3) Risk response:

Appropriate steps taken or procedures implemented upon discovery of an unacceptably high degree of exposure to one or more risks. Also called risk treatment.

4) Risk monitoring and control:

Risk management is a process of organizing and planning - just as important as strategic, financial, marketing and human resources planning. Like any good planning, the process should be continual or on-going.

Conclusion: In this way by using project management tool we have drawn gantt chart successfully.

Experiment No. 04

Aim: To create structured data flow analysis. (DFD)

Objective:

After completion of this experiment you will be able to:

- Understand how to design architecture of software engineering application.
- Learn and use automated analysis and designing tools for creating software engineering application architecture.
- To identify different behavioral modules and their flow of operations while designing any software applications.
- Ability to generate alarms or trigger for some irregular dymanic operations.

Outcome:

Ability to identify different building blocks and their flow of operations while designing any software applications.

Use of automated analysis and designing tools for creating architecture covering application modules, submodules, sub-sub modules and data flow between them.

- Able to identify and understand dynamic operations happening due to uncertain or trigger inputs and the ability of system to respond to such irregular activities.

Software Used: Design & analysis Tool

Theory:

Definition

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design).

Use a Data Flow Diagram (DFD) to show the relationships among the business processes within an organization to:

- external organizations,
- external systems,
- customers,
- Other business processes.

Types of DFD

Context DFD (LEVEL 0)

The context diagram clearly shows the interfaces between the system under investigation and the external entities with which it communicates. Therefore, whilst it is often

conceptually trivial, a context diagram serves to focus attention on the system boundary and can help in clarifying the precise scope of the analysis.

The context diagram shown below represents online Airline Reservation System. The OBKS receives personal details of passengers, User_id and details such User_name, Email id,Phone_no,city etc. as inputs. The output of OBKS is confirmation of Tickets and user interface. Level 0 consist of only one main process that is OBKS.

Level 1 Diagrams

A diagram 1 is a diagram showing the system itself.

In OARS, main process OARS consist of 6 major functions as follows:

2. Enquiry
3. Login
4. Provide Details
5. Reservation
6. Admin Function
7. Logout

Level 2 Diagrams

Level 2 DFD are explosions of higher level presentations and are detailed pictures of major processes or subsystems.

For example in OARS software DFD, we had given the detain description about major processes Provide Details, Select Provision, Admin Function as follows:

- Provide Details:

  Day, Date, Cost, Airline Name
  Source, Departure_time
  Destination, Arrival_time
  Available_seats:

- Select Provision:

- Cancellation:

  Provide personal details
  Provide flight details
  Refund
  Cancellation_ack:

- Reservation:

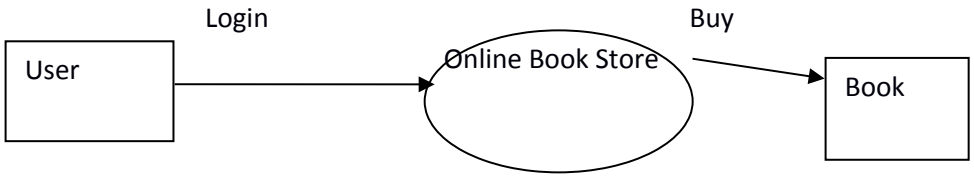
Provide personal details
Provide flight details
Payment

- Maintain DB:

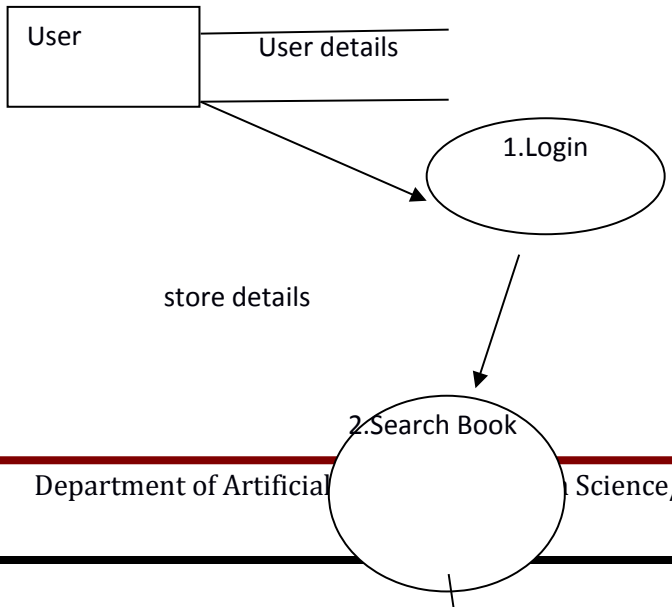
i. Payment_ack
ii. Insert
iii. update
iv. delete

Output:

Level 0:



Level 1:



Authenticated

Book found

credit card no

Payment Details

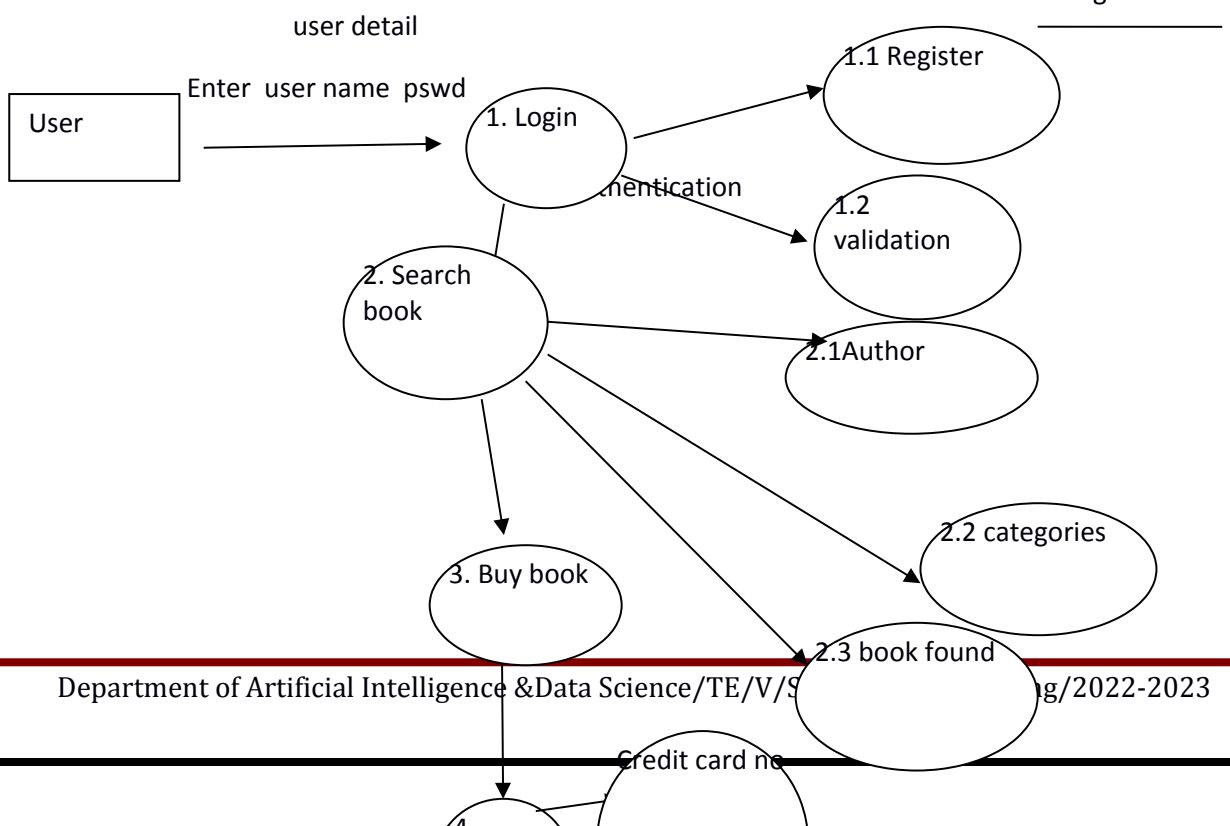
end

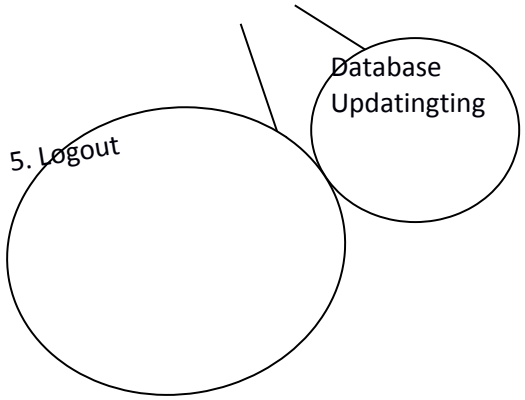
6. Logout

Level 2:

not registered

user detail





Conclusion: Hence using DFD for OARS we have studied the flow of data through the system and visualization of data processing as one of the way to create detailed architectural flow for selected Case study.

.

Experiment No:05

Aim: Use of metrics to estimate the cost**.**

Theory:

1) COCOMO Model:

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current project characteristics.

COCOMO was first published in Boehm's 1981 book Software Engineering Economics as a model for estimating effort, cost, and schedule for software projects. It drew on a study of 63 projects at TRW Aerospace where Boehm was Director of Software Research and Technology. The study examined projects ranging in size from 2,000 to 100,000 lines of code, and programming languages ranging from assembly to PL/I. These projects were based on the waterfall model of software development which was the prevalent software development process in 1981.

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level,