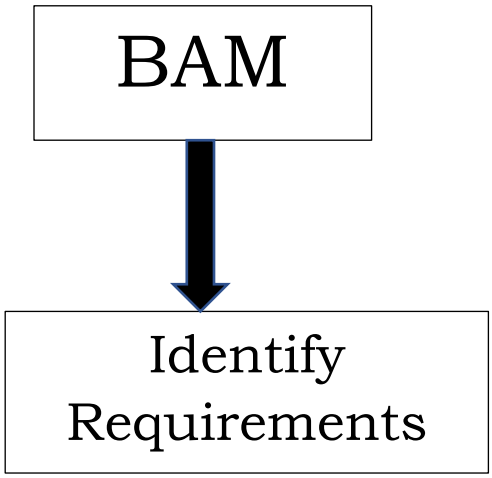
25th of January 2023 online lesson

Using BAM can analysis and understand how do their activities in the current system.

Client fully agreed with the Business Activity Diagram (drawn by the System Analyst), then go for the further.

Once you have sound understanding of the business environment, then need to identify the requirements (What are the requirements should satisfy?).




## Definition of requirements

**What are the requirements?**

They are specified conditions that are demanded by clients or stakeholders in the creation of a project.

**Requirements gathering**

SDLC (software development lifecycle) and is the process used as the framework for software development. Project managers and business organizations use the SDLC as a blueprint for completing each step of the lifecycle for software development. Each step of the SDLC is called a phase. The requirements gathering and analysis phase is the first phase of the SDLC.

Some of the commonly used requirements gathering techniques

- Interviews
- Questionnaires
- Observations
- Facilitated Workshops
- Focus groups
- Brainstorming
- Prototyping
- Documentation analysis






- Identify and then define system requirements in the **Requirement Catalog**
- Two types of requirements
  1. Functional requirements – What activities that the system should carry out
  2. Non-functional requirements – How well or within what limits functional requirements should be satisfied

  IEEE standard

  Essential (shall),Nice to have (Should),

  Actors-same given requirement can be define 2 points of view
  1. System's point of view
  2. User's point of view

# ATM Machine Requirement Catalog

| | Functional Requirements | |
|---|---|---|
| ID | Description | |
| 1 | ~~User~~ shall be able to withdraw money | Essential Requirements |
| 2 | ~~User~~ shall be able to view balance | |
| 3 | ~~User~~ should be able to deposit money - | Nice to have Requirement |
| 1 | ~~System~~ shall be able to dispense money | Essential Requirements |
| 2 | ~~System~~ shall be able to display balance | |
| 3 | ~~System~~ should be able to accept money - | Nice to have Requirement |

| | Non - Functional Requirements |
|---|---|
| ID | Description |
| 1 | ~~System~~ should provide a touch sensitive GUI – nice to have |
| 2 | ~~System~~ shall provide accuracy of 98% on withdrawal |
| 3 | ~~System~~ shall use 256 bits encryption for communication (security) |
| | |
| | |
| | |

Essential Requirements

Not strongly associated with teller functionality
(non- functional requirements)

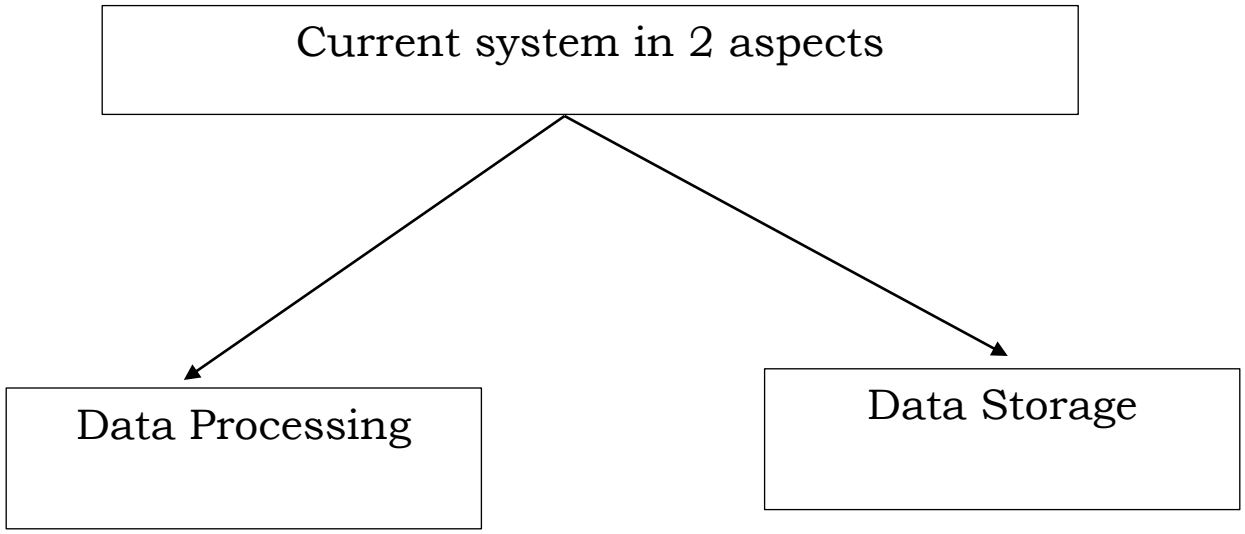
- Usability
- Reliability
- Efficiency
- Security
- User-friendliness
- Performance

Bookland Book store Requirement Catalog

| | Functional Requirements |
|---|---|
| ID | Description |
| 1 | Shall be able to keep book details |
| 2 | Shall facilitate to get hold – on requests |
| 3 | Shall be able to check book availability (status) |
| 4 | Shall facilitate to place a hold on requests |
| 5 | Shall be able to keep payment details |
| 6 | Shall be able to generate payment receipt |
| 7 | Shall be able to update inventory file |
| 8 | Should facilitate to make online enquiries through a web site |
| 9 | Should facilitate to make online payments |

| | Non - Functional Requirements |
|---|---|
| ID | Description |
| 1 | Shall provide a GUI |
| 2 | Shall run on windows XP |
| 3 | Shall make use of existing S/W and H/W |

We need to further analysis the current system.




## Data Flow Modeling

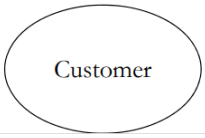
- Used to model processing of data in the system
- DFM consists of
  - A set of Data Flow Diagrams (DFDs) and
  - Associated textual descriptions
- Defines partitions into sub systems
- Presents different parts of the system at appropriate levels of details for different stakeholders of the project
  - Client
  - End-user
  - Developer

## DFD

- Illustrates
  - How data is processed within the system
  - How data is passed around the system
  - Where data is stored in the system
- Components
  - External Entities
  - Data Flows
  - Processes
  - Data Stores

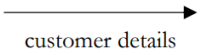
## External Entity

- Represents people, organizations or other systems external to the system under investigation
- Acts as a source or a recipient of data
- Name should refer to a generic type, not to an instance of that type
- Example



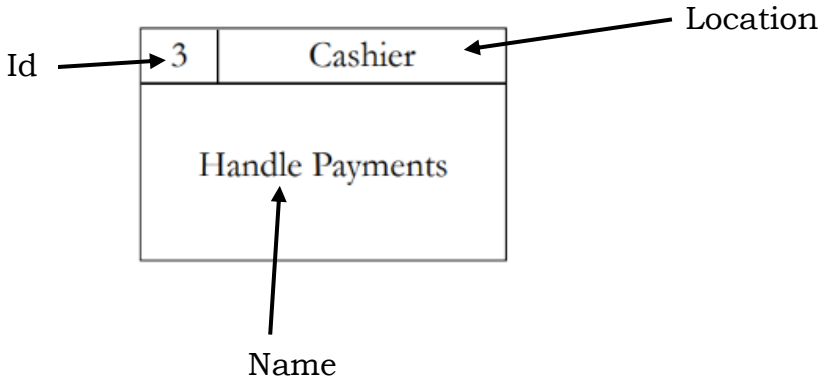

## Data Flows

- Show flows of data to, from and within the system
- Link other components in a DFD
- Could be unidirectional or bidirectional
- Represented with solid arrows
- Between two external components are shown by dashed arrows
- Intersections should be avoided
- Example



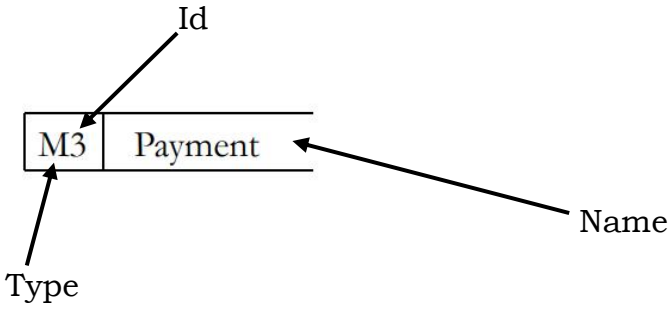
## Process

- Represents business activities carried out in the system
- Three properties
  - Id
  - Name
  - Location




## Data Store

- Used to hold data within the system
- Four types
  - Manual - M
  - Computerized - D
  - Temporary - T
  - Manual Temporary - T(M)
- Three properties
  - Id
  - Type
  - Name



## Context Diagram (Level 0 DFD)

- A DFD with the highest level of abstraction
- Represents the entire system as a single process
- Shows how system interacts with its external entities
- Provides an over-view of the system

## How to draw a context diagram?

- Identify all the external entities interact with the system
- Identify data flows from the external entities to the system as well as to external entities from the system
- Represent external entities in their graphical notation
- Represent the system as a single process
- Add data flows between external entities and the system