

Name # Kamran

ID # 15121

BS # (SE)

Semester # 4<sup>th</sup> / Spring 2020 .

Section # "B"

Subject # (Software Engineering)

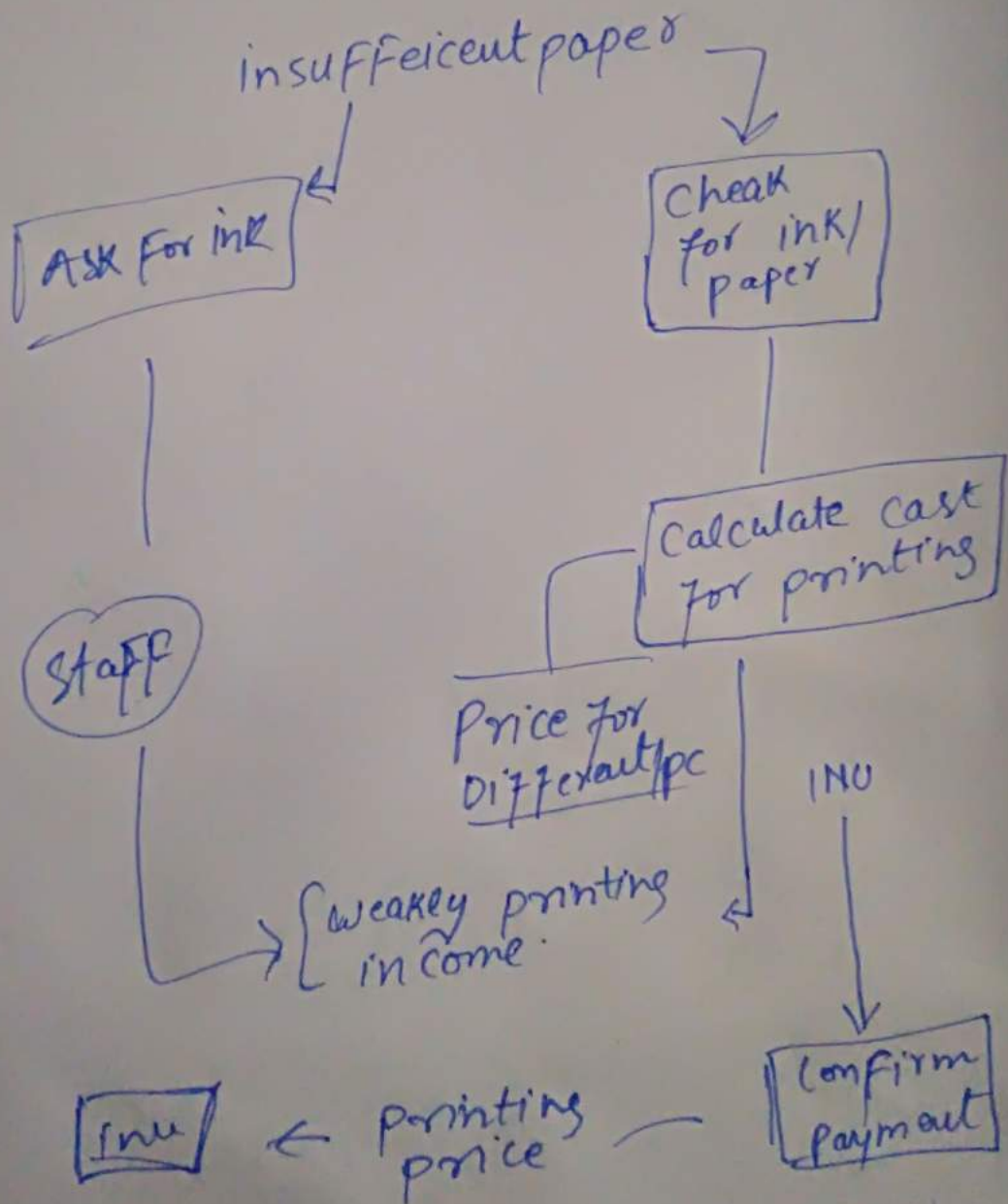
Final term ✓

Instructor # Engr. Ghassan Husnain .

Date # 22/06/20

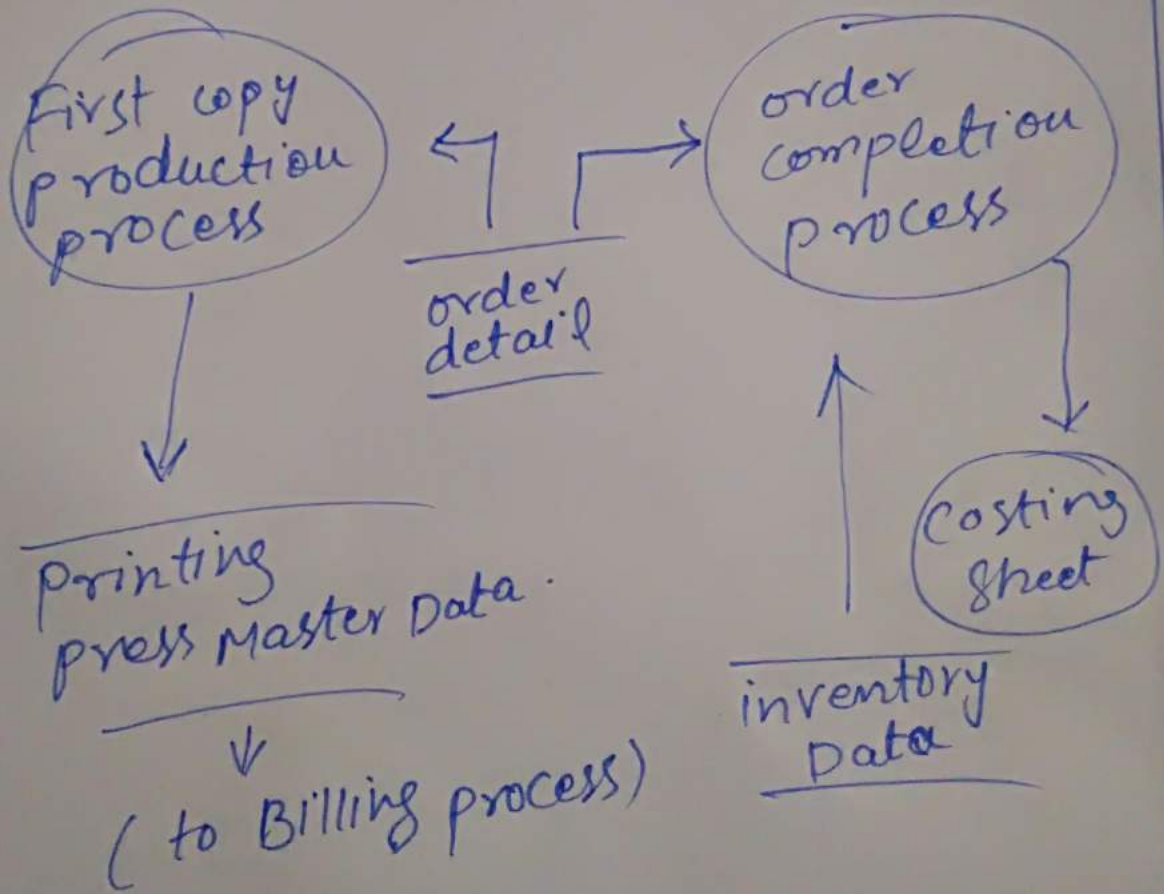
Q No: 1.1 :: Draw a context diagram for INU printing press?

Ans:



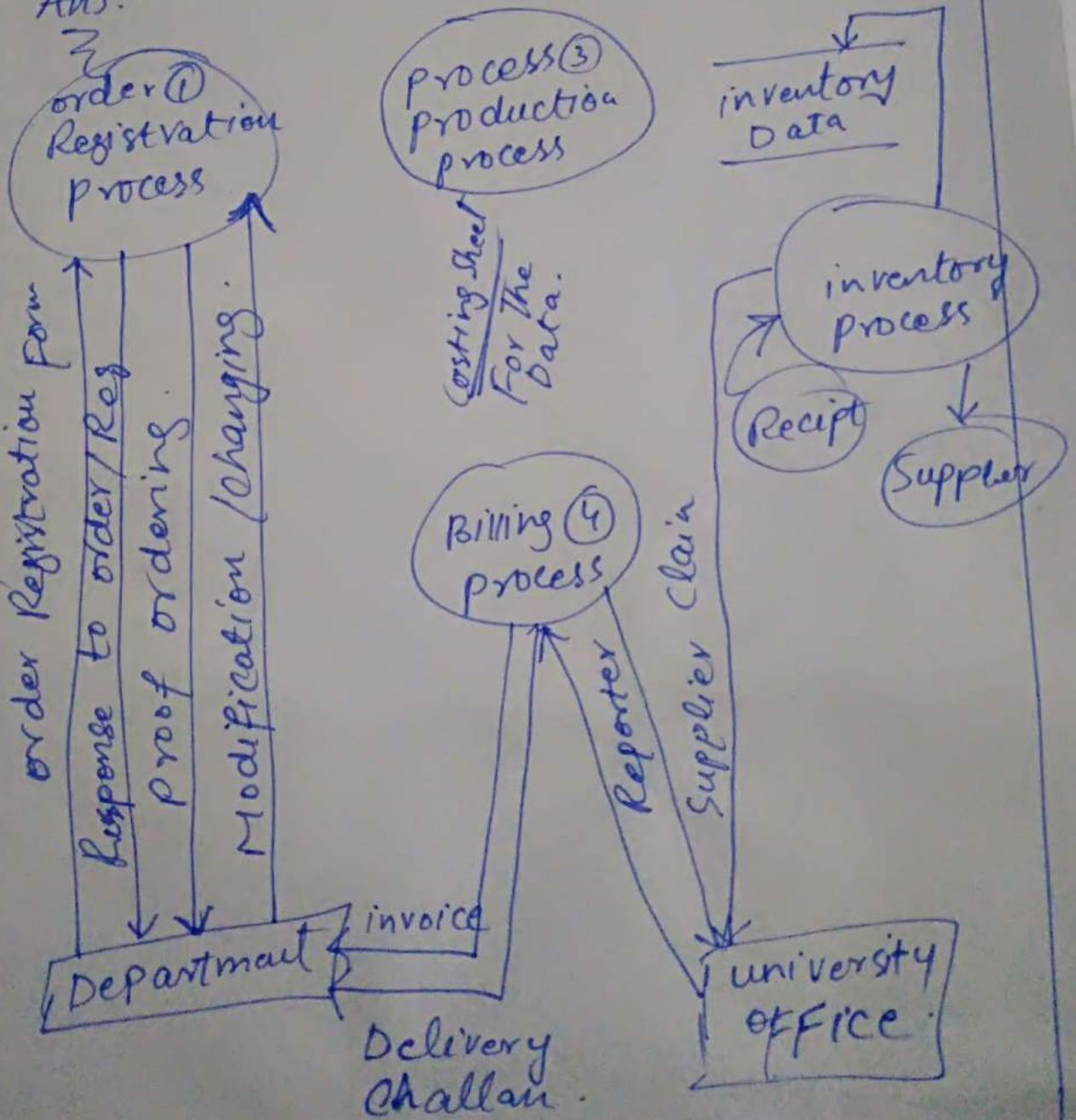
Q. 1.2: Draw a level 1 Data Flow Diagram (DFD) for the above case study?

Ans:



Q.1.3: Draw a level 2 DFD  
 For the order Registration  
 process, press production  
 process, inventory process,  
 and Billing process?

Ans:~



Question No 2 ::

Q.2.1 :: Explain why testing can only detect the presence of errors, not their absence?

Ans: the goal of software testing is to observe the software behavior to meet its requirement expectation. testing is a set of activities where testers try to make the software behave anomalously in order to detect a defect or anomaly to be later fixed.

Q. 2.2:- Define the Following terms:-

Ans:- (1) unit testing:- in software

Development, unit testing is a method by which individual unit of source code - set of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use -

(2) System Testing:- System testing is testing conducted on a complete integrated components that have passed integration testing -

(3) Black Box testing:- Black box testing refers to a software testing method where the SUT (Software Under Test) functionality is tested without worrying about its detail of implementation, internal path knowledge and internal code structure of software.

(4) white box testing: white-box testing is a method of software testing that tests internal structure or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.

Question No 3:

Q.3.1: Briefly describe the three main types of software maintenance - why is it sometime difficult to distinguish between them?

Ans: The main types of software maintenance are as follows:-

(1) Fault Repairs:-

Usually relatively cheap coding errors are errors are more expensive to correct, design may involve rewriting several program components - Requirements error are the most expensive to repair because of the expensive system redesign which be necessary.

(2) Environmental Adaptation:-

This type of maintenance is required when some aspect of the system's environment such as the hardware, the platform operating system, or other support changes the application system must be modified to adapt it to cope with these environmental changes.



(3) Functionality Addition:

This type of maintenance is necessary when the system Requirement change ~~Require~~ in Response to ~~the~~ organizational or Business change. The scale of the changes Required to the Software is often much greater than for the other types of maintenance -

⇒ why is it sometimes difficult to distinguish between them?

In practice, there is not a clear-cut distinction between these type of Maintenance, when the system Adapt to new Environment, then Add functionality to take Advantages of few Environmental Features. Softwares Faults are often exposed because users use the system in unanticipated ways - these type of maintenance are Recognized but a different person sometimes gives them different names -

- (i) Corrective Maintenance -
- (ii) Adaptive Maintenance -
- (iii) Perfective Maintenance -

Q.3.2: what are the principle  
Factors that Affect the  
Costs of system Re-engineering?  
Also briefly Explain the  
Reengineering process with the help  
of diagram?

Ans:- Software Re-Engineering is the  
Examination and Alteration of  
a system to Reconstitute it in  
A new form - the principle of  
Re-engineering when Applied to the  
Software development process called  
Software Reengineering - it effects  
positively at software cost, quality,  
~~improving to software~~  
service to the customer and speed  
of delivery - in Software Reengineering,  
we are improving the software  
to make it more Efficient and  
Effective -

⇒ Re-engineering Cost Factors ::

- \* The Quality of the Software  
to be Re-engineered.
- \* The tools support Quality  
Availability for engineering.

\* Extent of the data conversion which is Required.

\* The Availability of expert Staff for Re-engineering.

⇒ Re-engineering process Diagram ::  
≡ ≡ ≡

