, Stress Testing.
 c) Mention some Scrum principles. What are the three questions asked in a Scrum meeting? How does XP differ from IXP?
 d) Which approach does DSDM (Dynamic Systems Development Method) follow? Describe the three different iterative cycles of DSDM? (7)

PAGE 8

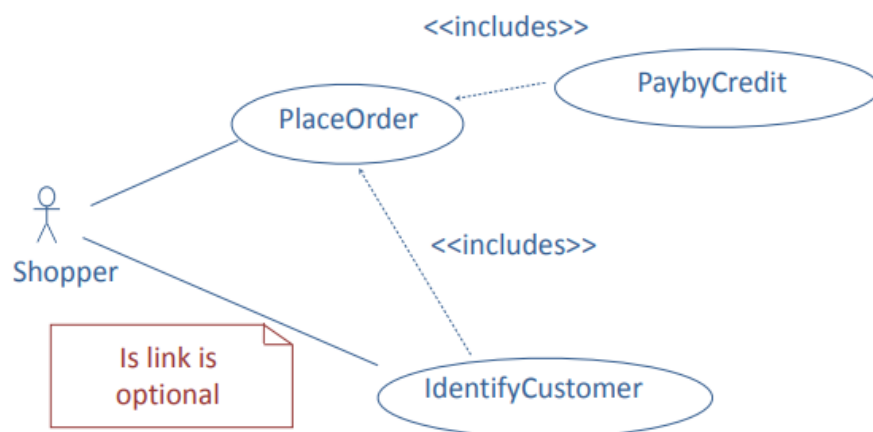
Course Title: Software Engineering and Information System Design (CSTE-3109)

Date: 15/04/2025

Marks: 25, Time: 40 Minutes

1. What do you mean by requirement engineering? Discuss the steps involved in requirement engineering. (1+4=5)
2. **Construct a use case diagram for the following Online Pizza Ordering System:**

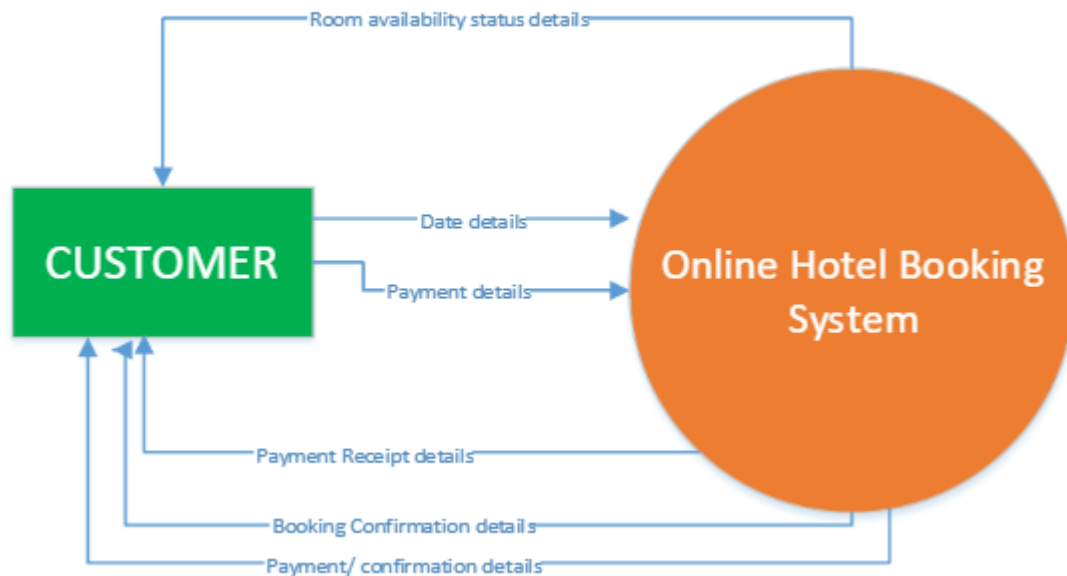
The Pizza Ordering System allows the user of a web browser to order pizza for home delivery. To place an order, a shopper searches to find items to purchase, adds items one at a time to a shopping cart, and possibly searches again for more items. When all items have been chosen, the shopper provides a delivery address. If not paying with cash, the shopper also provides credit card information. The system has an option for shoppers to register with the pizza shop. They can then save their name and address information, so that they do not have to enter this information every time that they place an order. (5)

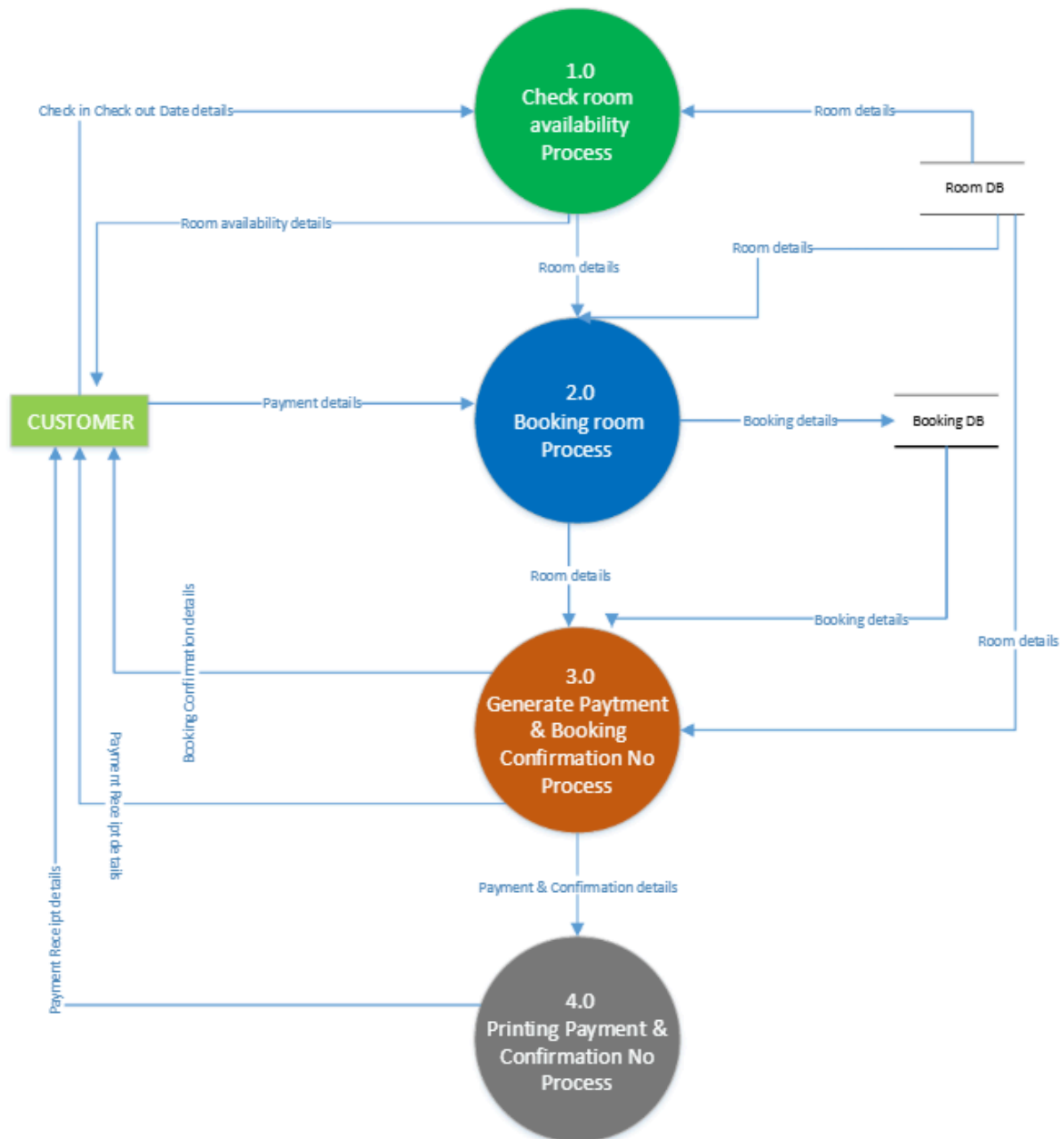


NOT FULL

1. Describe the different ways to manifest analysis classes in software engineering. (4)
2. **Draw Level 0 and Level 1 DFD for the following Hotel Booking System:**
This system will be used by customers who want to book a hotel room in advance. Customers can search for rooms, types, prices, and availability from the Room file. To search, customers must provide check-in and check-out dates. Based on the dates provided, the system checks for the availability of rooms and its specifications. Then, the system provides room status notification to the Customer. If a room is available, the

customer can book the room by making an online payment (via debit or credit card), and it will be stored in the Booking file. If payment is successful, the system will generate the payment receipt, and a booking confirmation number will be given to the Customer. The Customer can print the payment and confirmation details. (2+4=6)





1. What are the basic principles of project scheduling? Explain briefly. (5)

CT #2 (SET-B)

Course Title: Software Engineering and Information System Design (CSTE-3109)

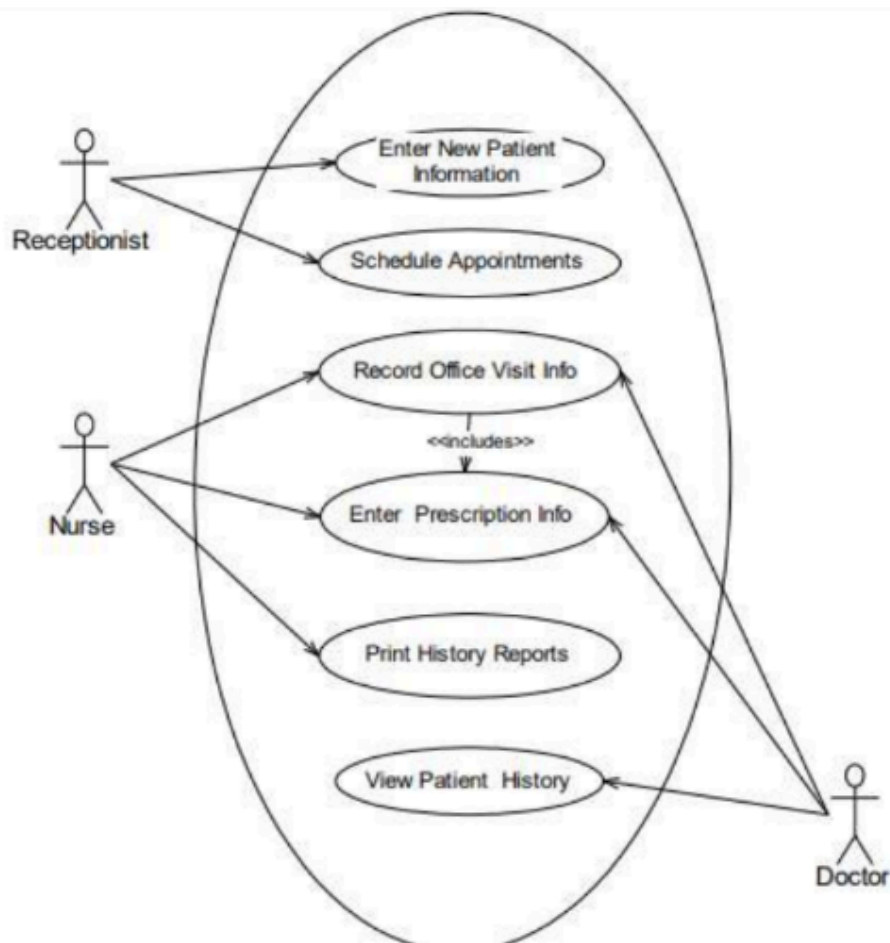
Date: 15/04/2025

Marks: 25, Time: 40 Minutes

1. What are the types of requirements in software engineering? Differentiate between functional and non-functional requirements with examples. (2+3=5)

2. Construct a use case diagram for the following Patient Record and Scheduling System:

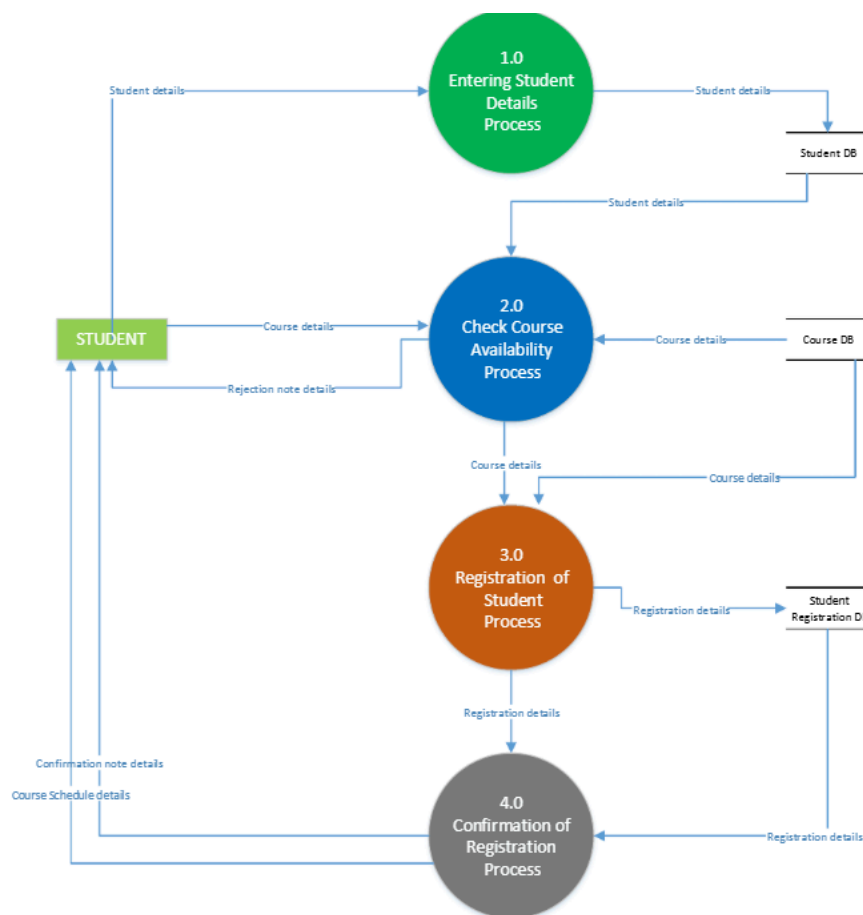
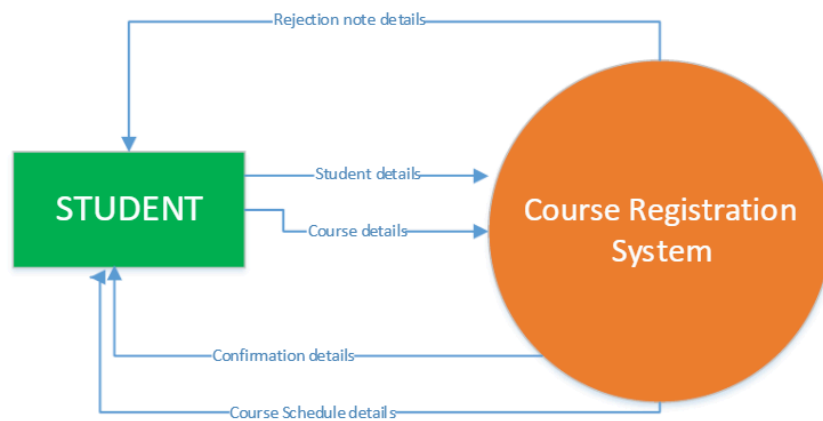
A patient record and scheduling system in a doctor's office is used by receptionists, nurses, and doctors. The receptionists use the system to enter new patient information when first-time patients visit the doctor. They also schedule all appointments. The nurses use the system to keep track of the results of each visit, including diagnosis and medications. For each visit, free-form text fields are used to capture information on diagnosis and treatment. Multiple medications may be prescribed during each visit. The nurses can also access the information to print out a history of patient visits. The doctors primarily use the system to view patient history. The doctors may enter some patient treatment information and prescriptions occasionally, but most frequently, they let the nurses enter this information. Each patient is assigned to a family. The head of the family is responsible for the person with the primary medical coverage. Information about doctors is maintained since a family has a primary care physician, but different doctors may be the ones seeing the patient during the visit. (5)



1. Describe how to identify potential classes for a class model in software engineering. (4)

2. **Draw Level 0 and Level 1 DFD for the following Course Registration System:**

A student wants to register for a course in the Course Registration System. The student sends the Application details containing the student's details and desired course details. The Registration System checks that the course is available from the course file. If the course is available, the student is enrolled in the course, and it will be stored in the Registration file. The Registration System confirms the registration by sending a confirmation note that they are registered for the course to the student with the course schedule details. Or, if the course is unavailable, the student is sent a rejection note. (2+4=6)



1. What are the project scheduling techniques? Explain briefly. (5)

PAGE 9

Course Title: Software Engineering and Information System Design (CSTE-3109)

Date: 14/05/2025

Marks: 25, Time: 40 Minutes

1. "It is the process of managing risks to achieve desired outcomes." - What is it? Explain the steps on how to manage it. (5)
2. What are the guidelines for software quality? Explain. (6)
3. Why do high cohesion and low coupling generate good design? Explain with an example. (6)
4. Differentiate between LOC and FP with proper examples. (4)
5. Discuss any four white box testing techniques with examples. (4)

CT #3 (SET-B)

Course Title: Software Engineering and Information System Design (CSTE-3109)

Date: 14/05/2025

Marks: 25, Time: 40 Minutes

1. "There are several steps to perform the RMMM plan." - Explain them. (6)
2. What are the FURPS quality attributes? Explain briefly. (5)
3. What do you mean by refactoring? Discuss different types of architectural design. (2+4=6)
4. Discuss any four tools and techniques for software cost estimation. (4)
5. Differentiate between incremental and non-incremental testing. (4)