Name : BAZGHA RAZI

College : DGIT

Course & Semester : B.Tech (CSE) , 5th Semester

Subject Code : PEC-CSE-311G

Subject Name : Software Engineering

Total pages : 12

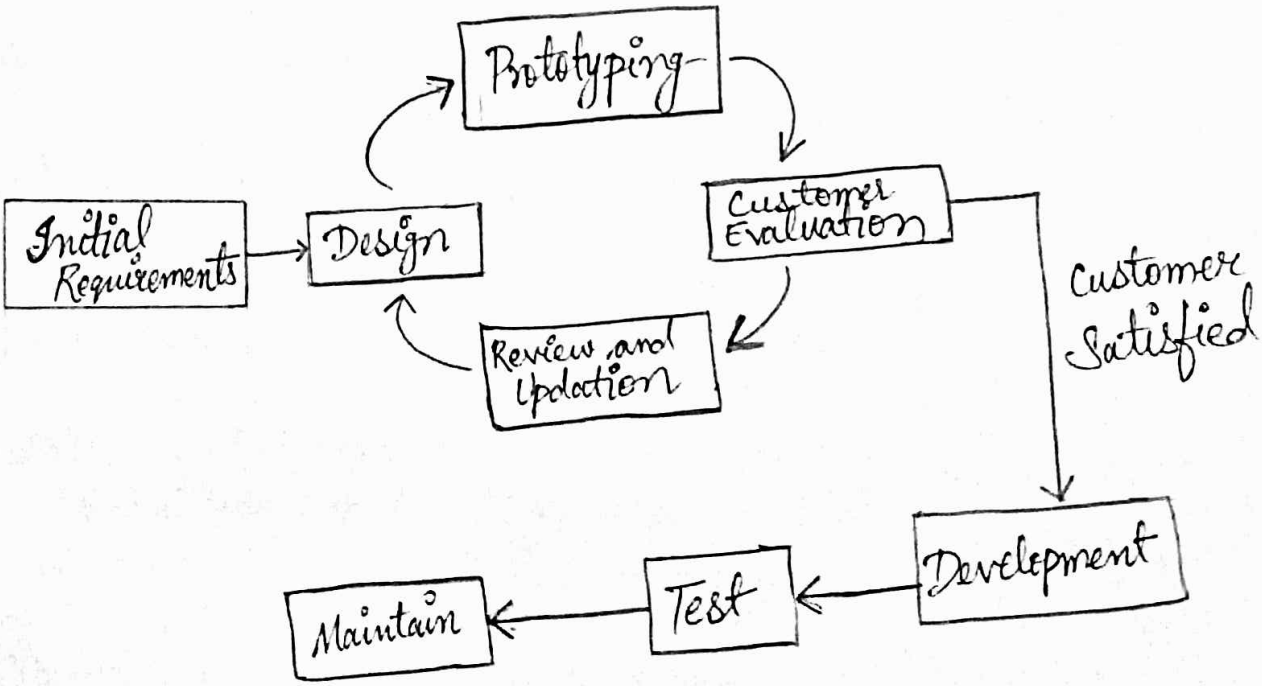
Date of Submission : 12/12/2021

Signature : Razgha Razi

Signature: Farzana Fazli

Ans 1 Prototyping Model

A prototyping model can be used when technical solutions are unclear to the development team. A developed prototype can help engineers to critically examine the technical issues associated with the product development. Often, major design decisions depend on issues like the response time of a hardware controller, or the efficiency of a sorting algorithm, etc. In such circumstances, a prototype may be the best or the only way to resolve the technical issues.



The prototype may be a usable program but is not suitable as the final software product. The code for the prototype is thrown away. However experience gathered helps in developing the actual system. The development of a prototype might involve extra cost, but overall cost might turnout to be lower than that of an equivalent system developed using the waterfall model. The effect of designing a prototype on the overall cost of a software projects is to actually reduce the additional costs of restructing and reframing it after its full-fledged development which might cost a fortune.

Ans 2 Desirable characteristics of a good SRS document:

a) SRS should be complete. It defines precisely all the live situations that will be encountered and the system's capability to successfully address them.

b) The logical, hierarchical structure of the SRS should facilitate any necessary modifications and that too with is greater ease.

c) SRS should be consistent. SRS capability functions and performance levels are compatible and the required quality features do not negate those capability functions.

d) A valid SRS is one in which all parties and project participants can understand, analyze, accept or approve it.

e) A verifiable SRS is consistent form one level of abstraction to another.

f) SRS must contain requirements statements that can be interpreted in one way only i.e., it should be unambiguous. This is another area that creates significant problems for SRS development because of the use of natural language.

g) SRS must be stated in such a manner that unambiguous assessment criteria can be derived from the SRS itself.

h) Each requirements in SRS must be uniquely identified to a source.

i) SRS precisely defines the system's capability in a real-world environment, as well as how it interfaces and interacts with it. This aspect of requirements is a significant problem area for many SRSs.

 Signature: Razia Razi

Ans3: Software Engineering is composed of two words. software and engineering.

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be a collection of executable programming code, associated libraries and documentation. Engineering, on the other hand is all about developing products, using well-defined, scientific principles software product.

So, software engineering as an engineering branch associated with the development of software product using well-defined scientific principles, methods procedures.

IEE defines software engineering as, the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software.

The need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working.

Signature: Fazila Rafi

Characteristics of Good Software are as follows:

A software product can be judged by what it offers and how well it can be used. This software must satisfy on the following grounds:

- Operational: The tells how well software works in operations. It can be measured on budget, usability, efficiency, correctness, functionality, dependability, security and safety.
- Transitional: This aspect is important when the software is moved from one platform to another. portability, interoperability, reusability and adaptability.
- Maintenance: This aspect briefs about how well a software has the capabilities to maintain itself in the everchanging environment modularity, maintability, flexibility and scalability.

Software engineering is a branch of computerscience, which uses well-defined engineering concepts required to produce efficient, durable, scalable, in-budget and on-time software products.

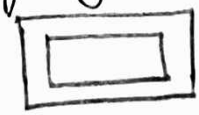
Signature: Farzana Kazi

Ans 4: Various notations are used in ER diagram are as follows:

i) Entity Type: An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

Symbol: 

ii) Weak Entity Type: An entity type that don't have any key attribute is called weak entity type. The weak entity type is also called the child entity type or the subordinate entity type.

Symbol: 

iii) Relationship Type: Which connect entities and represent meaningful dependencies between them. The way in which two or more entity types are related is called relationship type. Ex - "Enrolled in" is a relationship type that exists between entity type student and course.

Symbol: 

Signature: Fazila Kazi

iv) Attribute: It gives the characteristics of the entity. In other words, every entity has some attributes that characterize it.
Ex: A house can be described by its size, color, etc.

Symbol:

v) Key Attribute: We have define a key attributes as an attributes that have distinct value for each entity in an entity set. Ex - Roll no. in a student entity type is a key attribute.

Symbol:

vi) Multivalued Attribute: An attribute which have a set of values for the same entity is known as multivalued attribute. Ex - colors for a car entity.

Symbol:

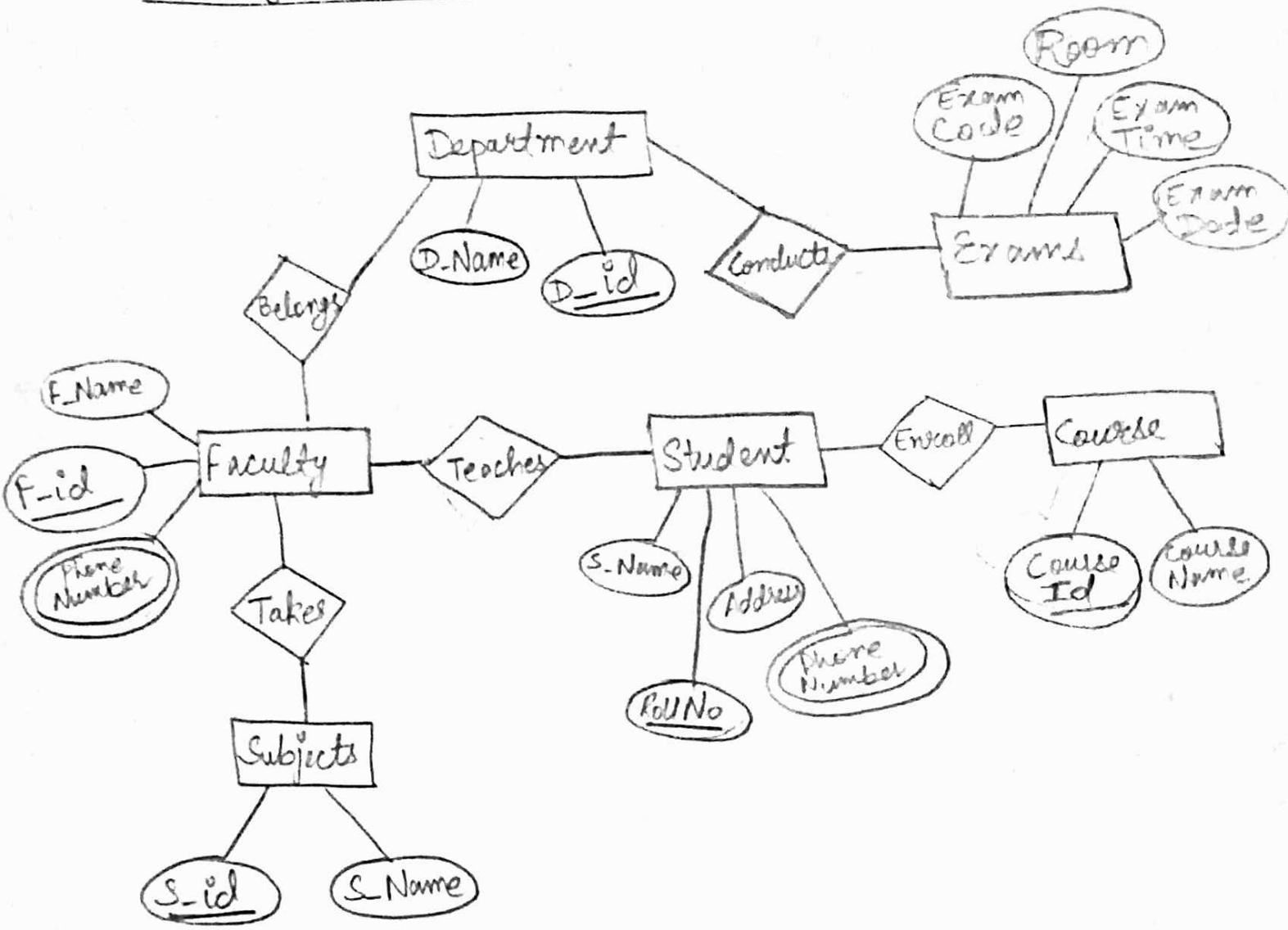
vii) Derived Attribute: An attribute which can be derived from another attribute is known as derived attribute.

Symbol:

Page No: 8

Signature: Fazila Rafi

ER Diagram for College



Signature: Farzana Kazi

Ans5: Various notations and rules used in DFD.

Notations of DFD

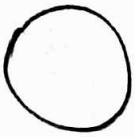
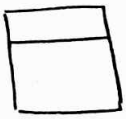
i) External Entity : It is also known as actors, sources and terminators, external entities produce and consume data that flows b/w the entity and the system being diagrammed.

Symbol : 

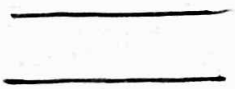
ii) Data Flow : Movement of data b/w external entities, processes and data stores is represented with an arrow symbol, which indicates the direction of flow.

Symbol : $\longrightarrow$

iii) Process : An activity that changes or transforms data flows.

Symbol :  or 

iv) Data Store : It doesn't generate any operations but simply holds data for later access. Input flows to a data store include information that change the stored data. Output flows would be data retrieved from the store.

Symbol:  or 

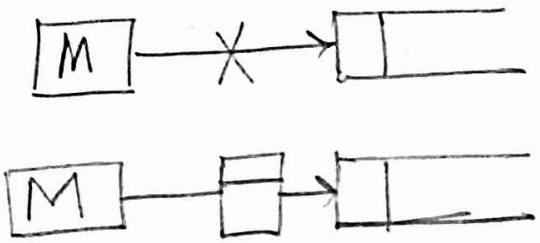
 Signature: fazgha Kazi

Rules in DFD building

Rule 1: Use only DFD notations to avoid confusion.

Rule 2: Use an action VERB to label a process.

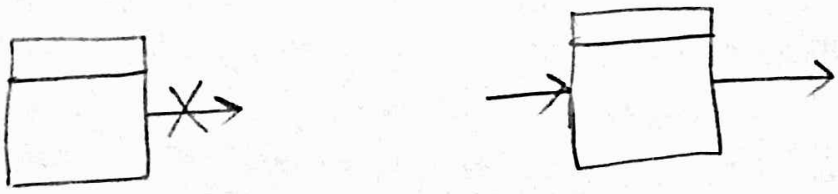
Rule 3: Must be one process associated with each data flow.



Rule 4: Shaded corner must appear in ALL occurrences of a duplicated symbol in a same diagram.

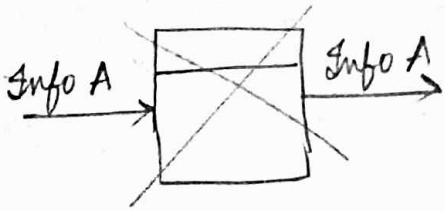
Rule 5: No process without output data flow.

Rule 6: No process without input data flow.



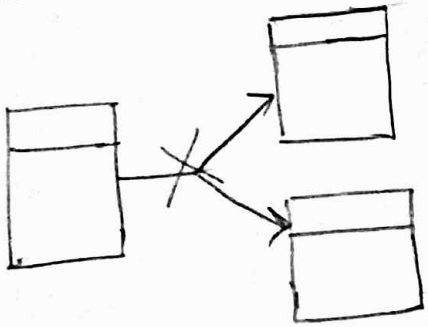
Signature: Fazgha Rasi

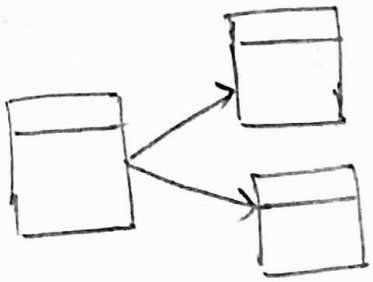
Rule 7: No need for routing (without transforming) a data flow with a process (non-value-added activities).



Rule 8: Identical input, output data flows for parent and child processes.

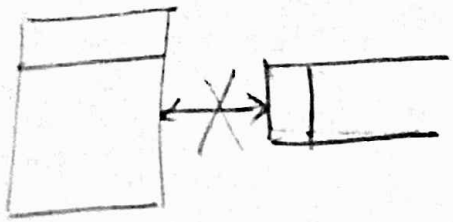
Rule 9: Data flows cannot split by themselves.

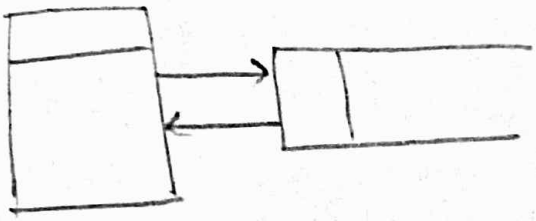




Rule 10: A data packet can combine many data elements being transmitted at the same time to the same destination.

Rule 11: Double-headed arrows are forbidden.





Signature: Fazgha Razi

DFD for Library Management System