Shivank Pandey
21BKT0021

Software Engineering
DA-2

Q1) When should a modular design be implemented as a monolithic software? How can this be accomplished? Is performance the only justification for implementation of monolithic software?

A1) Modularity refers to breaking down a system into smaller, interchangable components or modules, fascilating easier managment, maintainence and scalablity by promoting reusablity in design and implementation. Monolithic Software refers to an application architecture where all components are tightly integrated into a single program, lacking modularization and typically requiring the entire system to be deployed together.

A modular design might be implenented as monolithic software when system's component are highly interdependent and don't benefit from being used as distinct modules. This can be suitable for small-scale application with simple functionality, where the over head of manging seperate modules outweight the benifits of modularity.

This can be accomplished by structuring the ~~co~~ codebase as a single, cohesive unit without explicit modular boundaries ensuring tight integration between components. Additionally developers can avoid using technologies or framework that encourage modularazation, focusing ahead on simplicity and minimising overhead.

Shivank Pandey

21BKT0021

Date ______
Page ______

→ No, beside performance, another justification for implementing monolithic software could be simplicity in development, deployment & maintainence, especially for smaller application with limited functionality and whose modularity doesn't offer significant advantages.

Shivank Pandey
21BKT0021

Software Engineering
DA-2

Q2) Using a data flowdiagram and a processing narrative, describe a computer-based system, that has distinct transform flow characterstics. Define flow boundaries ~~for the~~ and map the DFD Software architecture ?

A2) Data flow diagram - It (DFD) illustrates the flow of data within a system, showing how information moves between processes, data stores and external entities. It simplifies system understanding by visualising inputs, outputs and data transformation aiding in system analysis and design.

Levels in DFD-

① Level-0 DFD- Presents overall view of system, showing external entities, processes and data flow.

② Level-1 DFD- Breaks down processes from Level-0. in more detailed process.

③ Level-2 DFD- Further decomposes Level-1 processes into finer grand subprocesses, providing greater details and clarity.

eg → Online Shopping System -

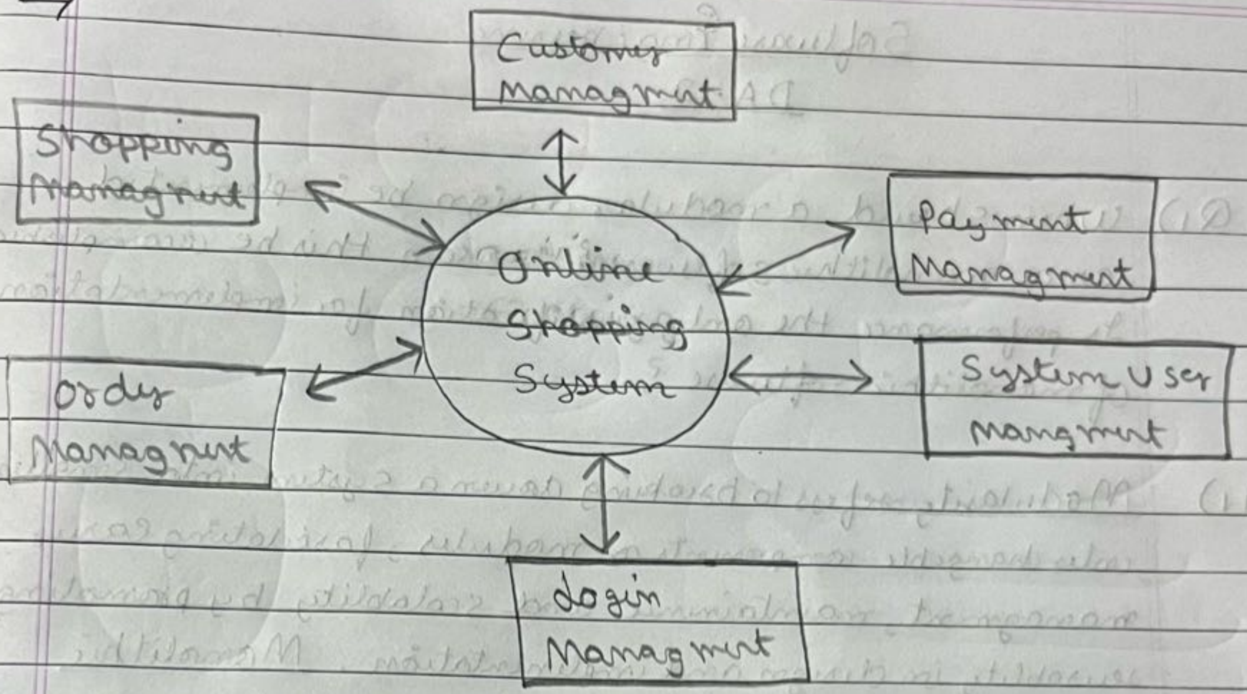
① Level-0 DFD

High level entities and process flow -

① Manging all the Shopping.
② Manging all the shopping cart.
③ Managing all the order.
④ Managing all the payments
⑤ Managing all products
⑥ Managing all delivery.
⑦ Managing all confirm order.

21BKT0021
Shivank Pandey

→



[Zero-level-DFD- online Shopping System]

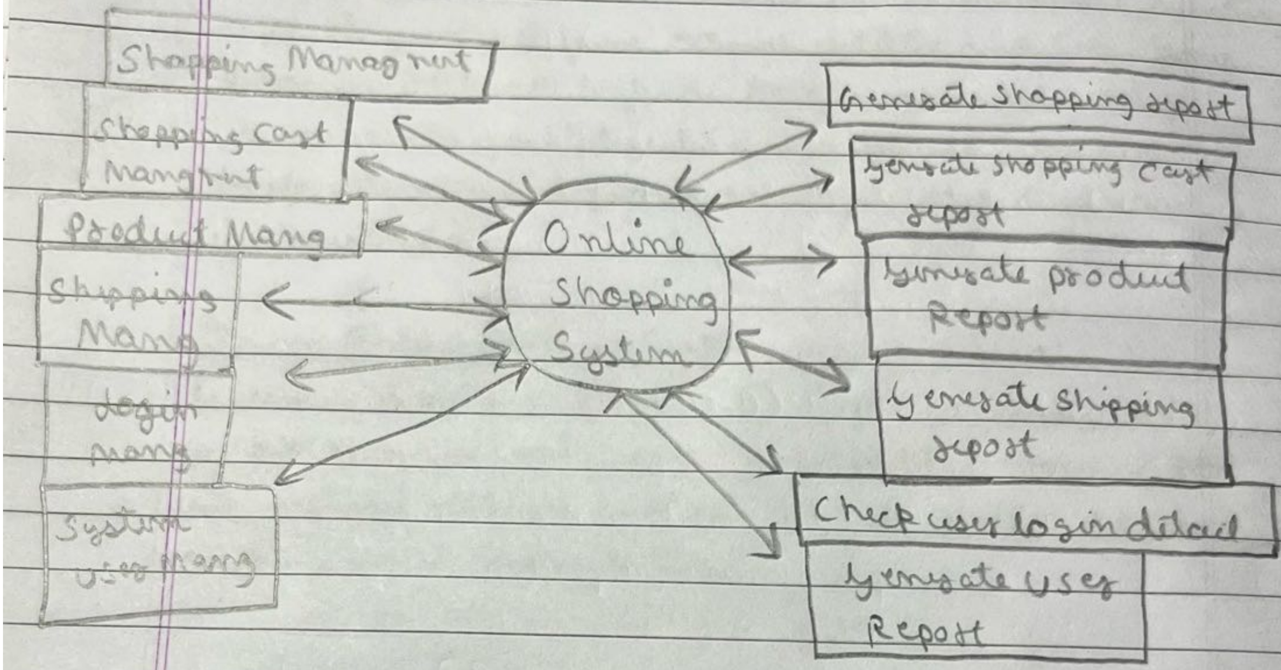
② Level-1 DFD-

Main Entities and output of (1ST Level DFD) -

① Processing Shopping records and generate report of all Shopping
② Processing Shopping Cart Records and generate report of all Shopping Cart
③ Processing Shopping Order Records and generate report of all orders.
④ Processing Payments and records and generate report of all ~~Products~~ Payments.
⑤ Processing Products and records and generate report of all Products.
⑥ Product Delivery Records and generate report of all delivery.
⑦ Processing all conform orders and report all conform orders.

Shivank Pandey
21BKT0021

→



[First -level- DFD -Online shopping System].

③ Level-2 DFD -

Low level functionality -

① Admin login to manage all functionality.

② Admin can add, delete, update and view records of shopping, Order, Product.

③ Admin can manage all details of Shopping cart, Payment, Delivery.

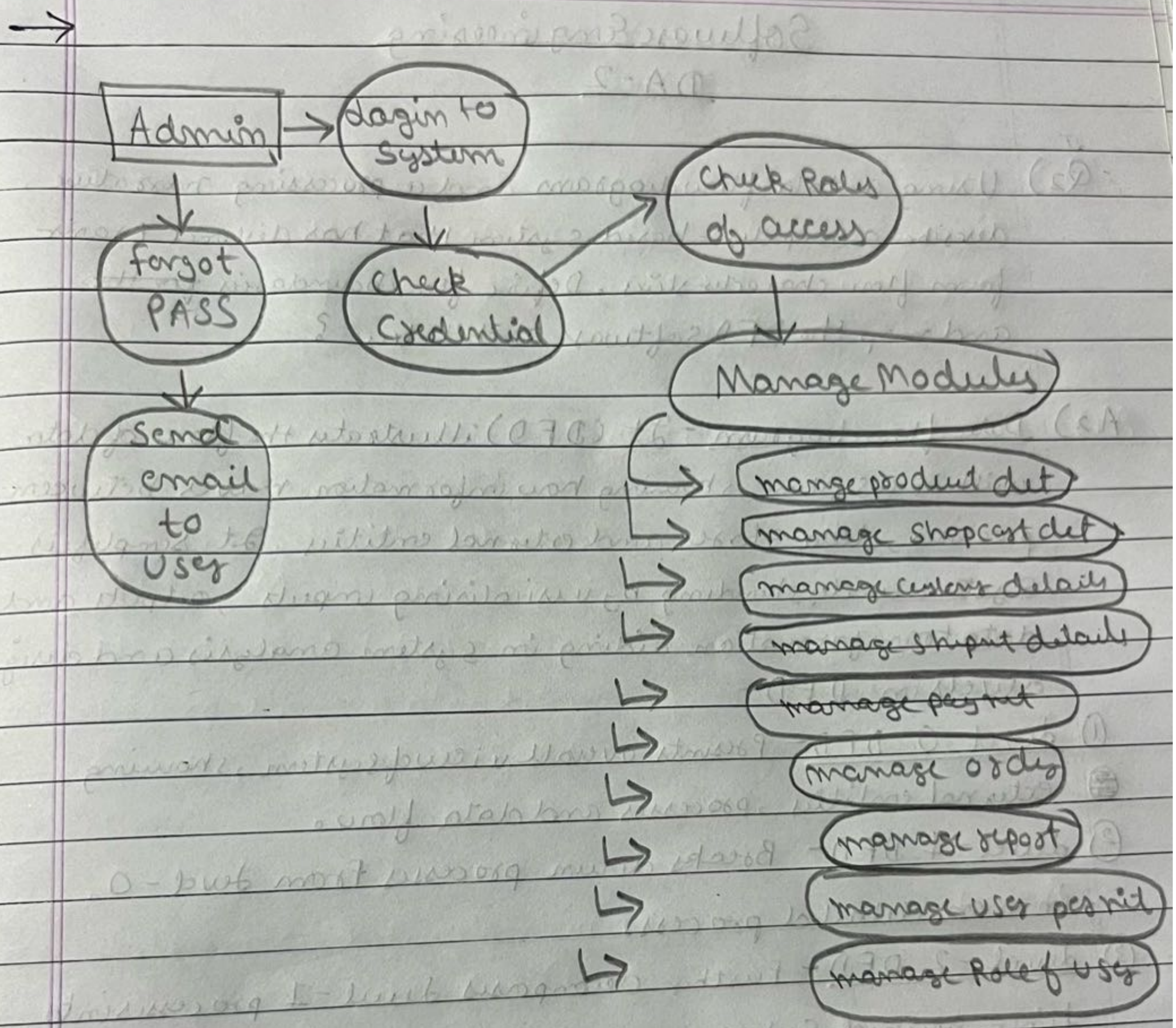
④ Admin can also generate reports of shopping, Shopping cart, Order, Payment, product, Delivery.

⑤ Admin can search for details of Shopping cart, Payment, Product

⑥ Admin can apply different filters across their search.

⑦ Admin can track the detailed info of Shopping cart, order, payment, Product.

Shivank Pandey
21BKT0021.



[Second - Level - DFD - Online Shopping System]

Shivank Pandey
21BKT0021

Date
Page

→ Flow boundaries in DFD tells system boundaries, seperating internal processes from external entities and data flows. Mapping DFD into Software Arch involves translating its components and interactions into software, classes and interface to ensure alignment between system design and implementation.

In an Online Shopping System defines interaction between customers, admin and the system processes ensuring clarity in ~~DF~~ data flow and system functionality. Mapping DFD into software arch involves components like product database, customer carts, order processing.