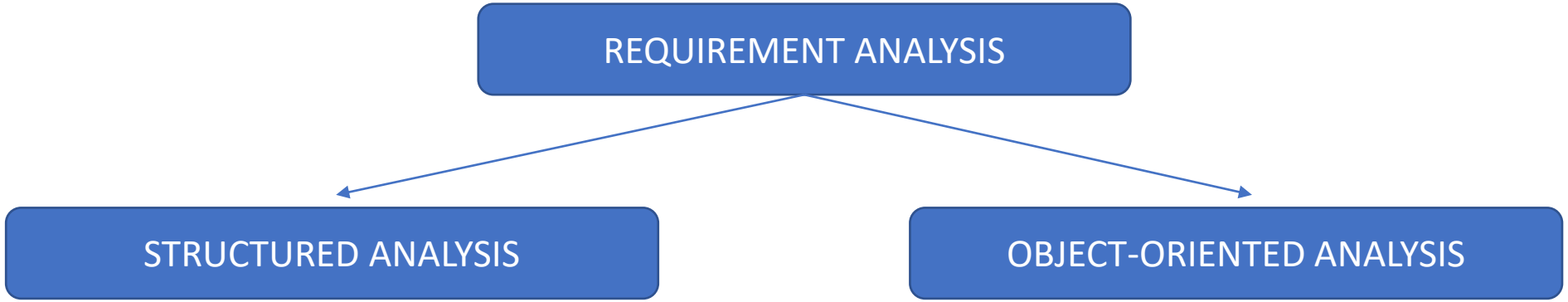
Design and Architectural Engineering

Design and Architectural Engineering

- Characteristics of Good Design
- Function Oriented vs Object Oriented System
- Modularity, Cohesion, Coupling, Layering
- Design Models
- UML (Unified Modeling Language)

Modeling Techniques with Examples

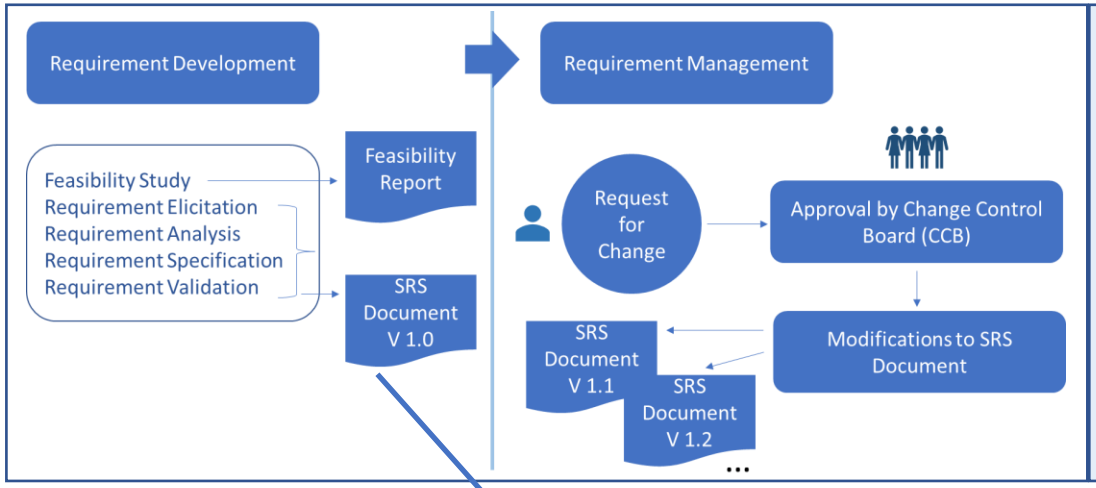


- **Data Modeling** - Entity Relationship (ER) Diagram
- **Functional Modeling** – Data Flow Diagram, Control Flow Diagram
- **Behavioral Modeling** – State Transition Diagram

- **Static Modeling** - Class Diagram, Object Diagram, Use Case Diagram
- **Dynamic Modeling** – Sequence Diagram, Collaboration Diagram, State Transition Diagram
- **Functional Modeling** – Data Flow Diagram

In this session, we will discuss specific examples with details.

Problem Space and Solution Space



- Problem Space ('WHAT' is required or 'WHAT' problem needs to be solved)
- At this stage, decisions related to design and implementation are avoided.

Design and Architectural Engineering

- Solution Space ('HOW' will the requirements be implemented or problem be solved)
- At this stage, decisions related to design and implementation are made.

Challenges in Software Design

1. An error in requirements will carry forward as a design error.
2. Mapping each functional requirement to a design representation requires competency and experience.
3. Creating simple and easy-to-understand designs that can help developers write correct code is difficult.
4. Keeping the design updated by making changes in order to accommodate the changes in requirements (SRS).
5. Balancing the quality attributes (simplicity vs. efficiency, clarity vs. code safety, robust design vs. performance)
6. Error and exception handling

Characteristics of a Good Design

- **Completeness** - Should convert all requirements into design. Should not have gaps.
- **Understandable and Maintainable** – Should be simple and easy to understand and maintain. Should not be cryptic.
- **Ease of Change** – Should be easy to change (make minor adjustments or significant changes). Should not be rigid.
- **Scalable** – Should be easily scalable (to accommodate new features or modules).

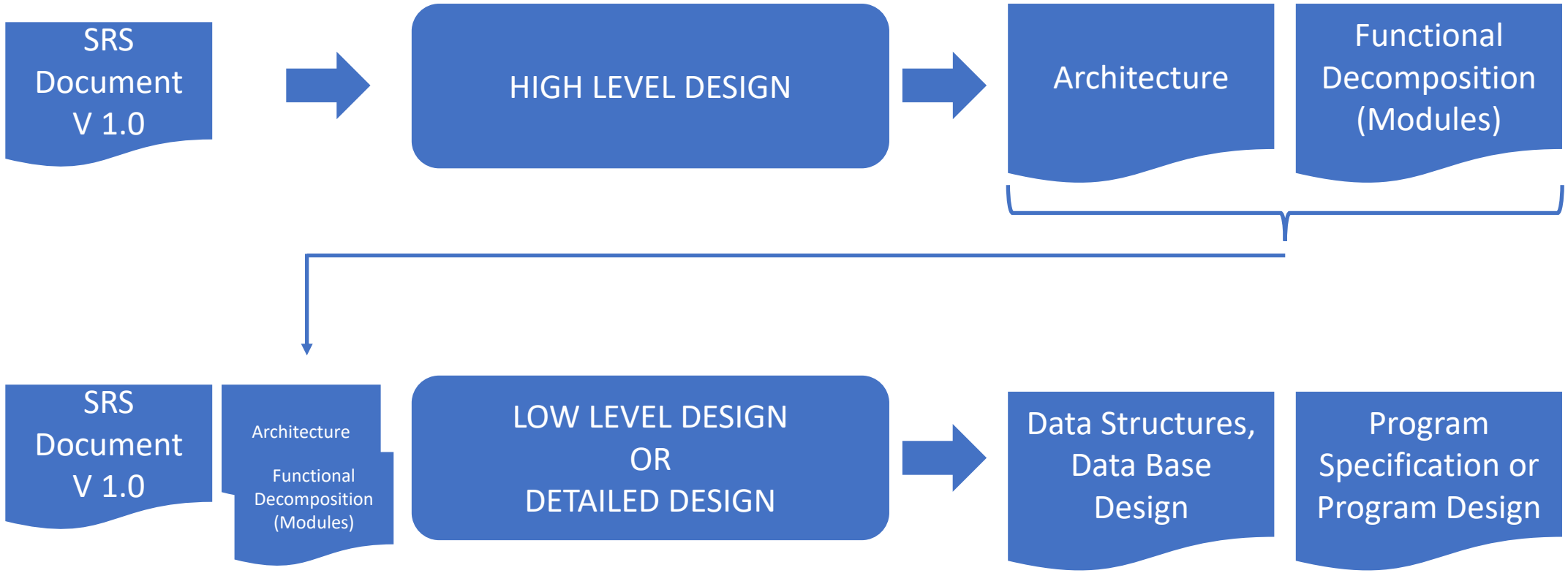
Question 1

Software Design is a challenging activity because,

A. Mapping each functional requirement to a design representation and balancing the quality attributes (for example, clarity vs. code) are very challenging
B. It requires diagramming techniques to depict design model
C. It takes several weeks or months to perform design
D. Once a design is created, reviewed and approved, it cannot be modified.

ANSWER: A

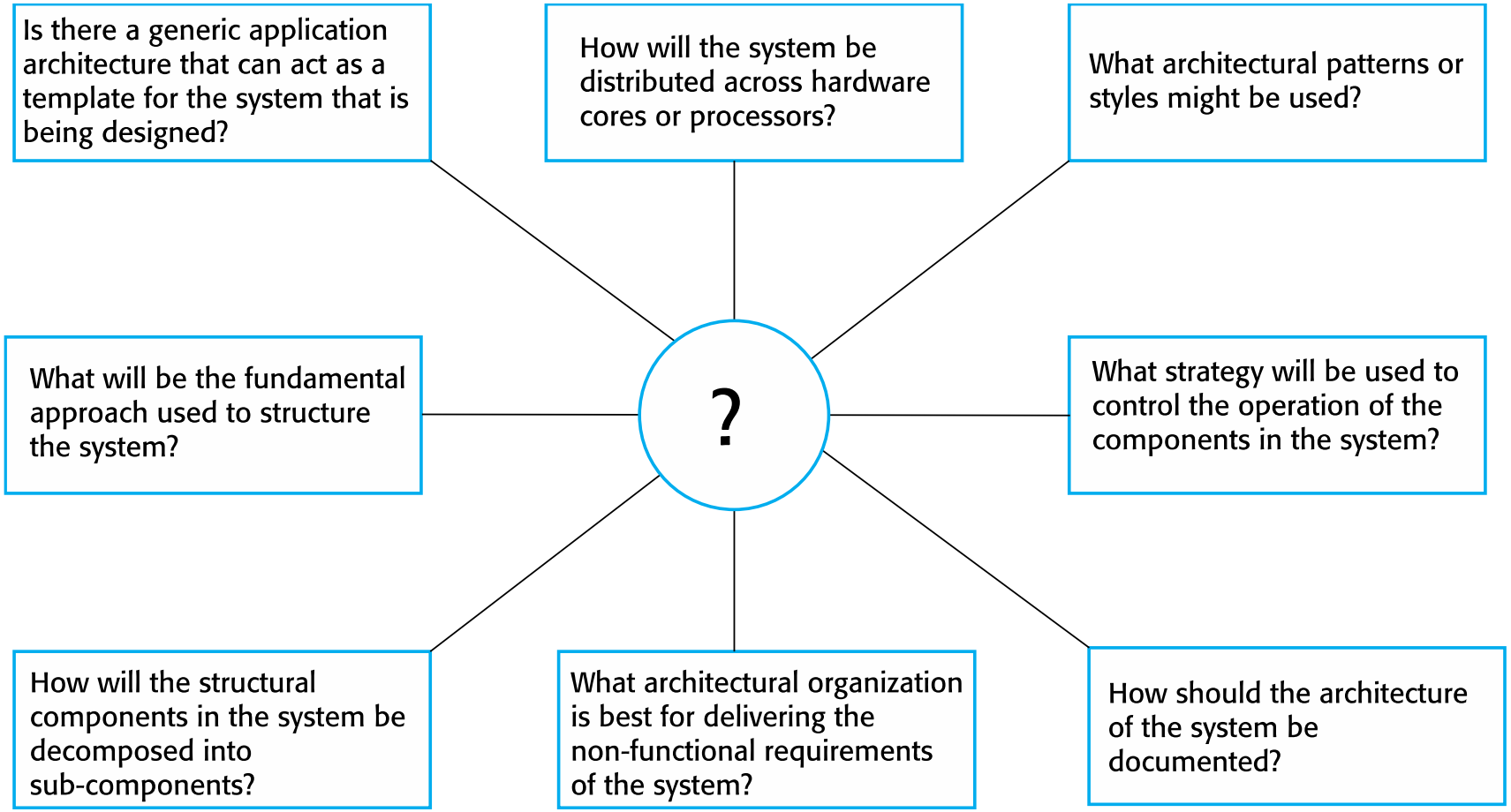
Design Activities



Architectural Design

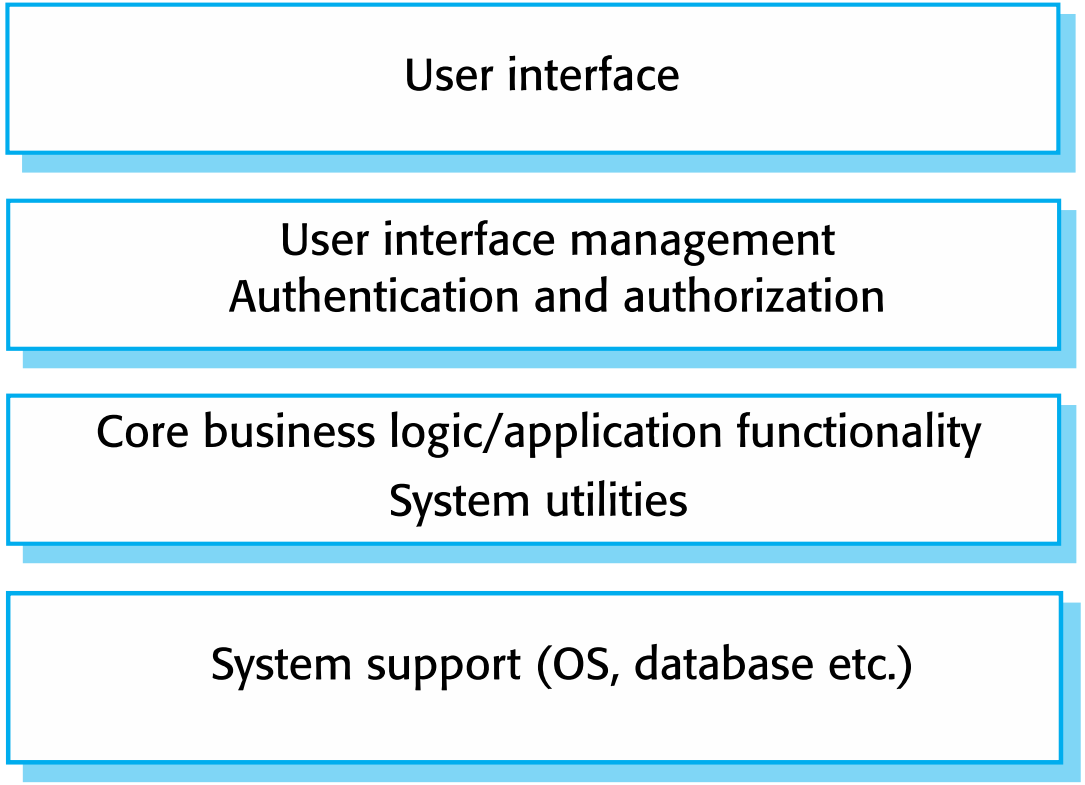
- Is concerned with understanding how a software system should be organized and designing the overall structure of that system.
- Is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them.
- The output of the architectural design process is an architectural model that describes how the system is organized as a set of communicating components.

Architectural Decision Making

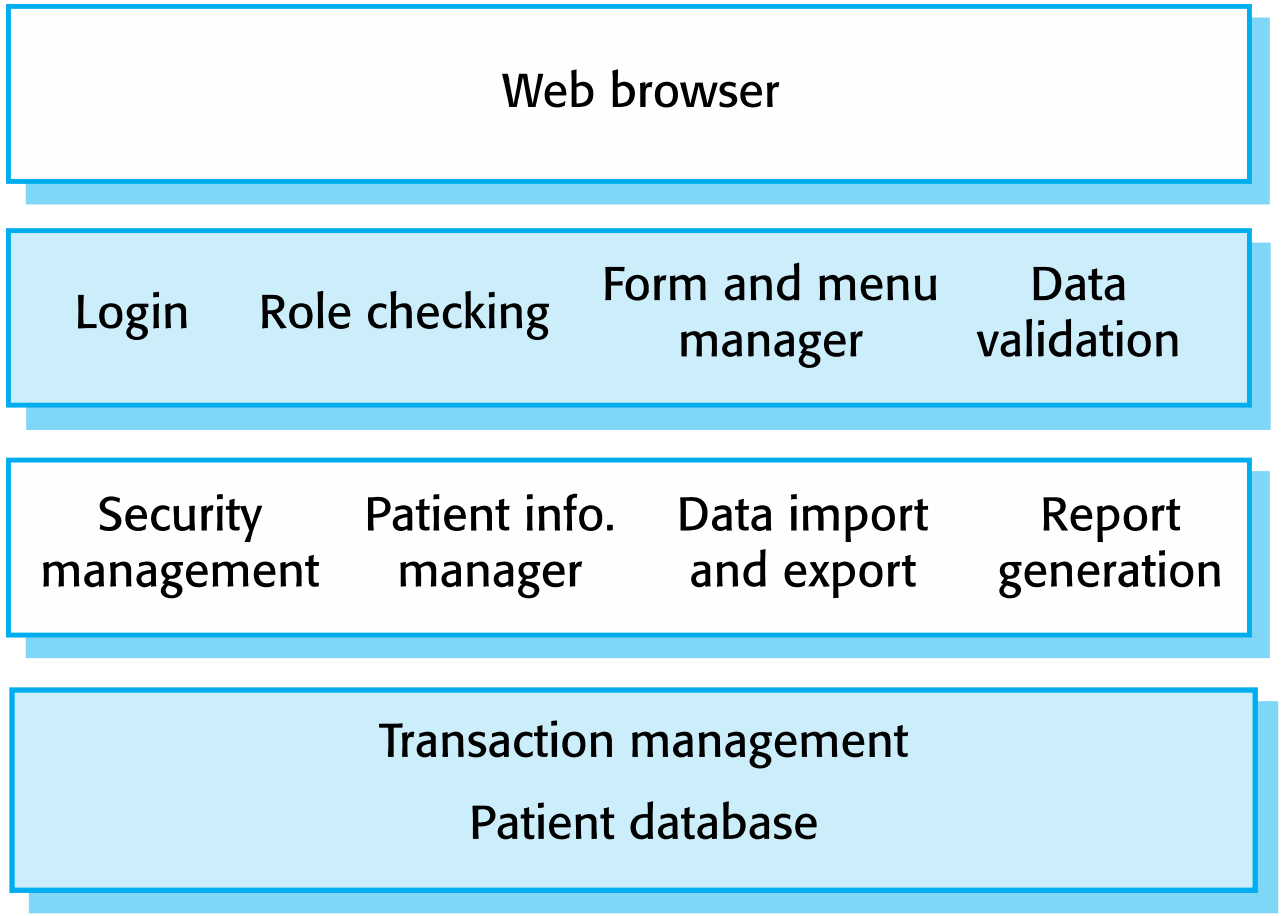


Layered Architecture

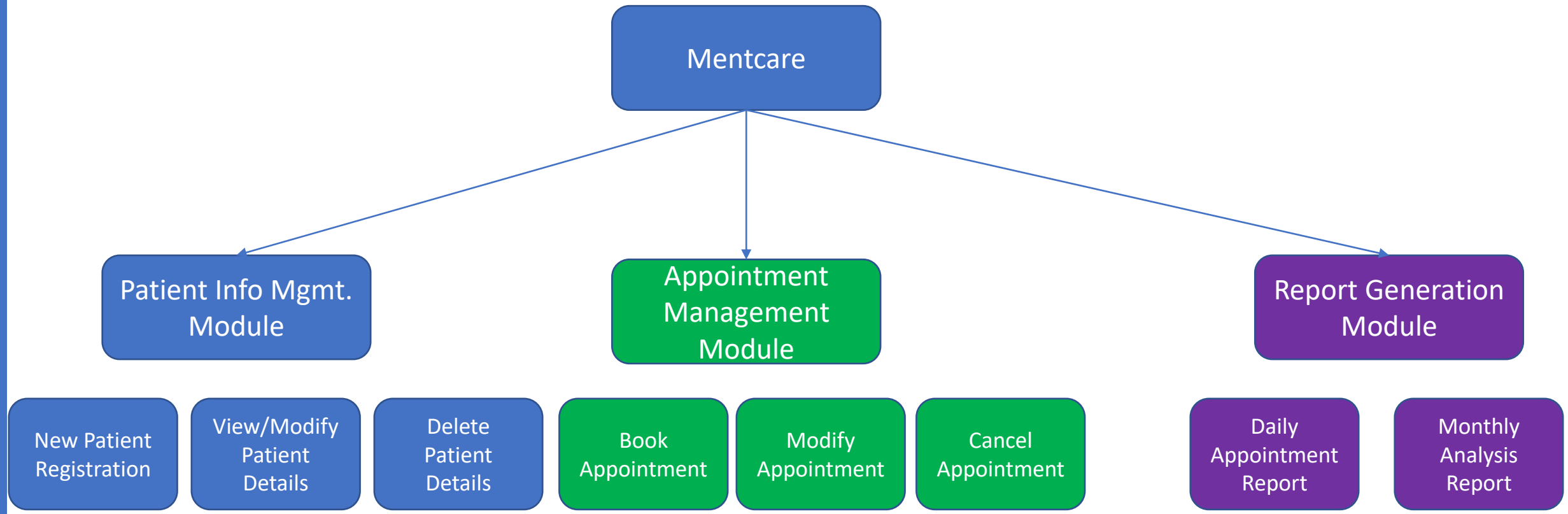
- Organises the system into a set of layers each of which provides a set of services.
- Supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.



Architecture of the Mentcare System



Functional Decomposition (Mentcare)



Modularity

The possibility of designing a system as a collection of multiple modules is known as modularity.

A module provides one or more related functions or features as specified in the SRS document.

Example: Functional decomposition and modular design of Mentcare system

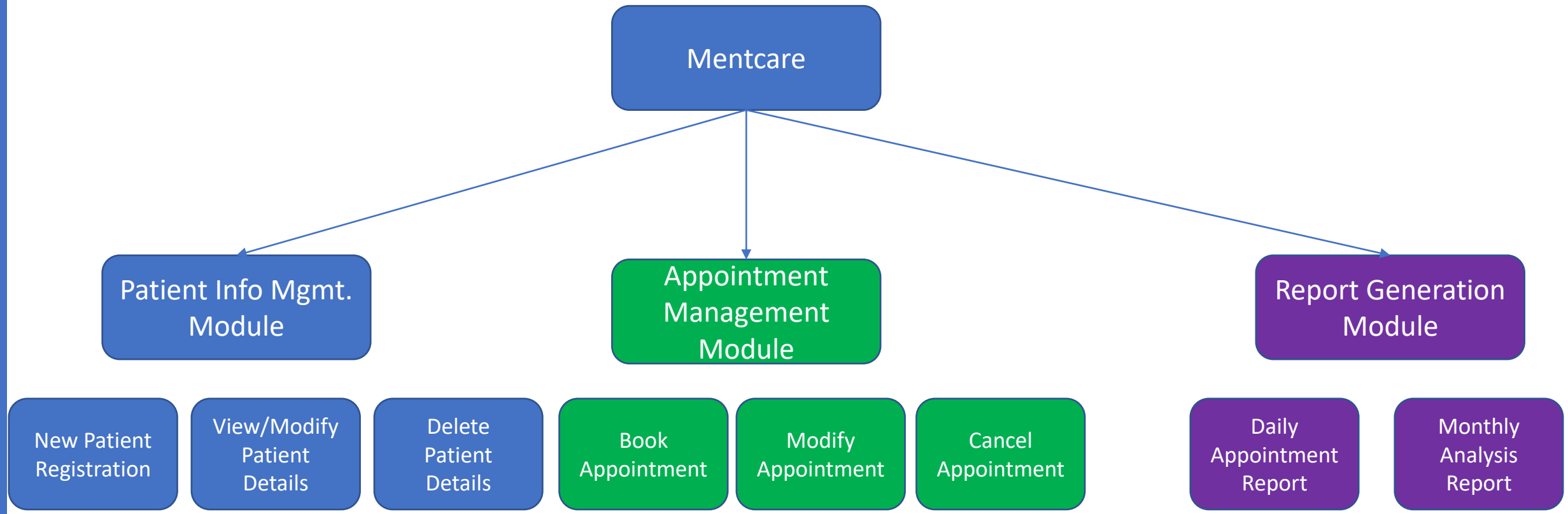
Cohesion

- A module provides one or more related functionalities.
- Cohesion is a measure of ‘how strongly’ the elements of a module are functionally related.
- If the elements are strongly related, then the module is highly cohesive.
- A good software design will have high cohesion.

Read these real life examples. Comment on the cohesion level of these ecosystem (High/Medium/Low).

1. A housing society with flats occupied by owners or tenants for living
2. A housing society where some of the flats are occupied by a retail business, a large catering service, and a hospital
3. An university that has student hostels, faculty residents, class rooms, labs, research infrastructure, sports facility, parking lots, swimming pool, bank, super market, and a food court
4. A large hospital comprising of wards, doctor rooms, medical shop, staff quarters, ATM and parking lots.

Modules with a high level of cohesion



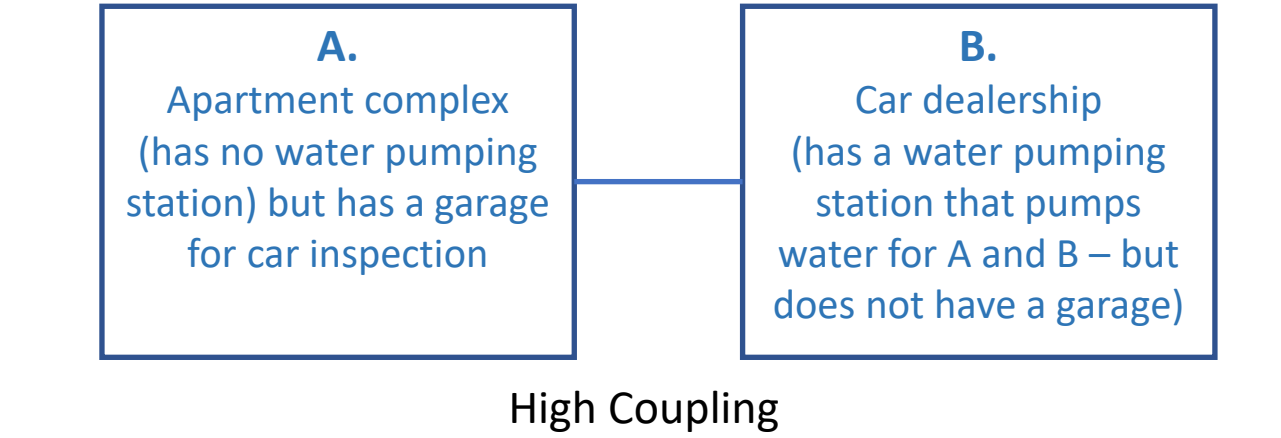
There are different types of cohesions. The best one is 'Functional Cohesion' where all functions in a module are related to each other.

Coupling

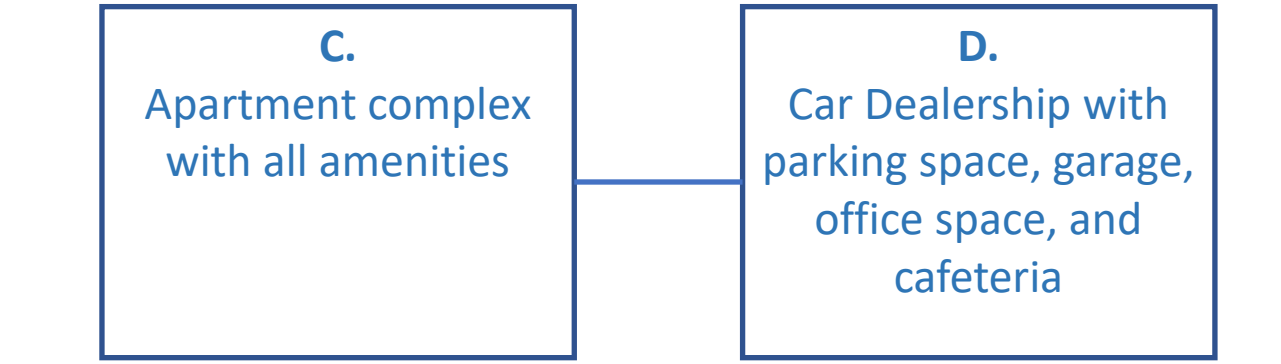
Coupling is the measure of the degree of interdependence between the modules.

A good software will have modules or components with low coupling.

Real life examples.



High Coupling



Low Coupling

Question 2

Which of the following is not considered as one among the characteristics of a good design?

A. Clarity and Standardization - Should include all types of standard design diagrams and notations. Should not have any explicit conditions.
B. Understandable and Maintainable – Should be simple and easy to understand and maintain. Should not be cryptic.
C. Ease of Change – Should be easy to change (make minor adjustments or significant changes). Should not be rigid.
D. Scalable – Should be easily scalable (to accommodate new features or modules).

ANSWER: A

Question 3

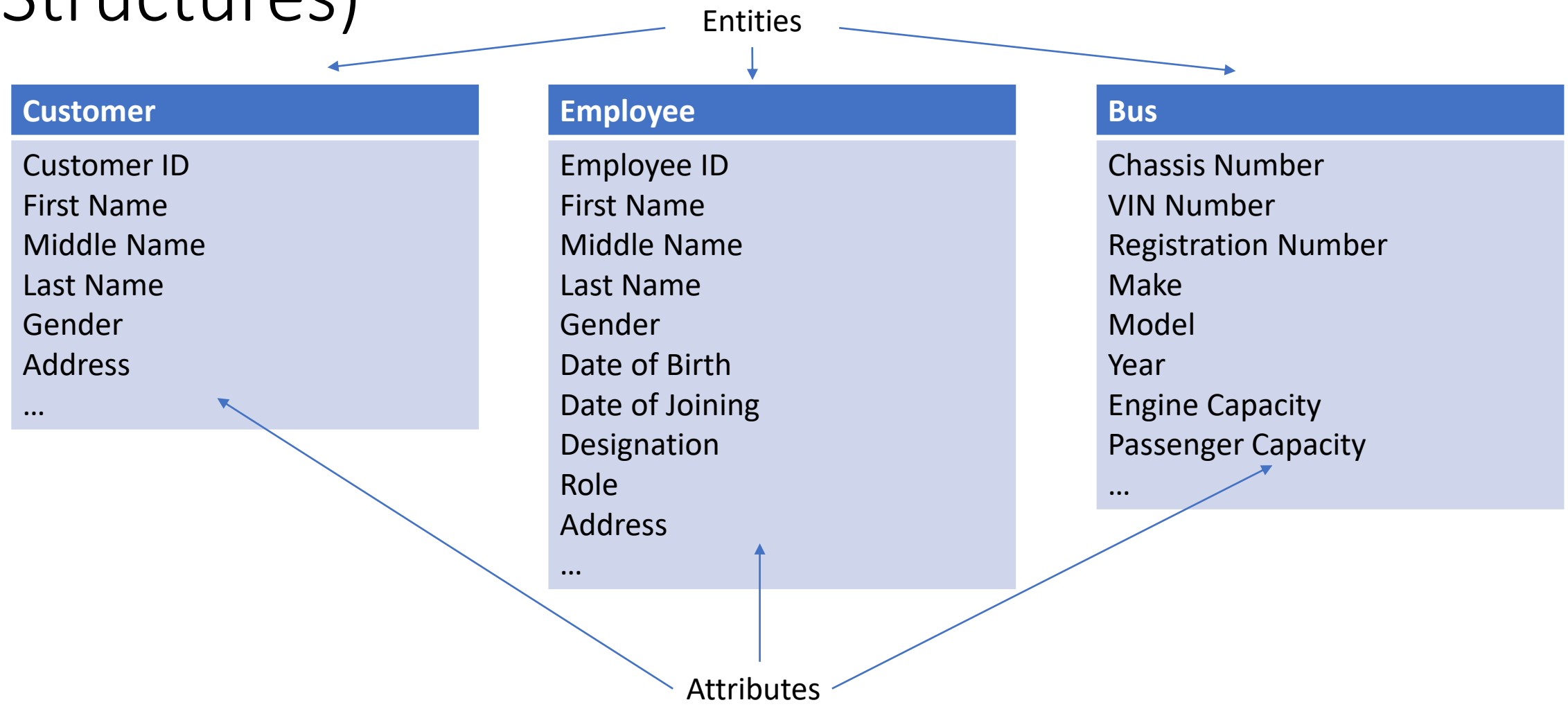
Which is a measure of ‘how strongly’ the elements of a module are related to each other?

A. Coupling
B. Layering
C. Modularity
D. Cohesion

ANSWER: D

Design Models

Data Design (Entities and Attributes + Data Structures)



Customer Table Definition in a Relational DB

Attributes	Type & Length	Constraint
Customer ID	Number (8)	Not Null
First Name	Character (30)	Not Null
Middle Name	Character (30)	Not Null
Last Name	Character (30)	Not Null
Gender	Character (10)	Not Null
Address Line1	Character (30)	Not Null
Address Line2	Character (30)	
City	Character (30)	Not Null
District	Character (30)	Not Null
State	Character (30)	Not Null
Pin Code	Number (6)	Not Null
Mobile Number	Number (10)	Not Null
…		

Architectural Design

Software architecture includes all major structural elements (such as layers, components) and relationship between them at a high level.

System architecture includes the physical components (computers, disks, devices, things) along with underlying software and services (such as OS, RDBMS, Web Server, Application Server, Compilers, Cloud Services).

User Interface (UI) Design

Involves creation screen layout which is a common element of user interface.

Focuses on

1. How UI components interact with the system (UI Components: Buttons, Pull Down Menu, ...)
2. How the system interacts with humans (E.g., conformation message, error messages, help, …)
3. How the system interacts with other system through the UI

UI – 3 Golden Rules

1. **Place the users in control** (provide customizable screen based on user profile and preferences, flexibility (more than one way of interacting with the system, interactive display). When a system provides all these to a user, the users start liking the system.

2. **Reduce users' memory load** – Make it easy with visual cues, clarity , short cuts etc. (so that users need not remember where they started and how far they have navigated in the system).

3. **Keep it consistent** (so that all modules provide the same level of look and feel, responses, interaction and navigation experience)

Procedural Design

- Also known as component design.
- It is based on process specification (PSPEC) or control specification (CSPEC)
- Usually done after interface design.

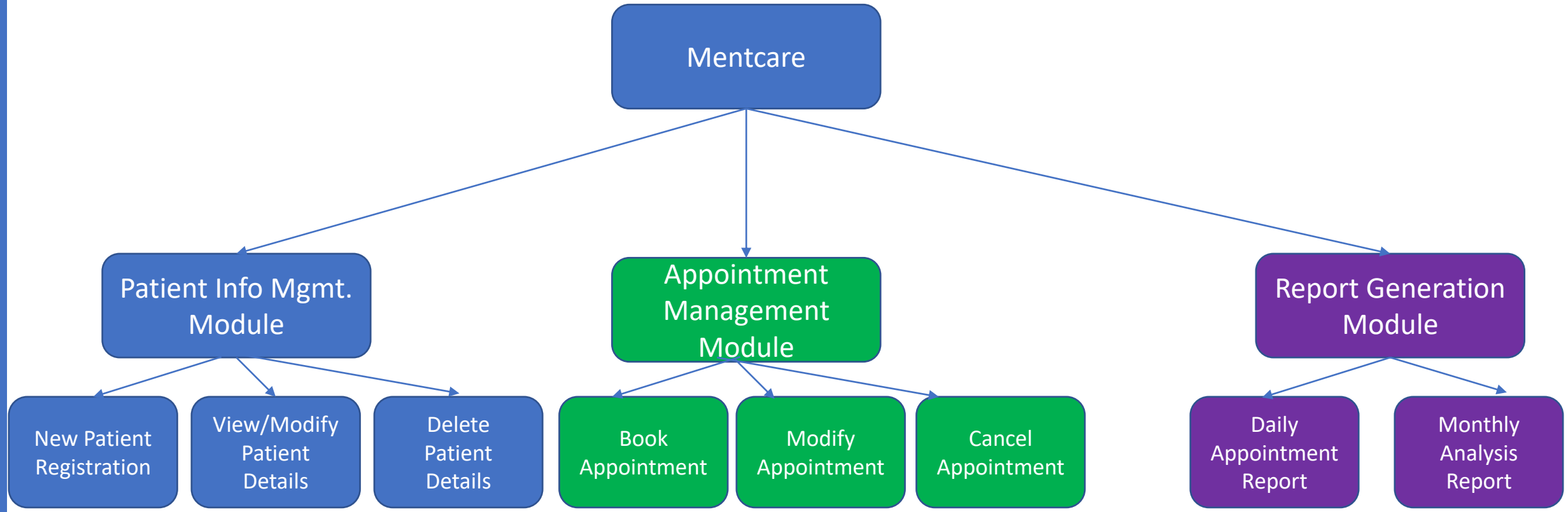
Examples:

1) Calculate ‘No Claim’ discount while renewing a policy (PSPEC)
2) Check if authentication fails 3 times and change the user state to ‘Blocked’ (CSPEC)

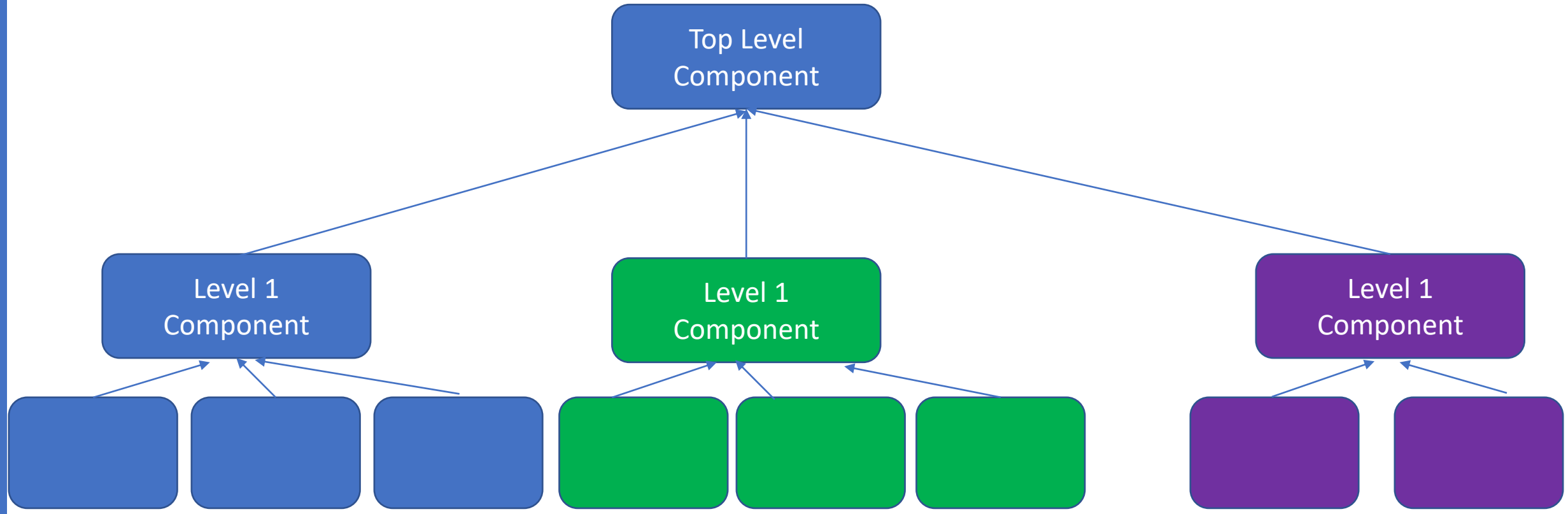
Function Oriented & Object Oriented Systems

Function Oriented System	Object Oriented System
The focus is on real world functionalities	The focus is on real world objects
Data and functions are separate	Objects hold both data and functions (methods) associated with “Class”
Functions are grouped together. Functional decomposition of the system provides a clear view.	Functions are grouped together based on the objects they are related to and data they operate. Classes (abstraction of objects) are associated with their methods.
Top down approach	Bottom up approach
We start with identifying functionalities or features or use cases	We start with understanding the context and identifying objects

Top Down Design



Bottom Up Approach



Question 4

Which of the following is a measure of interdependency between two modules?

A. Coupling
B. Cohesion
C. Modularity
D. Layering

ANSWER: A

Question 5

Which of the following is a desired factor of a good design?

A. High Coupling, High Cohesion
B. High Coupling, Low Cohesion
C. Low Coupling, Low Cohesion
D. Low Coupling, High Cohesion

ANSWER: D

Design Documentation

High Level Design -> System Architecture Document (SAD)

Low Level Design -> Detailed Design Document (consists of data structures and program specifications)

UML (Unified Modeling Language)

1. UML is a modeling language in the field of software engineering.
2. Developed at Rational Software (1994-95) with an objective of standardizing and unifying different types of notational systems and approaches to design.
3. Object Management Group adopted UML as a standard in 1997.
4. UML is not a development methodology. UML was designed to be compatible with Object-Oriented Methodologies (E.g., Object Modeling Technique, Booch Method, Objectory, Rational Unified Process(RUP))
5. Used in projects that follow Object-Oriented Analysis and Design

Question 6

Which of the following is not a model in structured analysis?

A. Entity Relationship (ER) Diagram
B. Sequence Diagram
C. Control Flow Diagram
D. Data Flow Diagram

ANSWER: B

Question 7

Top down approach to system design is followed

A. In the design of function oriented systems
B. When we represent the hierarchical relationship of modules or functions
C. To describe the top-down organization of software components or modules
D. All of the above

ANSWER: D

Question 8

Which of the following is true about UML?

A. UML is a development methodology. UML was invented for both Function Oriented and Object Oriented Systems.
B. UML is not commonly used in projects that follow Object-Oriented Analysis and Design
C. UML is not a development methodology. UML was designed to be compatible with Object-Oriented Methodologies (E.g., Object Modeling Technique (OMT))
D. UML is not approved as a standard by Object Management Group (OMG)

ANSWER: C

Question 9

Which of the following is not one among the golden rules of UI design?

A. Place the users in control
B. Reduce users' memory load
C. Perform validations in the UI
D. Keep it consistent

ANSWER: C

Question 10

Typically, the architecture of web applications comprises of three tiers. Tier 1 is the bottom most level and it includes data base (data structures and user data). Tier 2 is at the next level and it includes application server, web server and related components. Tier 3 is the top tier that has user interface delivered on a web browser. This is an example of

A. Tightly coupled architecture
B. Layered architecture
C. Tiers that are highly cohesive with no interdependency
D. Modern architecture

ANSWER: B

Summary

- Characteristics of Good Design
- Function Oriented vs Object Oriented System
- Modularity, Cohesion, Coupling, Layering
- Design Models
- UML (Unified Modeling Language)

Thank You!