

System Development Methods

CT00046-3-2

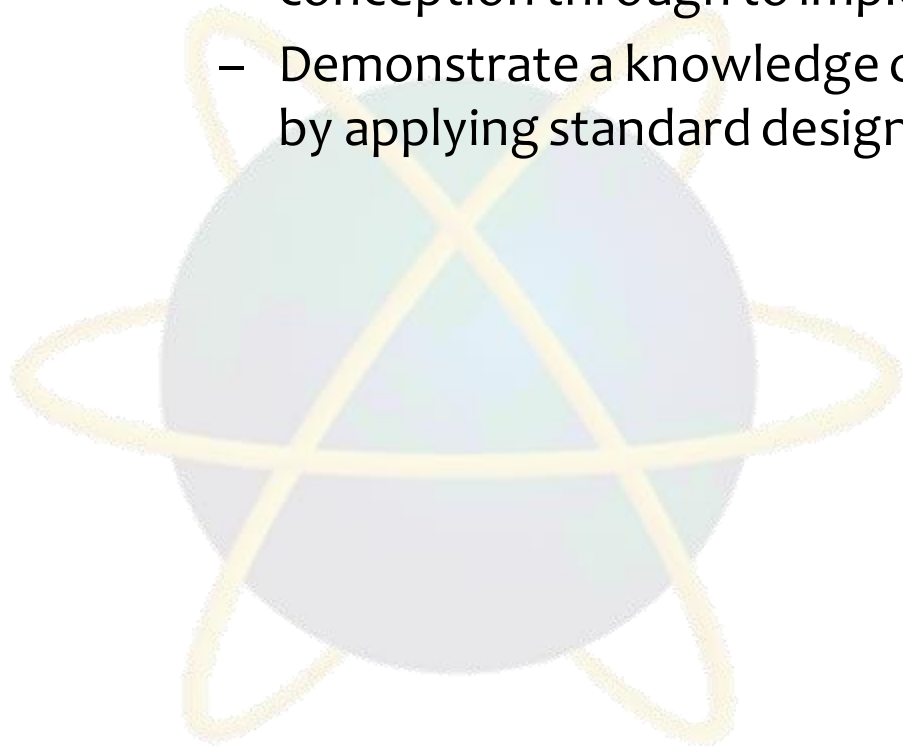


A . P . U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

System Design – Part 1

Learning Outcome

- At the end of the module, you should able to:
 - Apply UML analysis, design and implementation techniques to develop a simple prototype, with a suitable interface, from conception through to implementation.
 - Demonstrate a knowledge of the fundamental issues of prototyping by applying standard design principles.



Key Terms you must be able to use

- If you have mastered this topic, you should be able to use the following terms correctly in your assignments and exams:
 - Design Concepts
 - Design Techniques
 - Design Tools



System Design

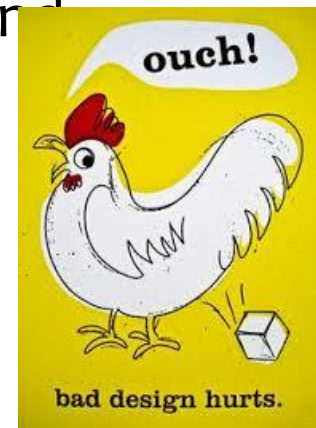
- Systems Design
 - Process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements.
 - Aka; System Modeling
- System Design in a typical software project;
 - **Input – System Specification**
 - Including SRS
 - **Output – System Design Specification.**
 - Has details about what to design
 - Created by designers for programmers.

System Design Areas

- Hi-level Design
 - System Architecture
 - Hardware, Software, Network, etc
- Software Design
 - Software Architecture
 - Conceptual Design
 - Logical Design
 - Physical Design

System Design Objectives

- To creating a solution to a problem in hand
- To breakdown and understand complex concepts and process.
- To create a product safe and secure for use (HCI)
- To add vital details to product
- To come-up with efficient and effective product / solution
- To ensure product comply with set specifications and standards.
- To implement modern technology into product
- To test product early and avoid costly rework



System Architecture

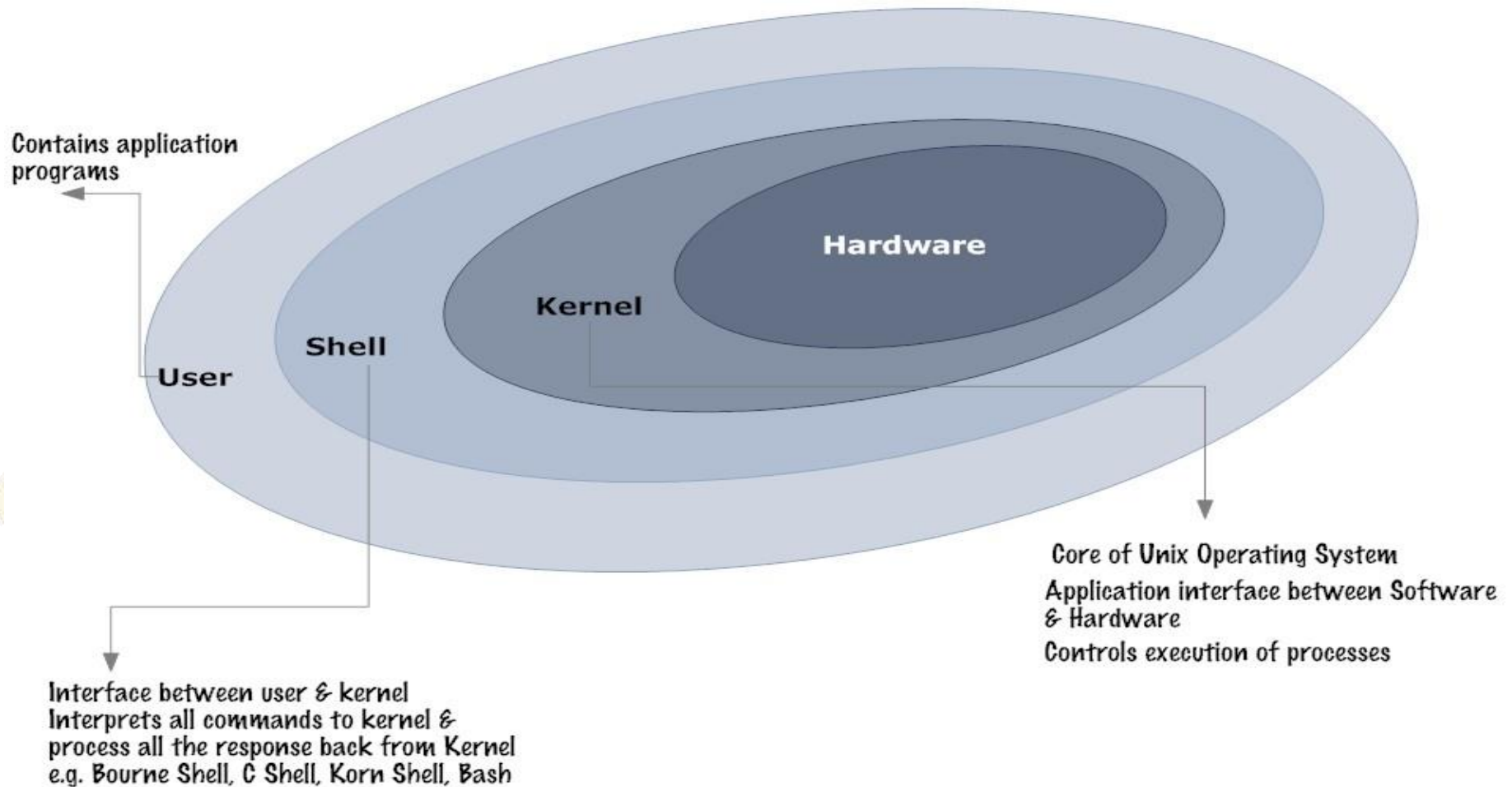
- A high-level view of the system where developer need to determine;
 - The types of users
 - Environment (Different ‘views’ of the system)
 - Behavior of the system
 - Availability of technology
 - The platform
 - Laws and Legislations
 - Industry Standards and Protocols
 - Security
 - Network and connectivity
 - Maintenance

System Architecture

20-05-17

example

Unix Architecture Layers



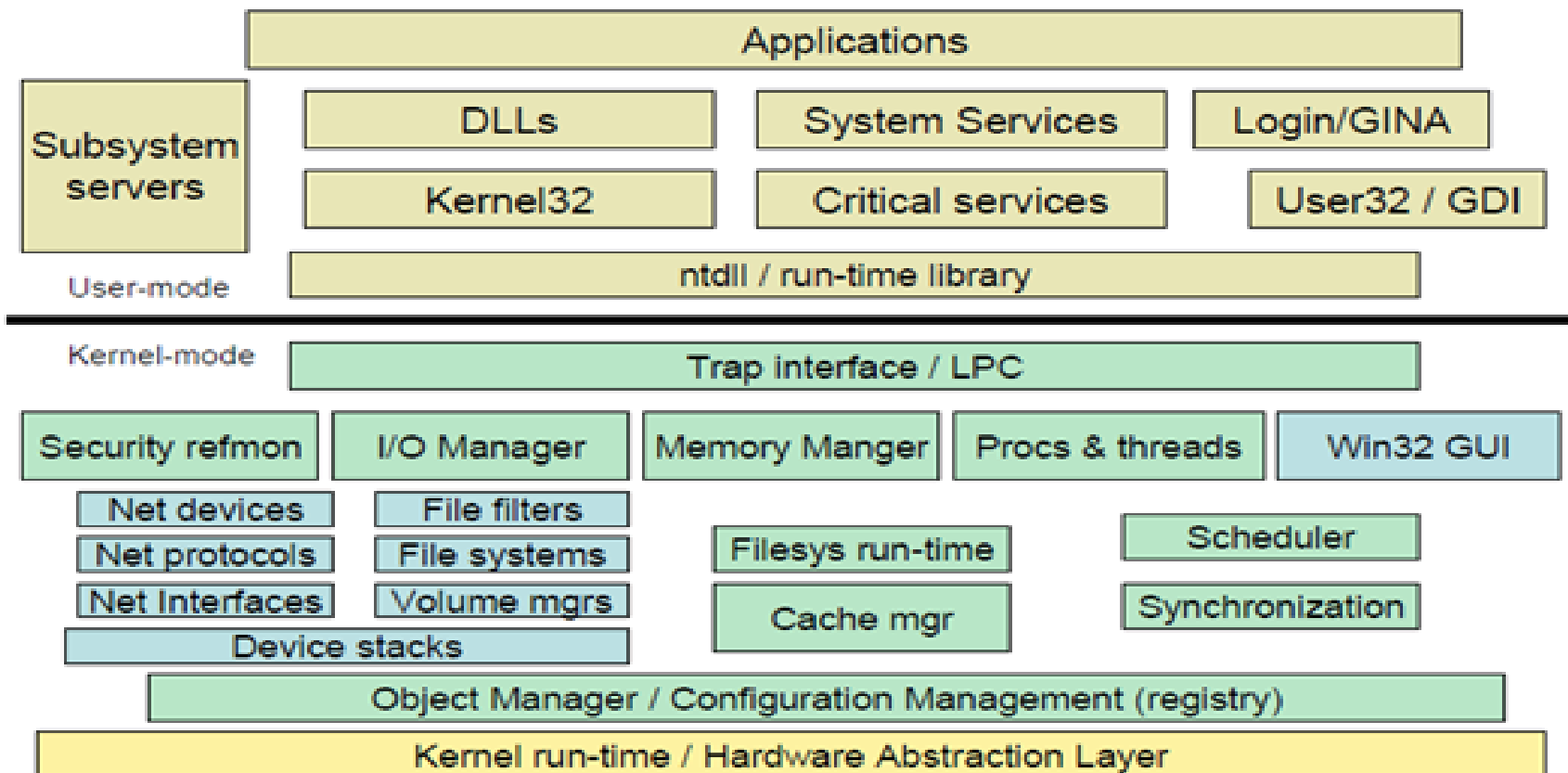
Software and Hardware Architecture

- Subcomponent of System Architectures
- Developers need to be concerned of;
 - Operating system and versions
 - Input / output requirements
 - Storage requirements
 - Processing requirements
 - Interfaces (including GUI)
 - Security, System control and backup or recovery
 - System support

Software Architecture

example

Windows Architecture



v3

© Microsoft Corporation 2006

Conceptual Design

- Understanding and presenting the concept of the product
 - Usually to non-IT people, users, customers, etc
- Contains basic information, major process and components only.
- Example (in IT);
 - Context Diagram
 - Use Case Diagram
 - Sequence Diagram
 - Flow Chart
 - Story Board
 - Rich Picture
 - Posters, etc.



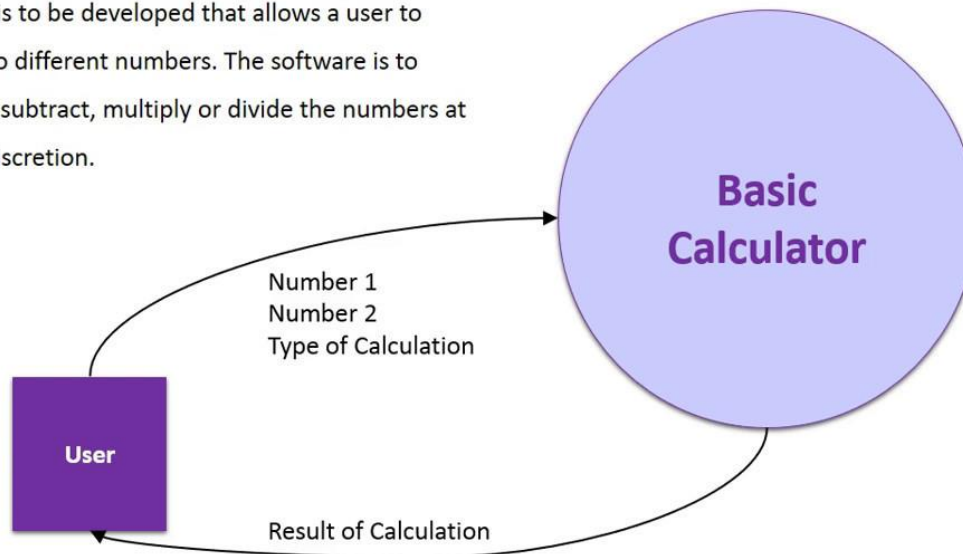
Context Diagram

- show the interactions between a system and other actors (external factors) with which the system is designed to interface.

Context Diagrams

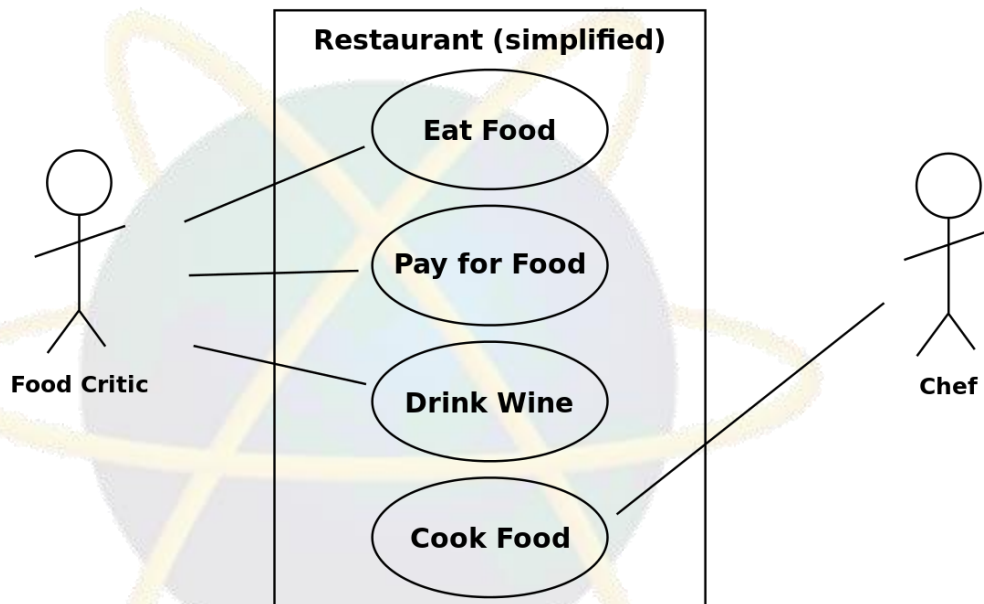
Example: Basic Calculator

A program is to be developed that allows a user to enter in two different numbers. The software is to either add, subtract, multiply or divide the numbers at the users discretion.



Use Case Diagram

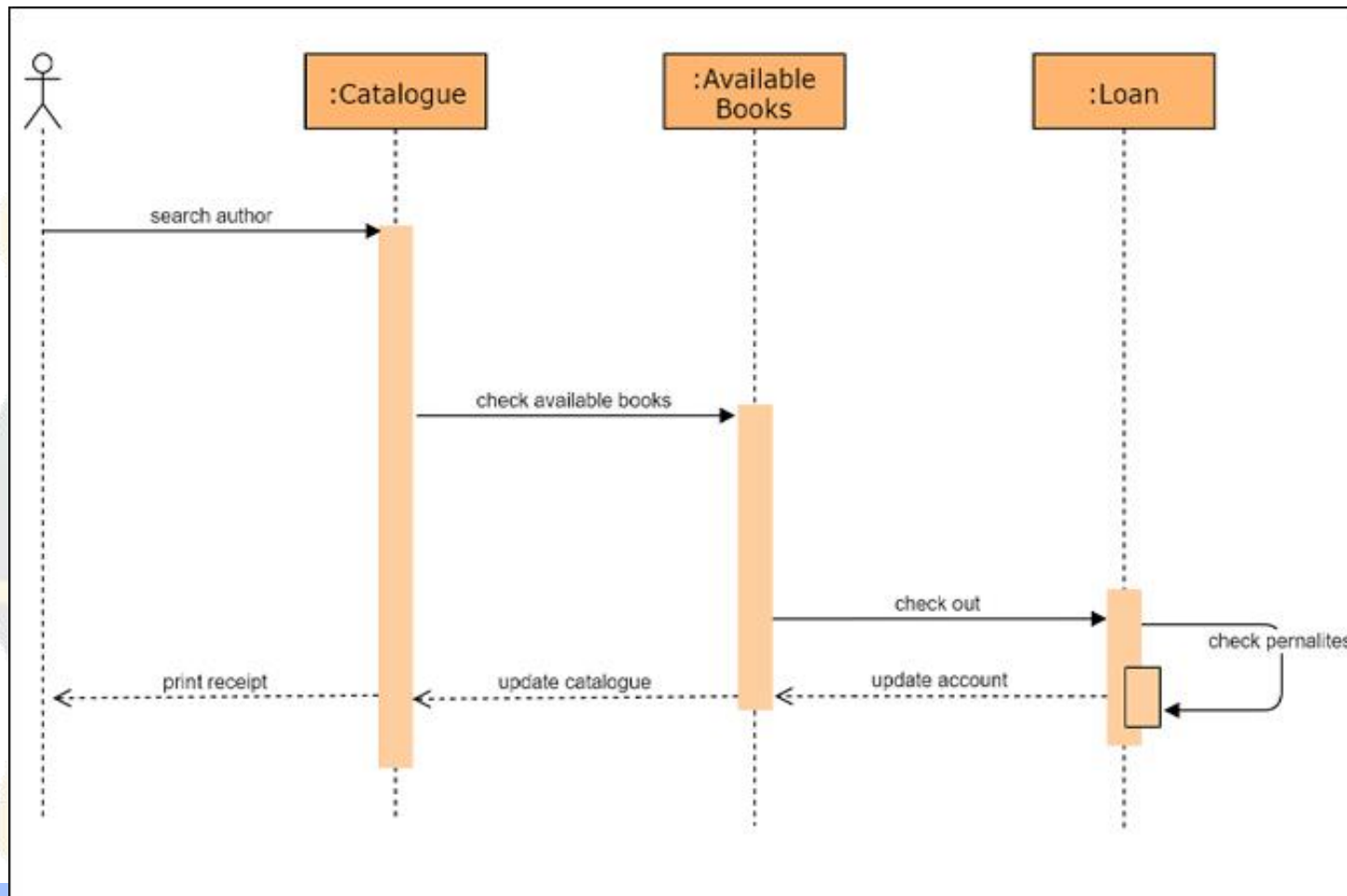
- used to gather the requirements of a system including internal and external influences



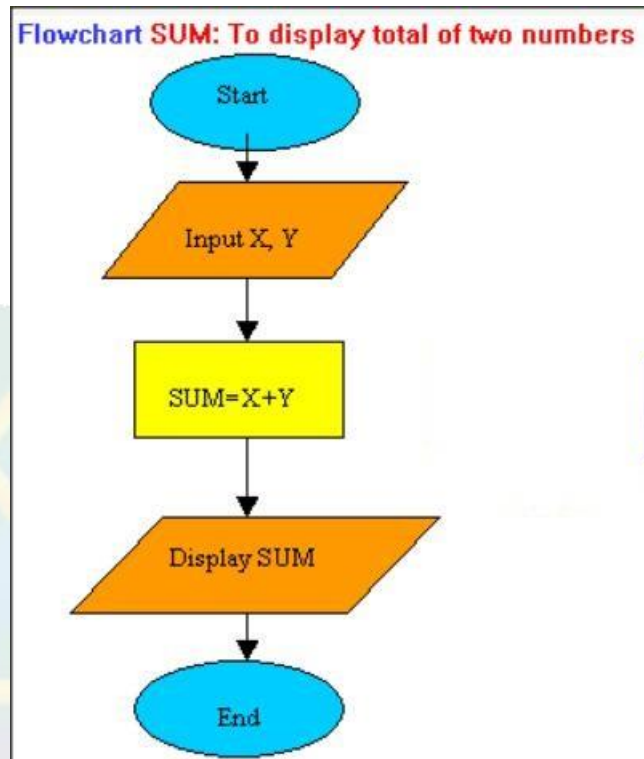


Sequence Diagram

- describes how—and in what order—a group of objects works together.



Flowchart

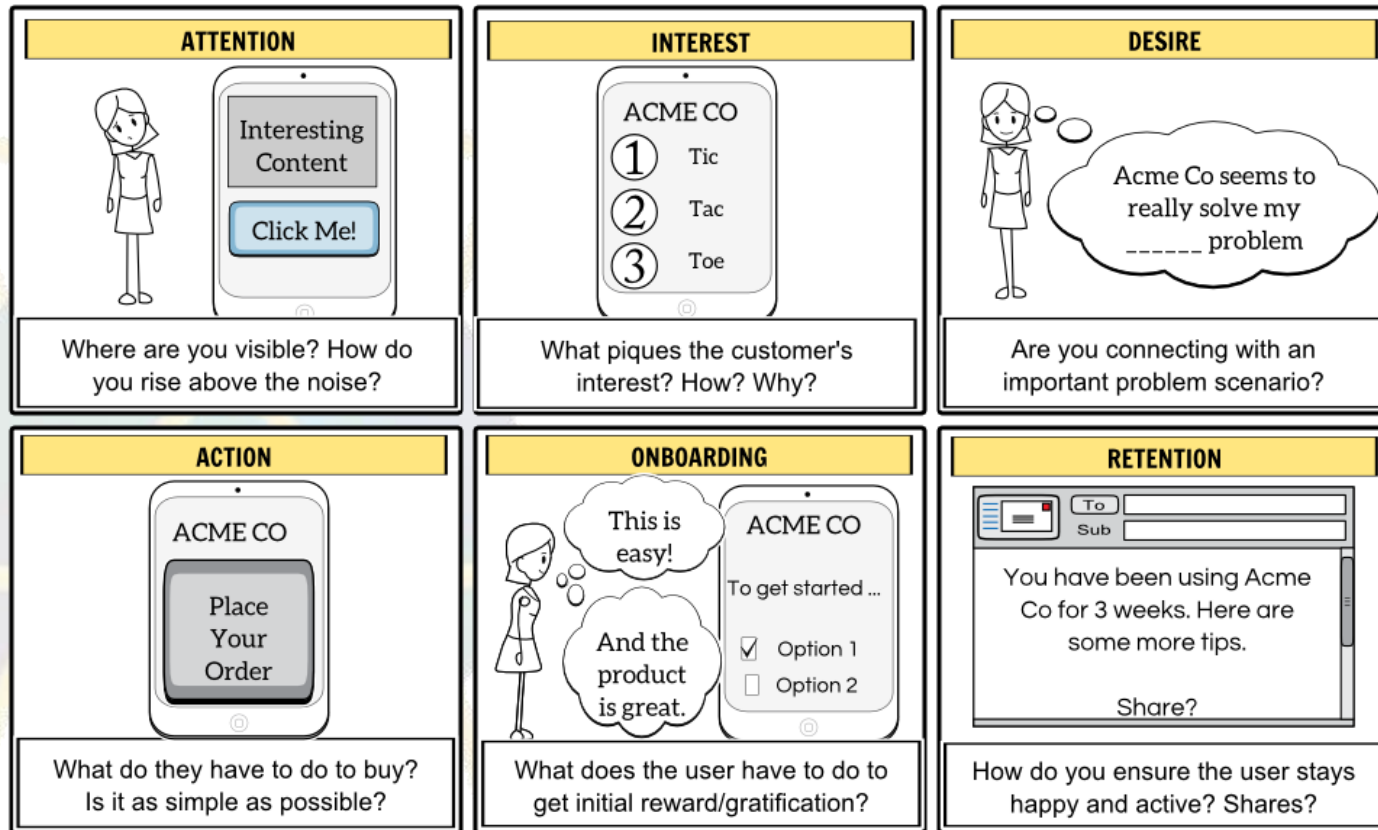


www.EasyCodeBook.com
For Perfect Programming Tutorials



Storyboard

- a graphic layout that sequences illustrations and images with the purpose of visually telling a story.



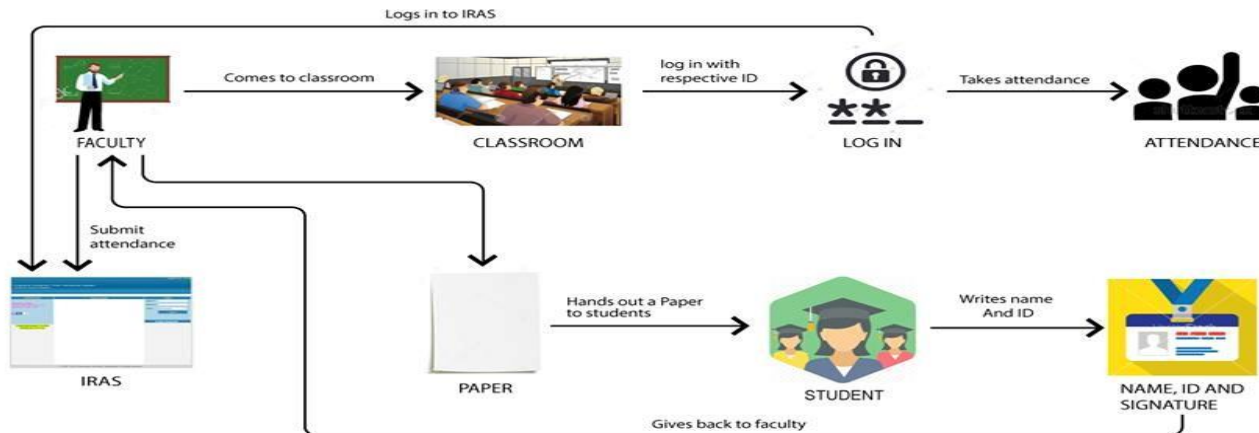
Create your own at StoryboardThat.com



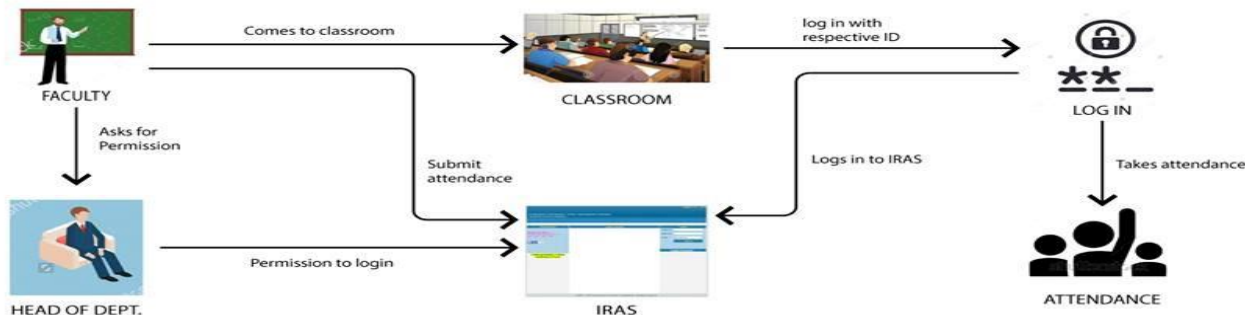
Rich Picture

- way to explore, acknowledge and define a situation and express it through diagrams to create a preliminary mental model.

ONLINE ATTENDANCE AND MANUAL ATTENDANCE



ATTENDANCE OVER RIDE




Posters

- A **poster** is a temporary promotion of an idea, product, or event put up in a public space for mass consumption

PRODUCT FEATURES

Front End - Main Screen



Top Advertising Banner
Display a custom banner to promote the event or advertising

Smart Search Function
Search based on author, title or description

Custom grid layout
Set the number of columns and rows

Navigational Buttons
Use the buttons or swipe to navigate through the pages

Screensaver
Display the screensaver by displaying promotional material or actual ePosters submissions

Filters
Filter based on the author or the poster groups

Display Single Poster
Touch the poster to display the single poster

Bottom Advertising Banner
Display a custom banner to promote the event or advertising

Logical Design

- Abstract of the system, usually presented in diagrams (models)
 - Process Design
 - Data Design
- Example;
 - Data Flow Diagram
 - Activity Diagram
 - Entity Relations Diagram
 - UML – Use Case Diagram
 - UML – Class Diagrams, etc

Data Flow Diagram

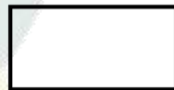
- A data flow diagram (DFD) **maps out the flow of information for any process or system.**
- It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.



Function



File/Database



Input/Output



Flow

Data Flow Diagram

- DFD consists of processes, flows, warehouses, and terminators.
 - **Process:** The process (function, transformation) is part of a system that transforms inputs to outputs. The
 - **Data flow:** Data flow (flow, dataflow) shows the transfer of information (sometimes also material) from one part of the system to another.
 - **Warehouse:** The warehouse (datastore, data store, file, database) is used to store data for later use.
 - **Terminator:** The Terminator is an external entity that communicates with the system and stands outside of the system.
 - It can be, for example, various organizations (eg a bank), groups of people (e.g. customers), authorities (e.g. a tax office) or a department (e.g. a human-resources department) of the same organization, which does not belong to the model system.

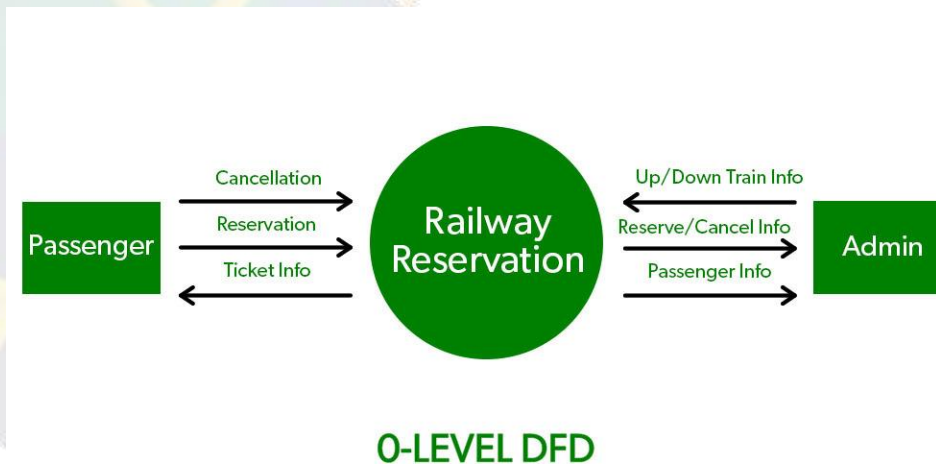
Data Flow Diagram

- DFD Levels

- DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond.

- 0-level DFD

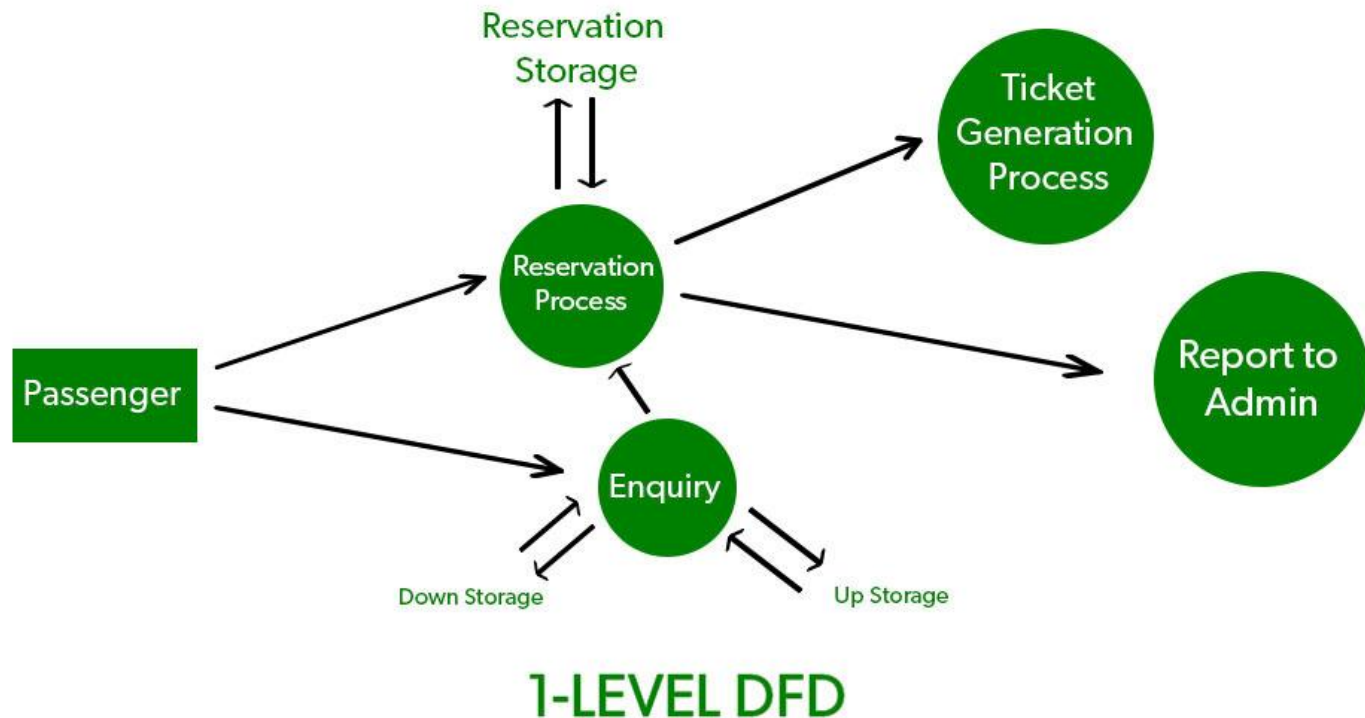
- It is also known as a context diagram.
- It's designed to be an abstraction view, showing the system as a single process
- It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.



Data Flow Diagram

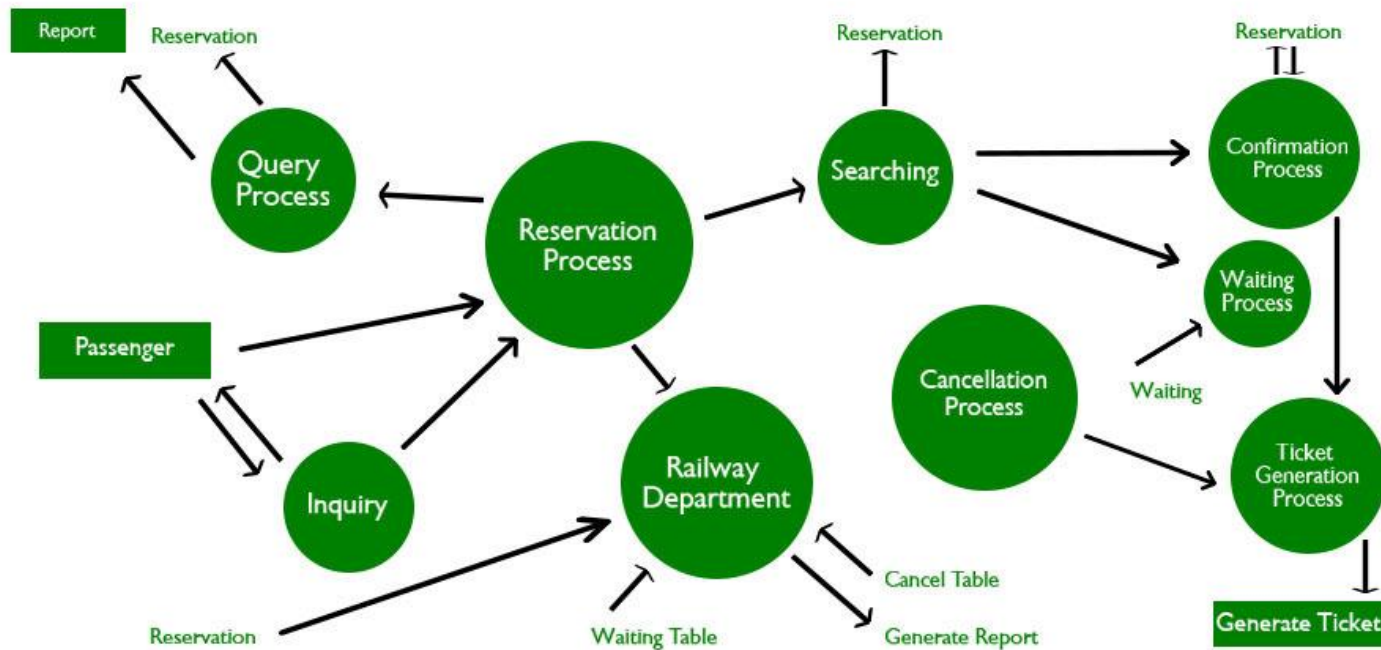
- **1-level DFD:**

- In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes.
- In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.



Data Flow Diagram

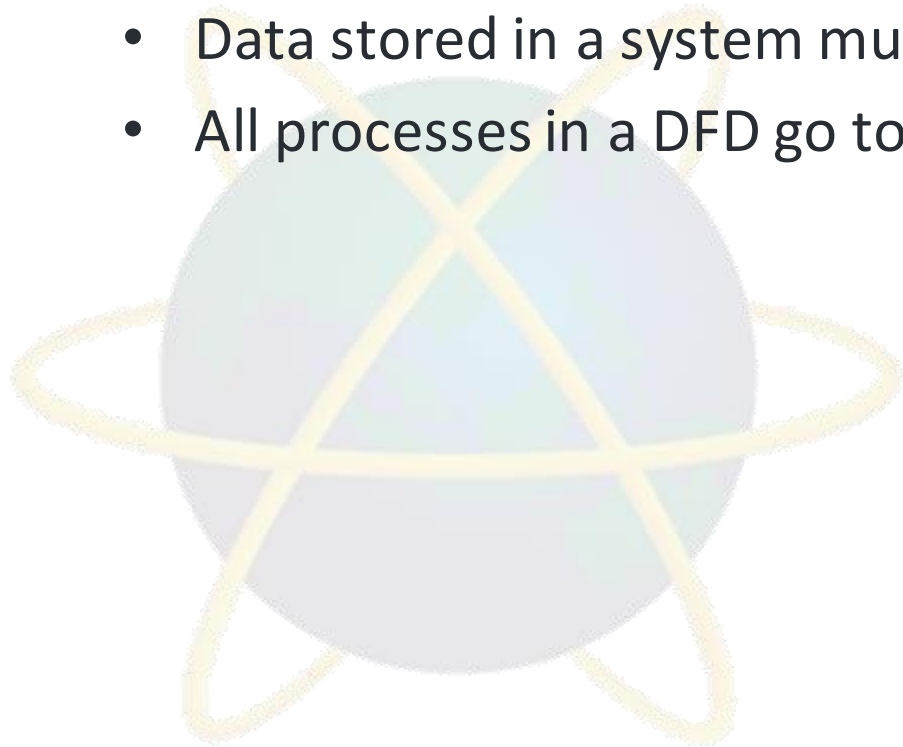
- 2 –Level DFD
 - 2-level DFD goes one step deeper into parts of 1-level DFD.
 - It can be used to plan or record the specific/necessary detail about the system's functioning.



2-LEVEL DFD

Data Flow Diagram

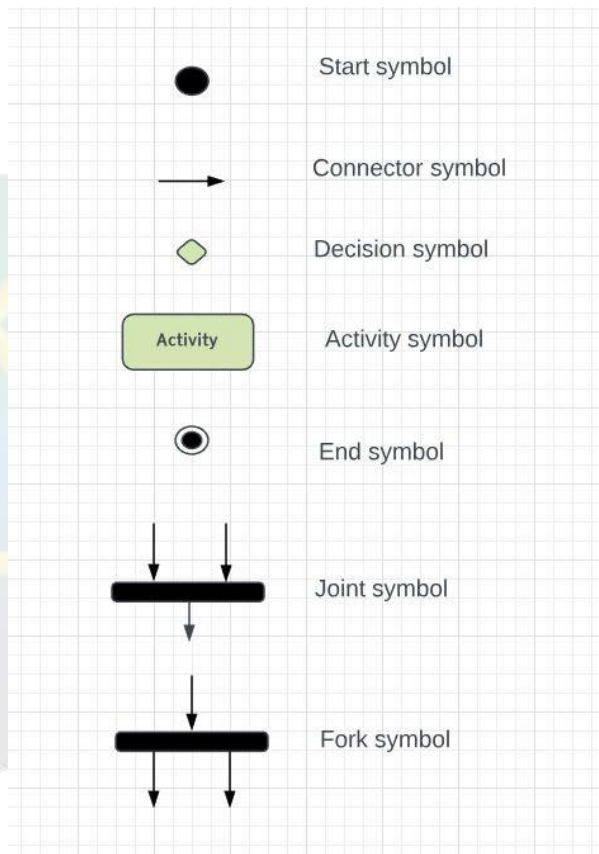
- Each process should have at least one input and an output.
- Each data store should have at least one data flow in and one data flow out.
- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store.





Activity Diagram

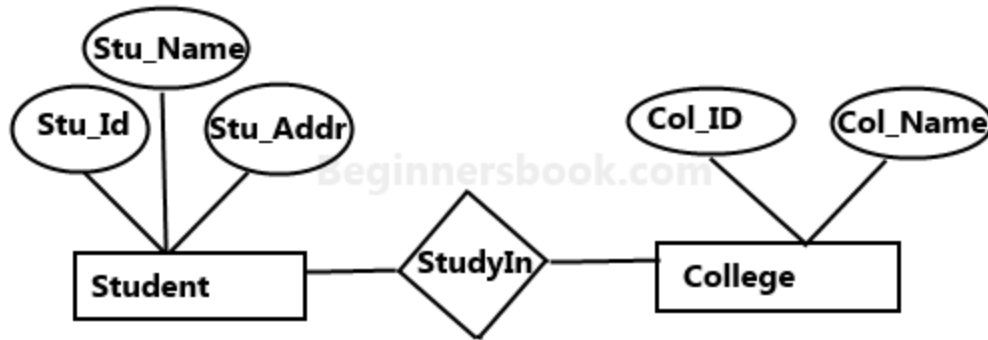
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity.
- The activity can be described as an operation of the system.



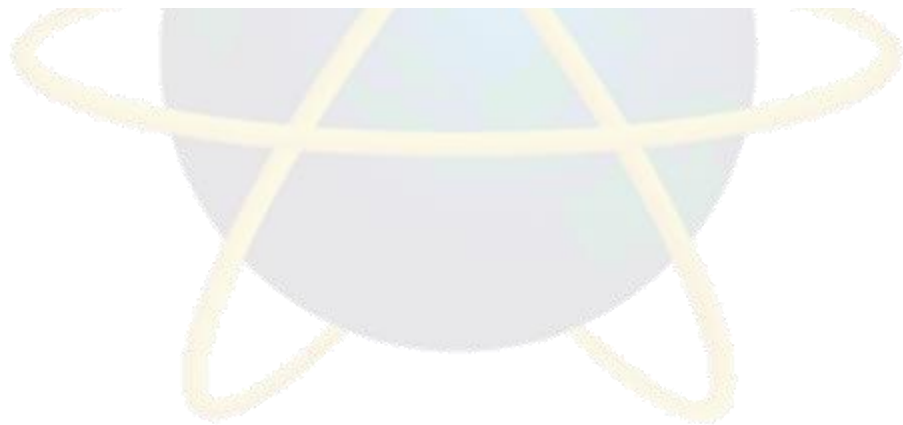
Entity Relationship Diagram

- An Entity Relationship (ER) Diagram illustrates how “entities” such as people, objects or concepts relate to each other within a system.
- ER Diagrams are composed of entities, relationships and attributes.
- **Entity:**
 - A definable thing—such as a person, object, concept or event—that can have data stored about it.
 - Typically shown as a rectangle.
- **Relationship**
 - How entities act upon each other or are associated with each other.
- **Attribute**
 - A property or characteristic of an entity. Often shown as an oval or circle

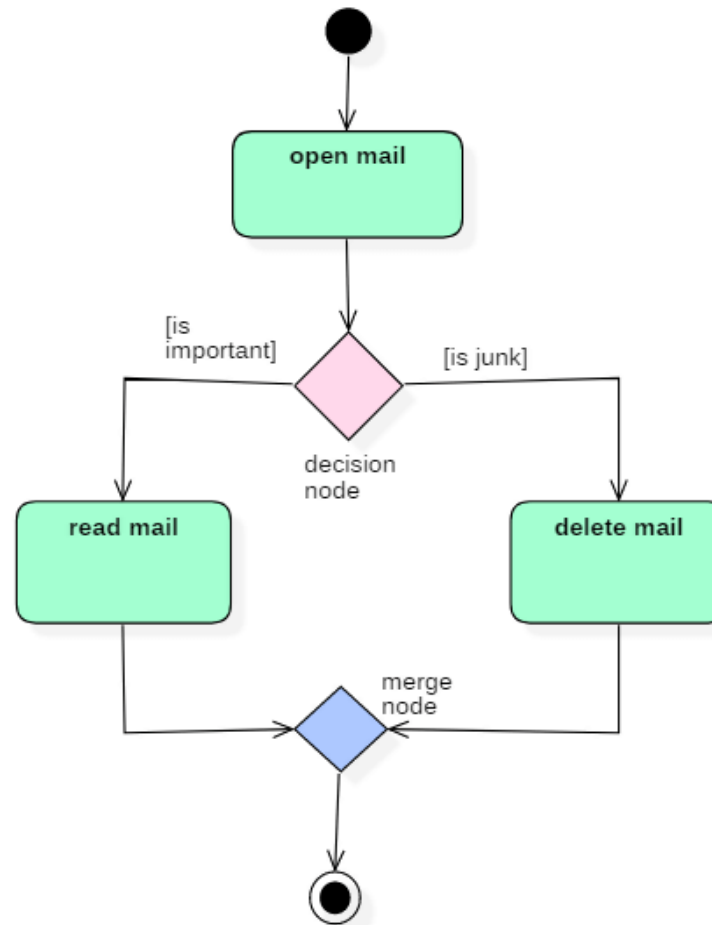
ER Diagram



Sample E-R Diagram

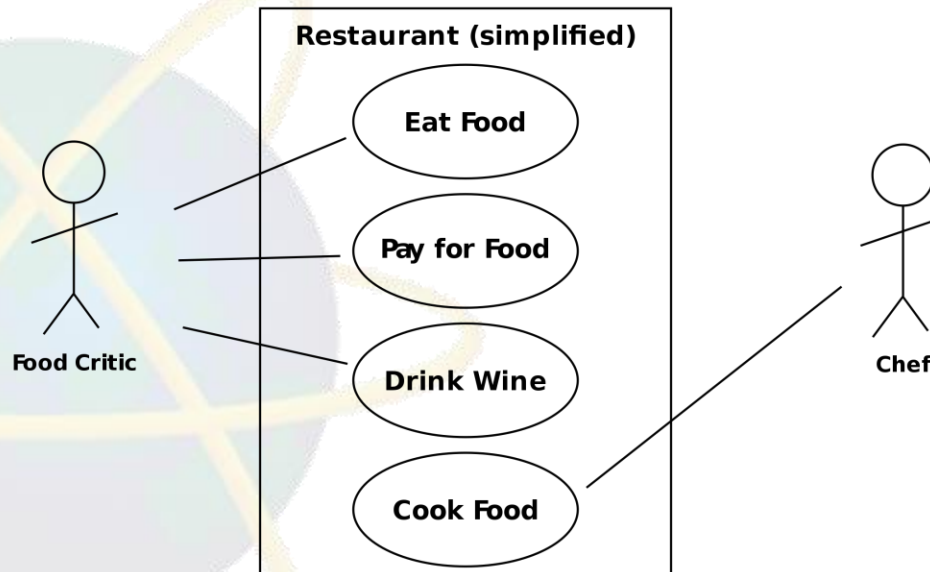


Activity Diagram



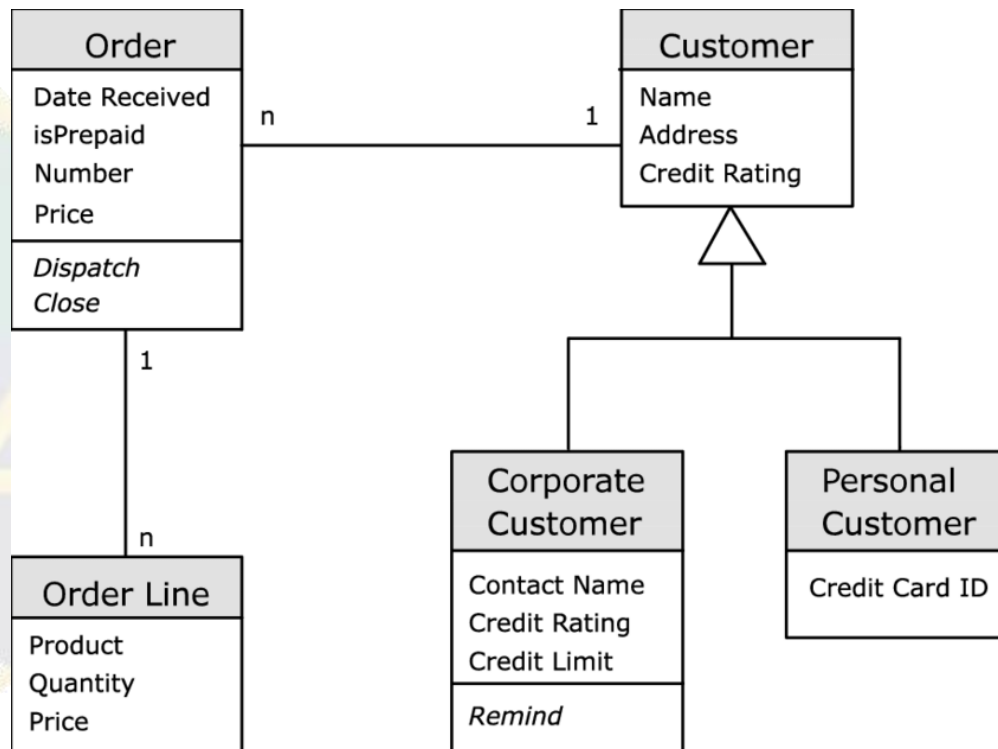
Use-Case Diagram

- In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system.



UML Class Diagram

- to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide.



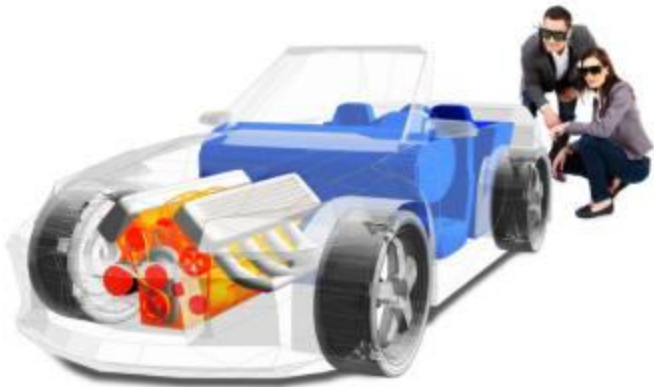
Physical Design

- Enhanced presentation of the design, usually graphical presentation of the product
- Usually presented as;
 - Input-output of the system for users via GUI
 - **Input-process-output** for developers
- Popular type of Physical Design
 - Prototype
 - Simulation
 - Animation
 - Video

Prototype

- Prototype is a working model of software with some limited functionality.
- The prototype does not always hold the exact logic used in the actual software application.
- Prototyping is used to allow the users evaluate developer proposals and try them out before implementation.
- It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

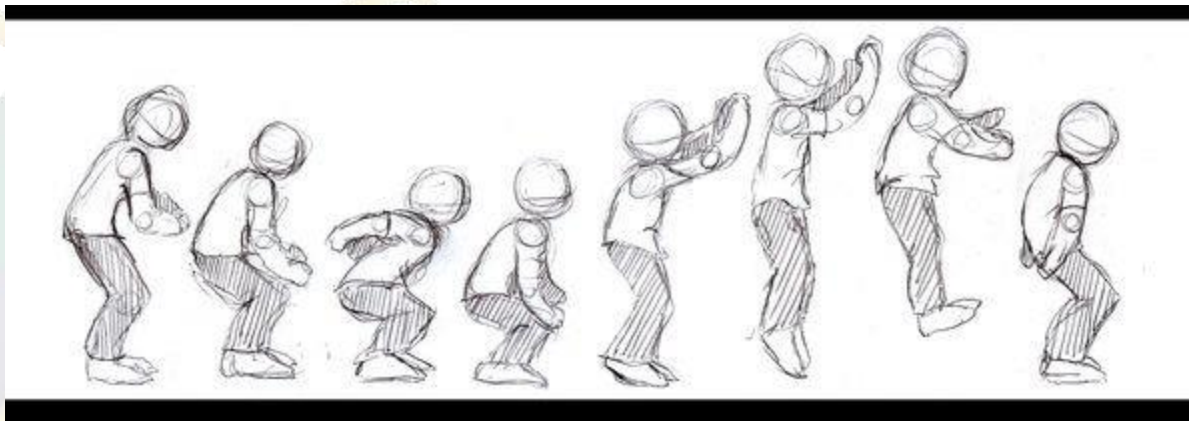
Simulation



- A simulation imitates the operation of real world processes or systems with the use of models.
- The model represents the key behaviours and characteristics of the selected process or system while the simulation represents how the model evolves under different conditions over time.
- Simulations are usually computer-based, using a software-generated model to provide support for the decisions of managers and engineers as well as for training purposes

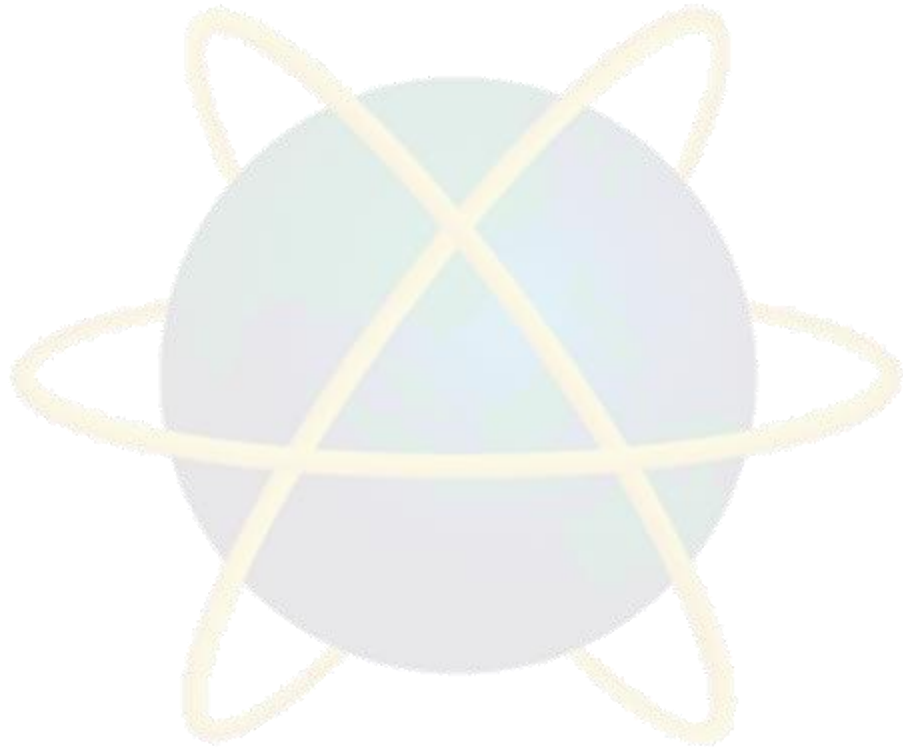
Animation

- Animation is a method in which figures are manipulated to appear as moving images



Video

- **Video** is an electronic medium for the recording, copying, playback, broadcasting, and display of moving visual media.

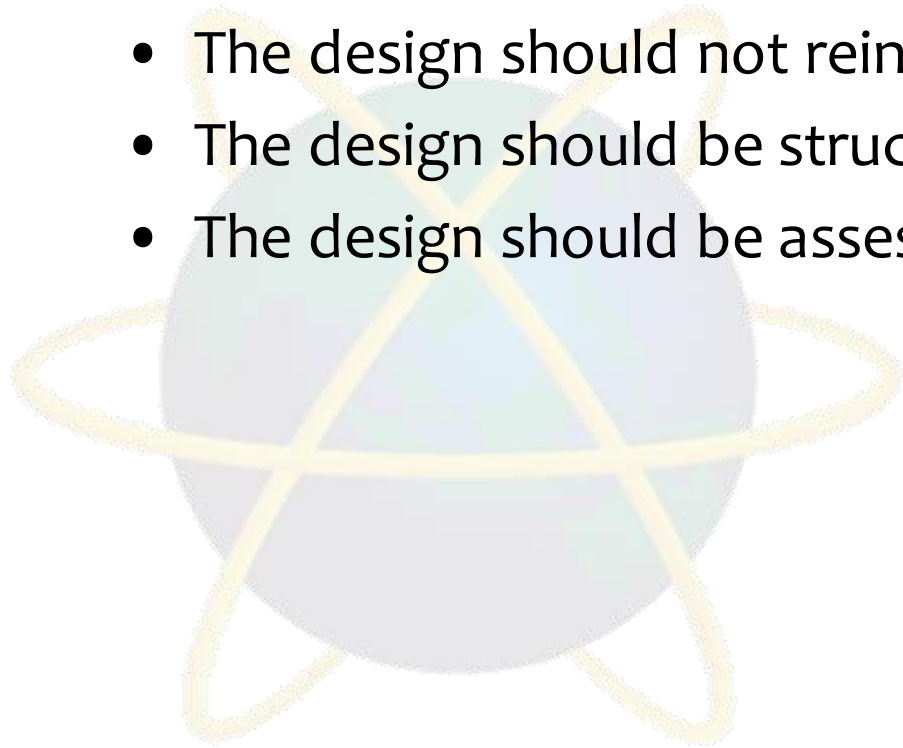


What makes good design

- Creativity and innovation
 - Finding ways/ ideas to improve design
- Good development tools
 - Using latest software to construct, test and present design
- Experience
 - Improvement of skill
- Commitment to high quality
 - Comply with design standards and principles
- Fast changes to demand and environment
 - R&D

Good Design Practices

- The design process should not suffer from "tunnel vision."
- A good designer should consider alternative approaches.
- The design should be traceable to the analysis model.
- The design should not reinvent the wheel.
- The design should be structured to accommodate change.
- The design should be assessed for quality.





Question & Answer

Tutorial

1. Find a (simple) diagram of a popular system's Architecture. Present the diagram in class. Explain the vital components displayed in the diagram.
2. List the types of **design diagrams** that you would choose to present the below new system to users. Explain Why? *(consider types of design such as conceptual, data, process, etc)*
 - a) A new e-commerce website for a local book store.
 - b) A calculator application for smartphone.
 - c) A traffic light control system.
 - d) A football game for children in a PC.

Next Session

- System Design Tools and Techniques

