Laboratory 2

Title of the Laboratory Exercise: Requirement analysis and data modelling

1. Introduction and Purpose of Experiment

   The requirements analysis phase produce both data requirements and functional requirements. The data requirements are used as a source of database design and should be specified as detailed and complete form as possible. In data modelling, the designers first create a conceptual model of how data items relate to each other. By doing this lab, students will be able to perform data modelling of the application.

2. Aim and Objectives

   Aim

   - To analyse the given application and create a data model

   Objectives

   At the end of this lab, the student will be able to

   - Identify functional and data requirements from problem statement
   - Create a data model from the data requirements

3. Experimental Procedure

   - Read the problem statement and identify requirements
   - Perform data modelling
   - Document the requirements and ER diagram

4. Question

   Students have to choose one of the following problem statements and develop the software solution. The Course leader is the customer. Contact the Course leader for any clarificatpoions.

   1. Library management system
   2. Hospital management system
   3. Employee management system

   Perform the following based on the problem statement you have chosen

   a. Analyse the given application and list the functional and data requirements
   b. Perform data modelling based on the identified data requirements

5. Calculations/Computations/Algorithms

Functional Requirements

1. The system should allow the manager and the employee to login
2. The system should allow the manager to add employees and details to the database
3. The system should allow the manager to modify employee details
4. The system should allow the manager to assign projects to the employees
5. The system should allow the manager to add and manage departments
6. The system should allow the manager to manage employee logistics

Requirement Tag	FR1
Requirement Description	The system should allow the manager and the employee to login
Dependent on Requirements	None
User/System interacting with the requirement	Manager

Requirement Tag	FR2
Requirement Description	The system should allow the manager to add employees and details to the database
Dependent on Requirements	FR1
User/System interacting with the requirement	Manager

Requirement Tag	FR3
Requirement Description	The system should allow the manager to modify employee details
Dependent on Requirements	FR1, FR2
User/System interacting with the requirement	Manager

Requirement Tag	FR4
Requirement Description	The system should allow the manager to assign projects to the employees
Dependent on Requirements	FR1, FR3
User/System interacting with the requirement	Manager

Requirement Tag	FR5
Requirement Description	The system should allow the manager to add and manage departments.
Dependent on Requirements	FR1
User/System interacting with the requirement	Manager

Requirement Tag	FR6
Requirement Description	The system should allow the manager to manage employee logistics
Dependent on Requirements	FR1, FR2
User/System interacting with the requirement	Manager

Data Requirements

Requirement Tag	DR1
Item Name	Manager Details
Item Description	Maintains the manager details ID, Username, Password
Item Type	Integer, String, String
User/System interacting with the requirement	Manager

Requirement Tag	DR2
Item Name	Employee Details
Item Description	Maintains the employee details ID, Name, Type, Contact Number, Address, Last Paid On
Item Type	Integer, String, Enum<String>, Array<Integer>, Array<String>, Date
User/System interacting with the requirement	User

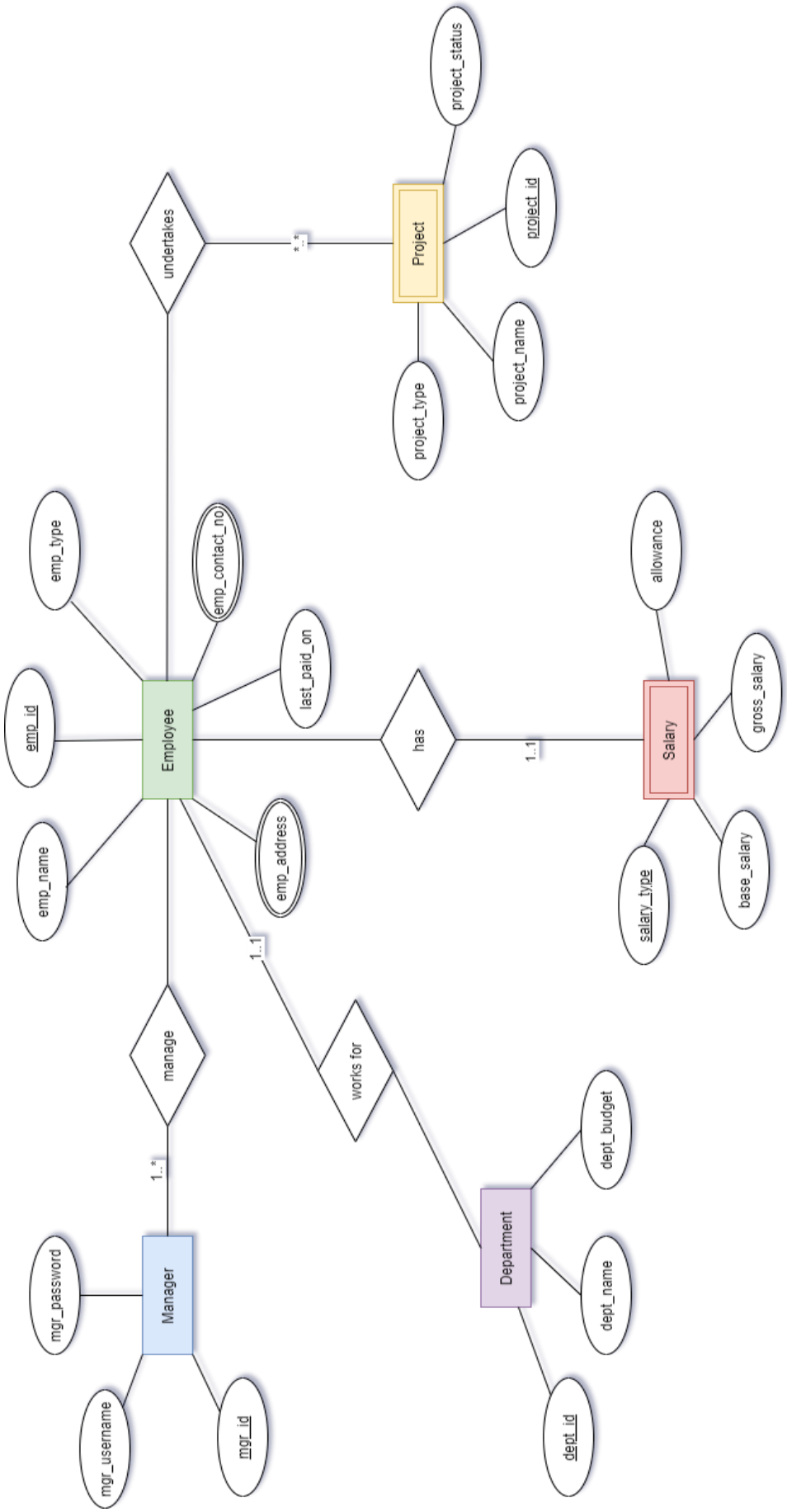
Requirement Tag	DR3
Item Name	Project Details
Item Description	Maintains the project details ID, Name, Type, Status
Item Type	Integer, String, Enum<String>, Enum<String>
User/System interacting with the requirement	User

Requirement Tag	DR4
Item Name	Salary Details
Item Description	Maintains the Salary details Type, Base Salary, Gross Salary, Allowance
Item Type	Enum<String>, Integer, Integer, Integer
User/System interacting with the requirement	User

Requirement Tag	DR5
Item Name	Department Details
Item Description	Maintains the Department details ID, Name, Budget
Item Type	Integer, String, Integer
User/System interacting with the requirement	User

6. Presentation of Results

NAME: SATYAJIT GHANA REG NO: 17ETCS002159



7. Conclusions

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other.

ER diagrams constitute a very useful framework for creating and manipulating databases. First, ER diagrams are easy to understand and do not require a person to undergo extensive training to be able to work with it efficiently and accurately. This means that designers can use ER diagrams to easily communicate with developers, customers, and end users, regardless of their IT proficiency. Second, ER diagrams are readily translatable into relational tables which can be used to quickly build databases. In addition, ER diagrams can directly be used by database developers as the blueprint for implementing data in specific software applications. Lastly, ER diagrams may be applied in other contexts such as describing the different relationships and operations within an organization.

8. Comments

1. Limitations of Experiments

- Limited expressiveness
- Not concise
- Can be ambiguous
- Mostly for relational database only.

2. Limitations of Results

- Loss of information content: Some information be lost or hidden in ER model
- Limited relationship representation: ER model represents limited relationship as compared to another data models like relational model etc.
- No representation of data manipulation: It is difficult to show data manipulation in ER model.
- Popular for high level design: ER model is very popular for designing high level design
- No industry standard for notation

3. Learning happened

- Conceptually it is very simple: ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.
- Better visual representation: ER model is a diagrammatic representation of any logical structure of database. By seeing ER diagram, we can easily understand relationship among entities and relationship.
- Effective communication tool: It is an effective communication tool for database designer.
- Highly integrated with relational model: ER model can be easily converted into relational model by simply converting ER model into tables
- Easy conversion to any data model: ER model can be easily converted into another data model like hierarchical data model, network data model and so on.

4. Recommendations

- Try to make unambiguous ER diagrams
- Make it more concise
- Try to be more expressive in the diagram