Kirirom Institute of Technology

Unit 4

OOAD

Use case diagrams:

- Explains the user’s perspective of the system
- High level design of the interaction

Components of Use case diagram:

- **System** (The entire project/system)
- **Actors** (People/system/organization that performs actions in the system)
- **Use cases** (All the functions of the system)
- **Relationships**

Types of Relationships:

- Association
- Include / uses
- Extend
- Inheritance

Types of use case description:

- High level - General description
- Expanded (Detailed) - Step by step
- Essential - Free of technological details
- Real - Adds technological details

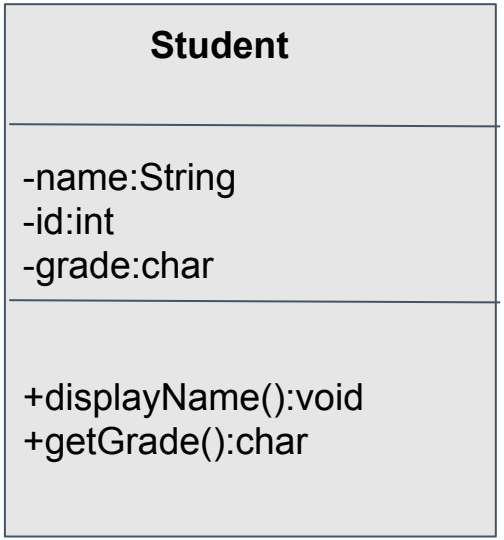
Class Diagram:

Kirirom Institute of Technology

- Popular UML diagram
- Describe the entire system in low level
- Visualize the objects and their relationships
- Defines attributes (Variables) and operations (Methods)
- Class diagram without attributes and methods is called simple classes

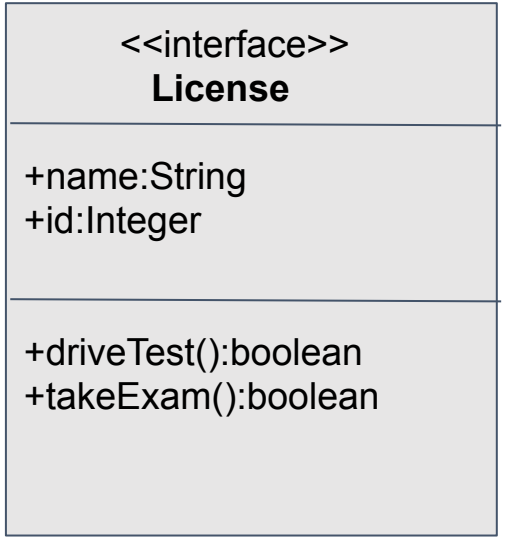
Classes:

- Fundamental and important element of the class diagram



Interfaces: <<interfaces>>

- Contains methods that has to be implemented by a class (Responsibility)



Type and control of Attributes and Methods

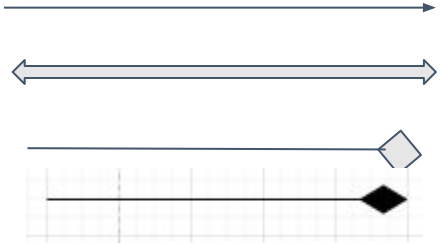
Kirirom Institute of Technology

Are defined by 3 properties:

1. Type / Return type
2. Access specifiers (public, private, protected, package)
3. Modifiers (static, unique, final, [], etc)

Relationships between classes:

- Association
  - Uni directional
  - Bi Directional
  - Aggregation
  - Composition



Example

- Check out the KIT exam system example in the below URL:

https://drive.google.com/file/d/1X-IVXZXy1sk8moxv0UWBkACfdZVcSzxl/view?usp=sharing

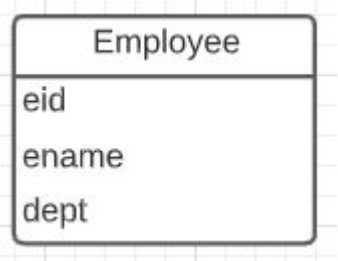
ER Diagram:

- ERD is a graphical representation of your entire database
- Mostly of High Level Design but more details can be added
- Entire database structure is represented as an ERD
- 3 important components are used (**Entity, Attributes and Relationships**)

1. Entity

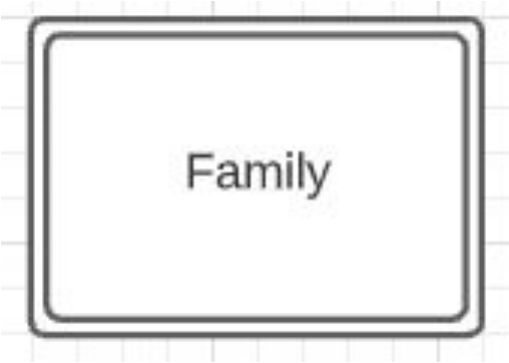
- It can be a person, object, thing etc.,
- Entity can be represented as a table
- Represented in Rectangles
- Will contain a **primary key** attribute

(OR)



1.1 Weak Entity:

- Doesn’t have a primary
- Depends on another Entity
- Represented in double rectangle



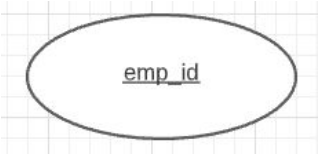
2. Attributes:

- Are names of properties (**column names**)
- Can be mentioned with or without data type (Based on the type of ERD)
- Data types are mandatory for Physical ERD
- Represented with ellipse
- Primary Keys are specified with **underline/bold/PK**
- Foreign keys are need not to be Optional

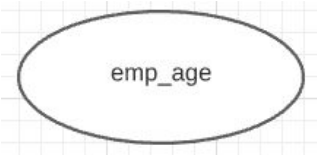
Types of attributes:

- Attributes (Primary key) : **Uniquely identifies entity**
- Attribute regular : **Ordinary attribute**
- Composite : **Can be split into two/many attributes**
- Multi values: **Can store multiple values**
- Derived : **Value obtained from other attributes**

Attribute types and their symbols:

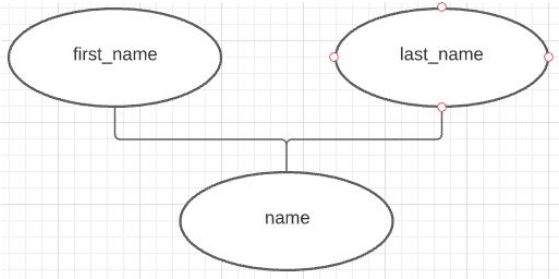


Primary Key Attributes

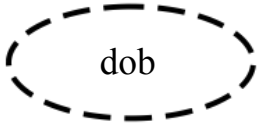


Regular Attributes

Multivalued Attributes



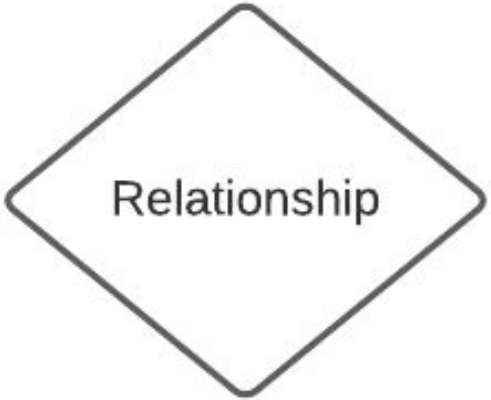
Composite Attributes



Derived Attributes

Relationship:

- Used to denote the relationship between classes
- Denoted with a diamond symbol



Cardinality:

Kirirom Institute of Technology

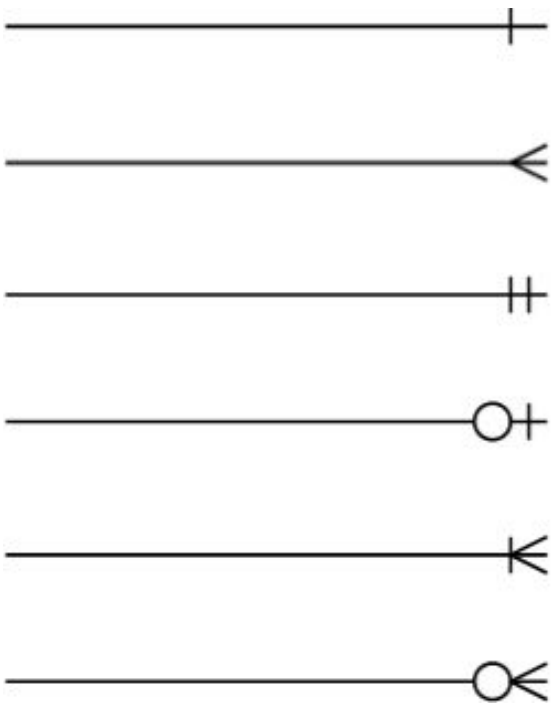
- Possible **number of occurrences** in one entity which is associated with the number of occurrences in another

- Basically there are three types of cardinalities available:

  a. One to One
  b. One to Many
  c. Many to Many

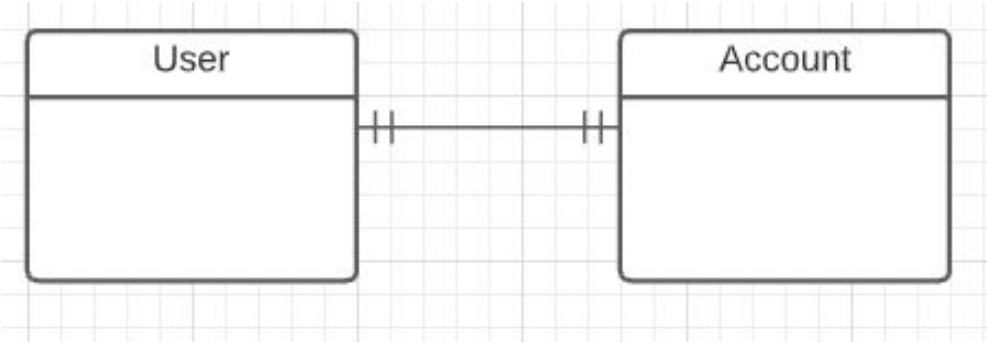
- In ERD cardinality explained with Crow's foot

Cardinality (Crow's foot):



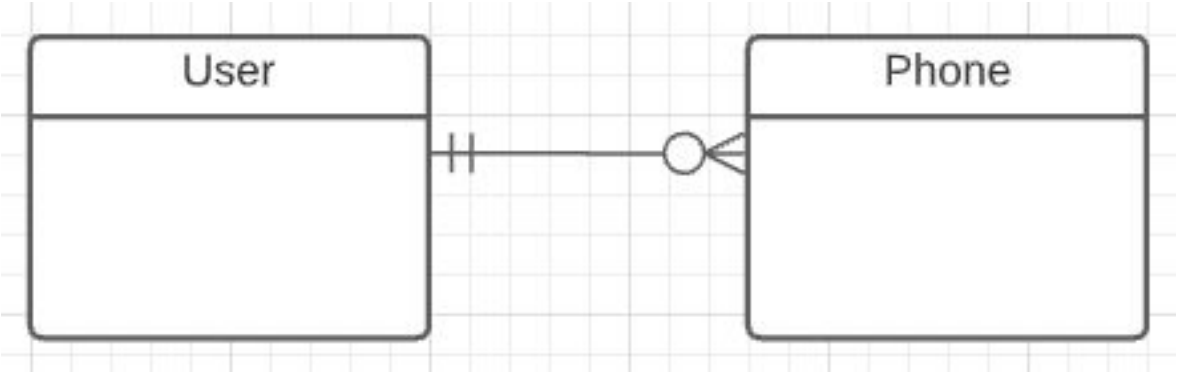
One to One:

- A user can have only one account in social media (Eg. FB, Instagram etc)



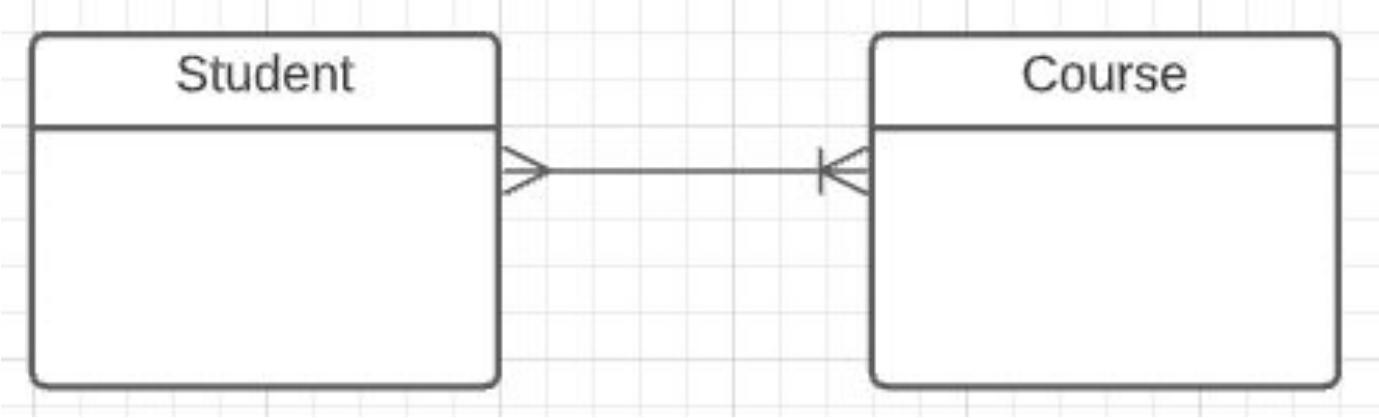
One to Many

- A user can have zero or many phones (Min: 0, Max: n)



Many to Many:

- **A student** can register for **Many courses** (Min: 1 course, Max: n)
- **A course** can be registered by **Many students**



Steps to create ERD:

1. Identify your entities
2. Identify the relationships among entities
3. Identify cardinalities
4. Identify attributes
5. Create the final ER Diagram

Data Flow Diagrams

- DFD is a graphical depiction of how data is flowing inside the system

  Provides data as well as functionality to software designers
- **Class diagram - Explains objects of the system ER diagram - Database schema**
- DF diagram - Intermediary (Explains objects, data flow and data sources)

Major components of the DFD:

Major components:

1. External entities (Rectangle)
2. Processes (Circles)
3. Data flows (Arrows)
4. Data stores (Pipe symbol)

External Entities:

- External object that consumes or provides data to the system
- Hardwares, sensors etc are external objects
- FB share, authentication etc.,

Process:

- Transforms data Gets data as inputs and converts into output)
- Must not be repeated twice

Data flows:

- Not bidirectional arrows allowed
(Use separate arrows for incoming and outgoing data)

Data sources:

- Anything that stores data (DBs, files, etc.,)

General rules:

- Data must not be directly flow from one entity to another entity. There must be a process in the middle same rule applies for entities as well

- No internal logic (Don't use programming constructs like loops, if else etc)

- Keep it uncluttered (Use joins and forks)

- There can't be any miracles or blackholes (Data sources must not only store data and not been able to read and vice versa)

- Meaningful labeling must be there

There are three levels of DFD:

- Level 0 (Context Diagram)
- Level 1 (Data Flow Diagram - System Overview)
- Level 2 (Data Flow Diagram - Detailed module)

Level 0:

- Birds eye view of the system

- Only one process to explain all the process

- No data sources in context level diagrams

- Data stores must be shown as an Entity

Level 1:

- Better view of the system

- Overview of the system but not in very detail

- Same entity given in the context level diagram must be used (No new entity added)

- Explains all the use cases and sub processes of the system

- Maximum 7 + or - 2 processes is optimal

Level 2:

- Detailed level
- Talks about all the details of a particular module