EMPLOYEE - TRACKING SYSTEM

High Level Documentation

ineuron

Enterprise java batch with springboot

SUBMITTED BY: Paramita Pal
[paramitapp10@gmail.com]

CONSOLE BASED PROJECT

CONTENTS

ABSTRACT

The Employee tracking application is prepared as a console-based core java project. The system tracks the performance of all registered employees in an organisation, involved in different projects. The detailed address of the employees is stored in a different entity and may be referred to when tracking of employees is necessary or their proximity to each other and involvement in different projects may need to be conveniently assigned.

The Manager entity of the system benefits the managerial staff to track the employees efficiently to know their allotment, department, progress, and scheduling data.

This system requires no web server and may be executed from a digital device within the onsite premises of an organisation which can effectively use it to understand the performance of its employees.

The system helps in

- Division of labour
- Understanding effectiveness of each employee
- Understanding employee potential
- Project constraints
- Effective scheduling
- Effective running of organisational framework
- Less cost involvement
- Tracking employee details

1. INTRODUCTION

1.1 Need of a High-Level Design Document:

The High-Level Design Document contains a detailed description of the application to be developed. High Level designing contains data flows and data structures to help developers in understanding and implement how the current system is being designed to function.

The document provides:

- explaining the connections between system components and operations which depict the logic
- architecture design needed for the system's functionality for each and every module of the system.
- The high-level design documentation presents the system structure as the application architecture, application flow and technology architecture.

Composition of HLD:

- Attributes and features of software entities.
- Relationships between different software entities.

Characteristics of HLD:

- A diagram representing each design aspect is included in the All design of high-level design, which is mostly based on business requirements.
- Description of hardware, software interfaces and user interfaces.
- The workflow of users' typical process is detailed, along with performance specifications.
- The projects architecture and design are contained in HLD.

Purpose of High-Level Design:

- Add necessary detailed description to represent a suitable model. This is designed to help with operational requirements and can be used as a reference material for how the modules act.
- HLD is a technical representation of functional requirements and the flow of information across assets and components.

1.2 Scope

This software system will be a console-based application with Core Java. This system will be designed to gather a first-hand information on which employee is involved in which project, their department name and location details.

In the application, employees may be tracked by the employer for the benefit of creating a clear understanding of their employees.

The project deals with a employee view as well as a manager view, to get a ready information on employees and projects.

1.3.Overview

- The HLD will:
  - present all of the design aspects and define them in detail
  - describe the user interface being implemented
  - describe the hardware and software interfaces
  - describe the performance requirements
  - include design features and the architecture of the project
  - list and describe the non-functional attributes like:

- security
- reliability
- maintainability
- portability
- reusability
- application compatibility
- resource utilization
- serviceability

1.2. Project Objectives

1.2.1. Admin Objectives

It is an admin dependent system which has no user functionality. The salient features of the Employee Tracking System are:

- It is easy to execute the java classes which is possible just with the presence of jdk .
- It is an instant system of entry and checking entries of employee details, which involve less complication and hassle.
- It does not involve a web server at present as it does not give a facility of browser execution or deployment as such.
- It does not involve a database. It therefore hardly requires storage space.
- However it facilitates updation as it is prepared in an IDE and can be easily converted into a web based project using hibernate + MySQL or any other database in future.

1.3 Constraints

A few functionalities could not be implemented.

- The system suffers from the drawback of not being able to function at times due to session creation error for the application.
- The employer or the manager constructor which is created as a inherited class from the employee class, has no existence if employees do not exist. So, Manager class bears a IS-A relationship with the Employee class. The Manager class is created as the child class of Employee. Employee is the parent class of the manager class.
- Address bears a HAS-A relation with Employee. It is an Association of Composition type, with the Employee class.

Mysql was not used as JDBC was not to be used in this project.

1.4 Risks

Document specific risks that have been identified or that should be considered.

- The authenticity of users is not determined.
- The Console based project is not equipped to provide detailed view of employees. The database is not used which makes calling of employees view on the basis of employee id not possible.
- The project may suffer from updation and deletion anomalies for which the development need to be converted to different classes for insertion or deletion.

1.5 Out of Scope

1. The project is developed in the local system and the code in github repository.
2. The project is not hosted in any cloud platform and is still not available in on internet.
3. Setting of parameters in database for further use is not possible.
4. Employee authentication or the authentication of user of the application is out of scope.

2. TECHNICAL SPECIFICATIONS

It includes the hardware software and other technical requirements of the system. Any platform and machine with an installed jdk can effectively run the application. As the project is developed in java it is:

- Portable
- Simple to understand and implement
- Any database can be used for stating minimum employee details
- Minimum storage and any RAM that effectively runs a java application can be used.
- The application uses the System Library which may vary from machine to machine.
- The application is developed in Eclipse IDE. It may be opened in any IDE or it may even run in command prompt.

2.1. Hardware Requirements

- Processor: Intel Pentium microprocessor with RYZEN
- Main memory: 512 MB
- Hard disk : 256 MB required
- Keyboard: Standard
- Monitor: 600x800 Resolution or above
- Mouse: Scroll

2.2. Software Requirements

2.2.1. Tools and platforms used

- Operating System: Windows11
- Platform: ECLIPSE IDE [2022-09]
- Language: CORE JAVA

2.2.2. Software interfaces

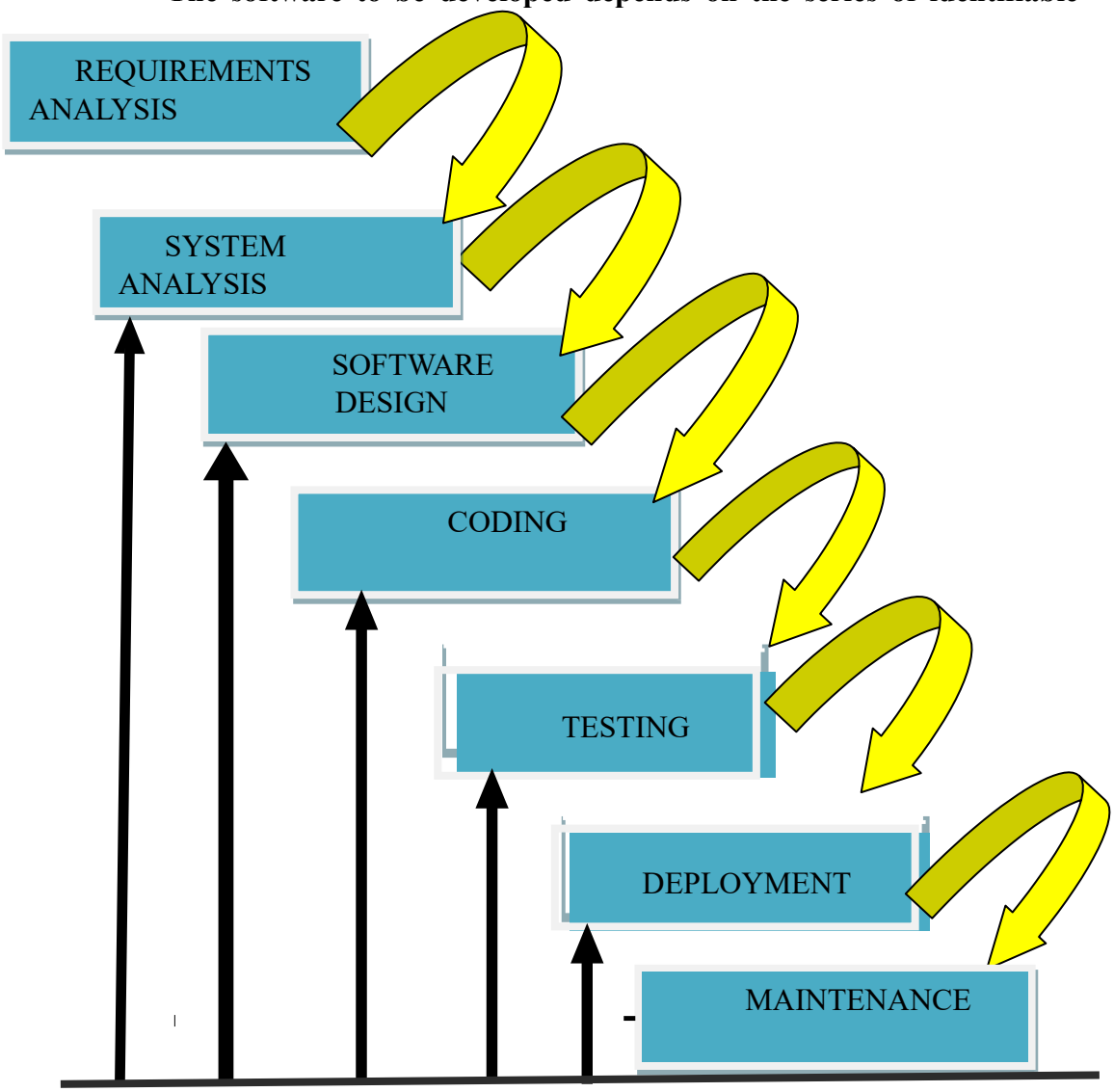
- Application: Eclipse IDE [2022-09]
- jdk : 8
- Additional API: MS OFFICE

No **DATABASE** used and no schema needed to be developed. Technological Specification was not a matter of concern for the project.

3. SOFTWARE DEVELOPMENT METHODOLOGY

3.1. SOFTWARE LIFECYCLE MODEL

The software to be developed depends on the series of identifiable



stages that would eventually lead to the product. The diagrammatic representation would follow in building the logical framework. It would be based on the requirements analysis and the design phase. The proposed software is planned after the requirements analysis and therefore may be developed on the SDLC method of the iterative waterfall model. In order to give it a stage to analyze the effects of theprior stage, the iterative approach has been followed.

4. REQUIREMENTS ANALYSIS

4.1. PROBLEM SPECIFICATION

The system to be developed is based on core java as a console-based project. The system is developed to track which employees are involved in which project and their details.

- Employees and their details are inserted in the system.
- Employee details, as Address may be updated in the system.
- Employees may be removed from a project.
- A Manager view is created in the manager class.
- No database and no connector are involved. Employees are stored in employee array.
- Linked hash map is used for a quick view of empid and employee_ name.
- The code is portable and easily maintained. It may well be enhanced and modified.

4.2. FEASIBILITY STUDY:

The development of any Software depends on the fact that whether it is feasible for development or not. This study is done for the various factors which might affect the software development, deployment and maintenance. It is also targeted for Customer Relationship Management in future.

4.2.1. TECHNICAL FEASIBILITY:

The software is a simple console-based application. It is purely coded in java.

The Software is to be managed and maintained after deployment

on an iterative basis,

i.e. updating requirements according to changing needs. As maintainability is easy and development is based on technical updations of academics and availability, the Software is technically feasible. The Software is least prone to attacks. It is to obtain a check on the go.

4.2.2. SOCIAL FEASIBILITY: This software is socially feasible as a fast checking of projects and employees involved in the projects. Visibility to employee details especially with their address and contact details becomes a necessity.
Creation of a manager view is of importance to checking employee records. The present system is a handy method of instant updating and checking.

4.2.3. ECONOMIC FEASIBILITY: The system does not need any installations. It does not involve a database or a web server. As it is coded in java it can be run on cmd if an IDE is not available. It implements the java features and the code being portable is a handy method.

4.2.4. LEGAL FEASIBILITY:

The Legal feasibility of the system development is based on whether the employee details checked are authentic or not. As employees are generally distributed and assigned projects according to their domain and specific study fields, they might need to share details with one another. This problem makes it necessary to understand the Residential Address authenticity of the employees. This problem may be solved in future by employing database storage facilities with accepting files for address proof.

The feasibility study conducted on the system has helped its development and further maintenance

4.3. SOFTWARE REQUIREMENTS SPECIFICATION

This part of the document provides a comprehensive description of the Software to be developed by the system. The different subsections provide the information of the Software and hardware to be used by the system.

4.3.1. PURPOSE: The SRS aims at the development of the system requirements. The employee tracking system is a must have for all concerned organizations.

The vision of the system is a unified platform of Employee details to track their whereabouts and their involvement in projects.
The mission is to strive towards the goal with a reduction of insertion and deletion anomalies to have a better track of every employed person involved in a company.

4.1.2. SCOPE: The system may be used by an individual data entry operator of an organisation with very few instructions to follow.

- The system runs offline and requires no web server installation.
- It gives instant entry and check facilities with less typing and entry hassles.
- The portable system created in pure core java, runs at present on the simple build level without any database and has a lot of maintenance scope and can be easily connected in time with any database just by integrating with Hibernate or by usage of JDBC.

- .ABBREVIATIONS:

SRS	Software Requirements Specification
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
ID	Identification Definition
jdk	Java development Kit
jre	Java runtime environment
IDE	Integrated Development Environment

Table : 5.1.

Specification: IEEE STD 830-1993

4.1.3. FUNCTIONAL REQUIREMENTS:

This gives specific details about how the system is supposed to behave after execution in the virtual machine. It also gives what inputs are provided to which process and what is the expected output of each. It also denotes how the system might behave and what are the specific data manipulations and calculations.

4.1.3.1. Employee insertion [address class gets executed first]:

Employee address is given as an input

Input: Address address (), zip_ code, street, city, state, country

Output: address input success

4.1.3.2. Employee class insertion:

Input: empid, employee _ name, employee _ email, employee_ phone, project_ name

Output: employee class inserted successfully

4.1.3.3. Manager class insertion:

Input: department_ name, no _ individuals _ reporting, timesheet _ begin , timesheet _end,

Employee object

Output: status, save()

4.1.3.4. Employee selection:

Input: empid

Output: employee_ name, employee_ email, employee_ contact, project_ name, address, department _ name,

4.1.3.5. Employee updation(address):

Input: empid, Address new _address

Output: save(), status

4.1.3.6. Employee deletion :

Input empid

Output: status , save()

4.1.4. NON-FUNCTIONAL REQUIREMENTS: These are directly related to thefunctioning of the system. The main constraints of the system are

4.1.4.1. Hardware interface:

- Screen Resolution of 600x800.
- Mouse for scroll
- PC or Laptop
-

4.1.4.2. Software Interfaces:

- Eclipse IDE for developing java Code
- Windows 11 OS

4.1.4.3. Communications Interfaces: None

4.1. 3..6.

Performance
requirements:
java classes
jre used : 1.8

4.1.5. Software System attributes:

- Maintainability: This is achieved by updating the system on the basis of present requirements and implementation of Client demand techniques, as gathered fromfeedback of users.
- Portability: This feature is achieved by using JAVA as a programming language. The OOPS feature helps the system to have a portable feature. It is therefore made to run onany Operating System on any machine.

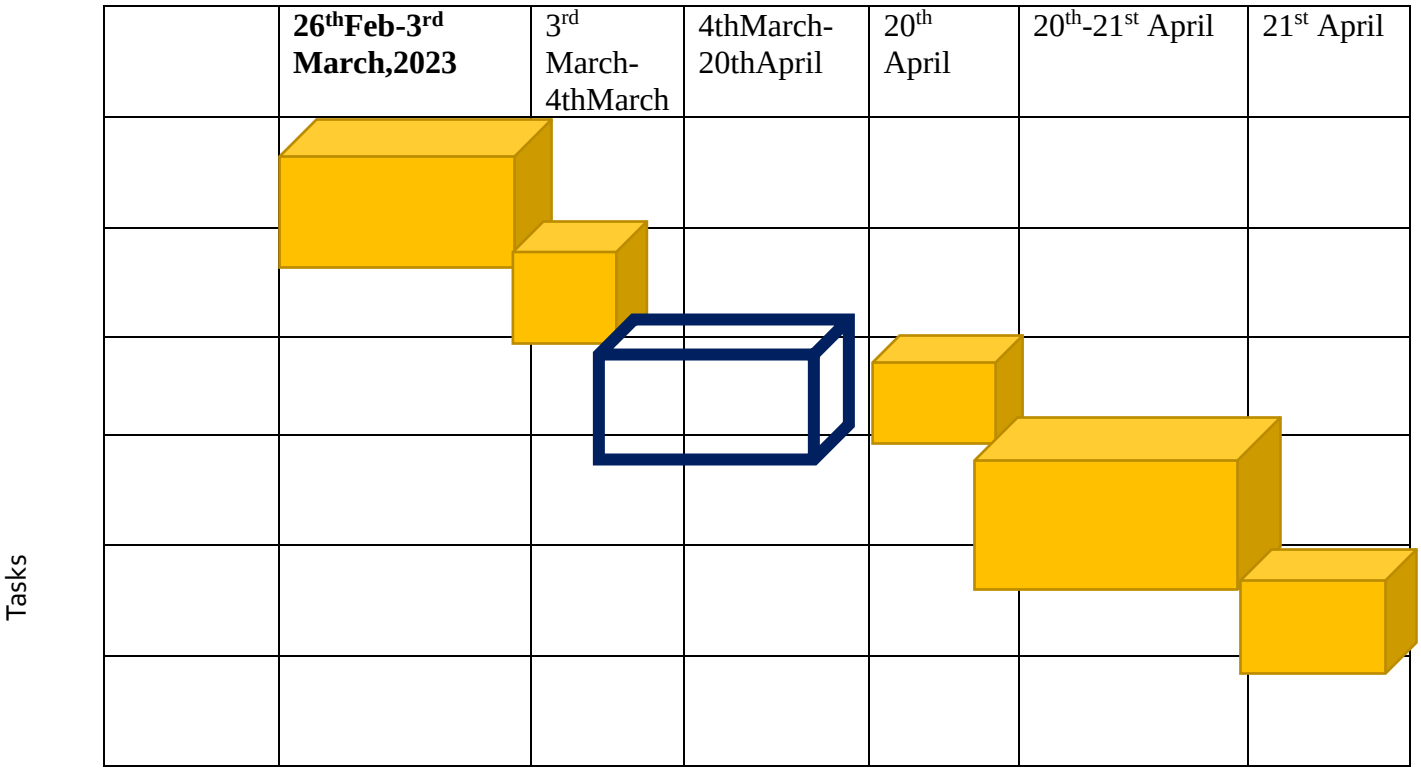
5. PROJECT PLANNING AND SCHEDULING

NAME OF PHASES	SUB PHASES OR DESCRIPTION	
Requirements analysis	Problem definition	
	Feasibility study	
	Software requirements analysis	
Milestone : Successful SRS and Feasible System. Proceeding to Designing the System		
System analysis	Project planning and Scheduling	
	System DFD	
	System Designing	
	Structure designing	
Milestone :Completion of Design. Proceed to code the System		
Coding	Coding with Comments	
	Code Efficiency	
	Error handling	
Milestone : Error free Code		
	System Testing	
	Debugging	
Milestone : Successful Testing		
Implementation and Maintenance	improvement	
Milestone : Successful Implementation		

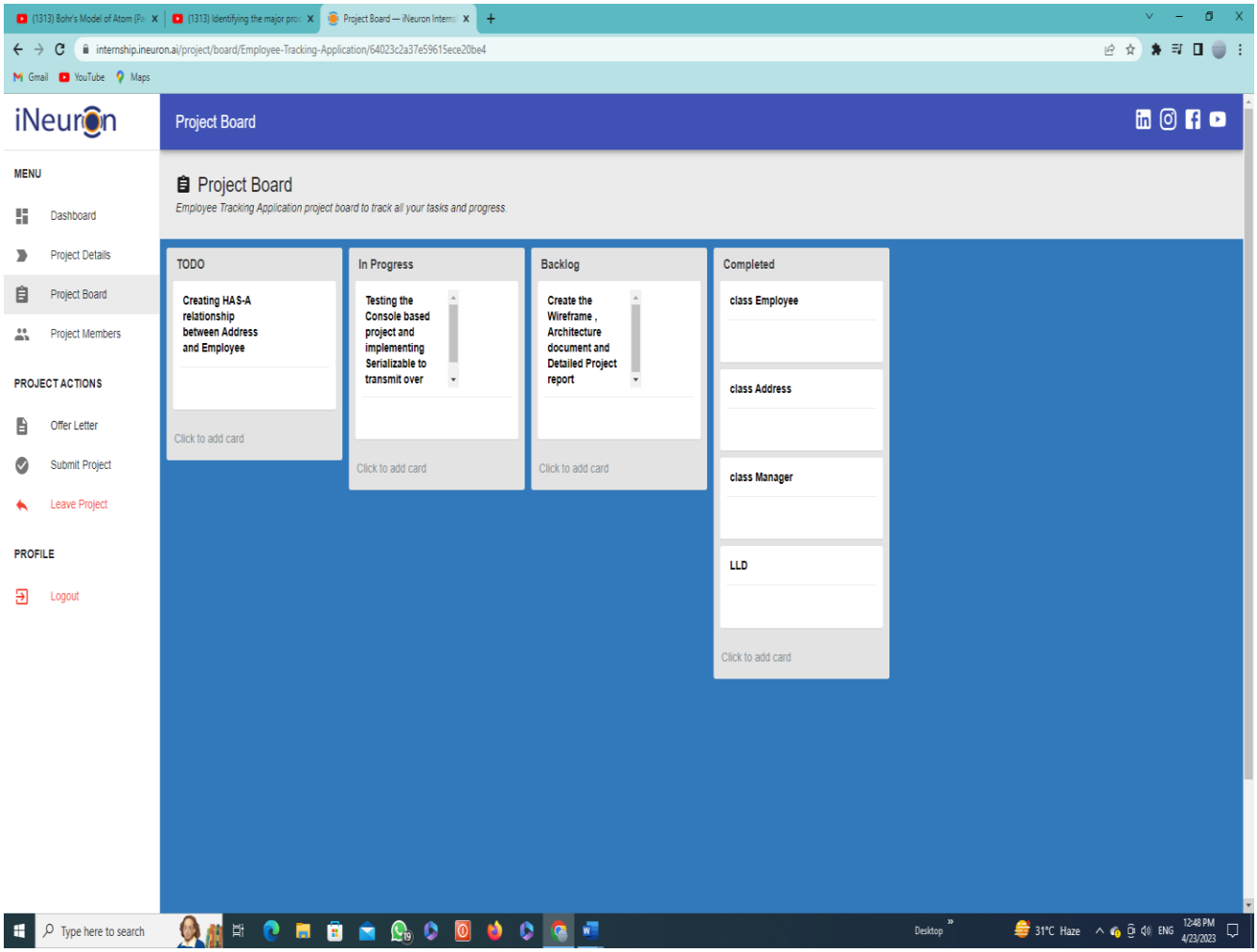
5.1. GANTT CHART: A horizontal bar chart which visually represents a project planover time. The chart shows status of each task in the project.

TASK	START DATE	DAYS TO COMPLETE
Requirements analysis	**26th February, 2023**	**5 days**
System Analysis	**3rd march**	**1 day**

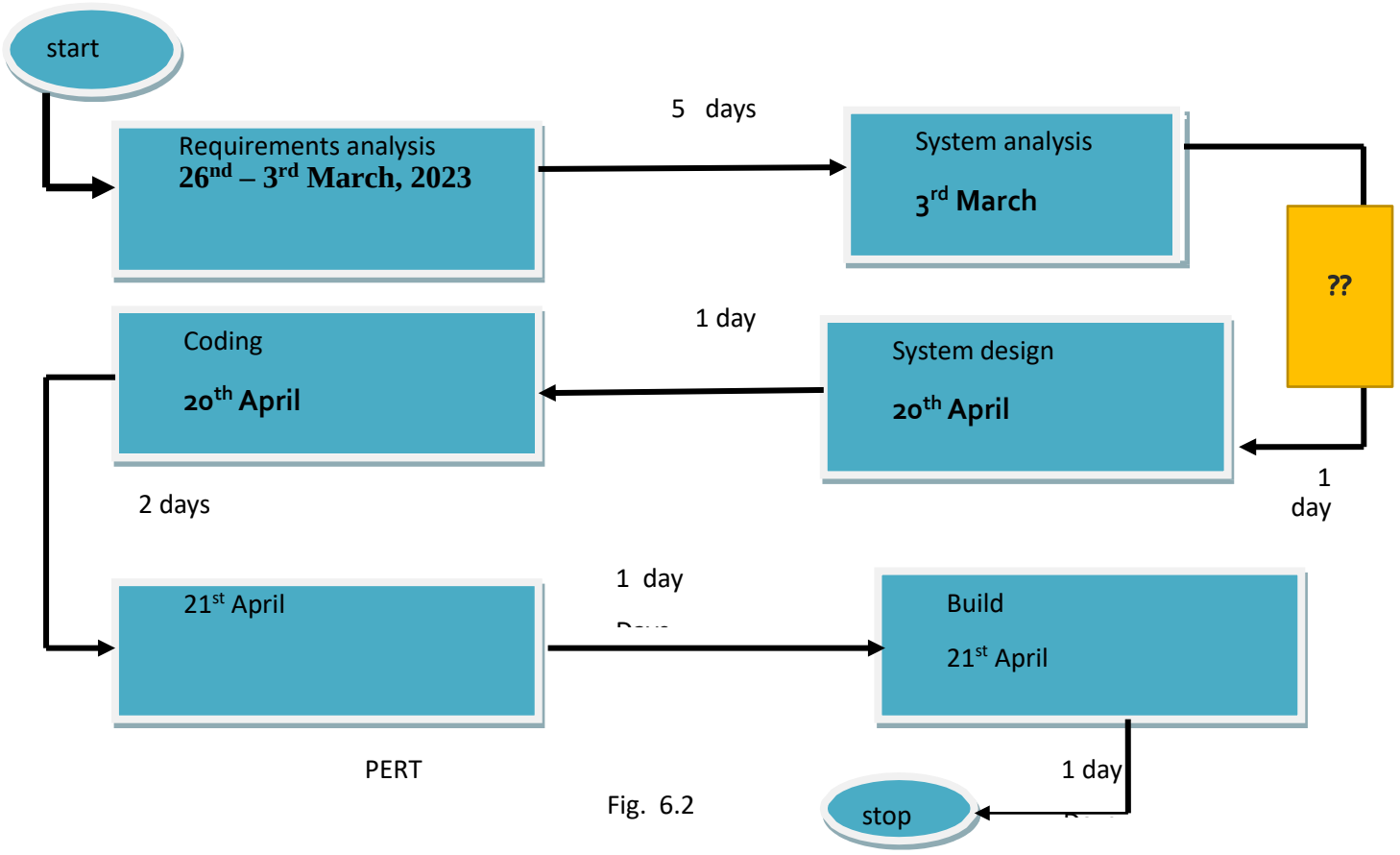
Procrastination + trial with Hibernate Integration		
System Design	**20th April**	**1 day**
Coding	**20th April-21st April**	**2 days**
Testing	**21st April**	**1 day**
Build	**21st April, 2023**	**-**



5.2. Kanban board the kanban board was used in ineuron platform



5.3. **Pert Chart:** A PERT Chart is a project management tool that provides a graphicalrepresentation of a project's timeline. The **Program Evaluation Review Technique** breaks down the individual tasks for project analysis.



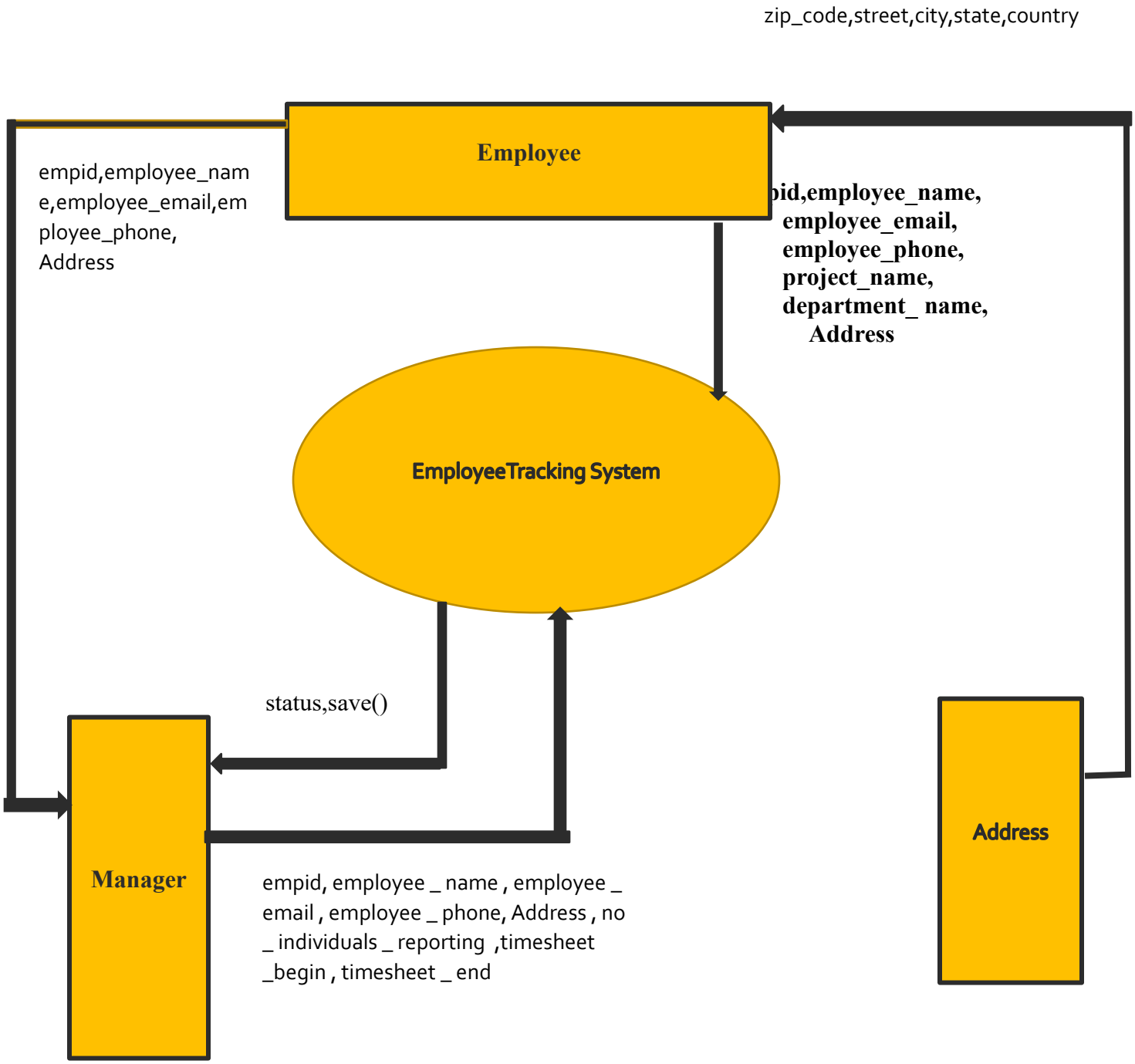
PERT

Fig. 6.2

6. SYSTEM ANALYSIS

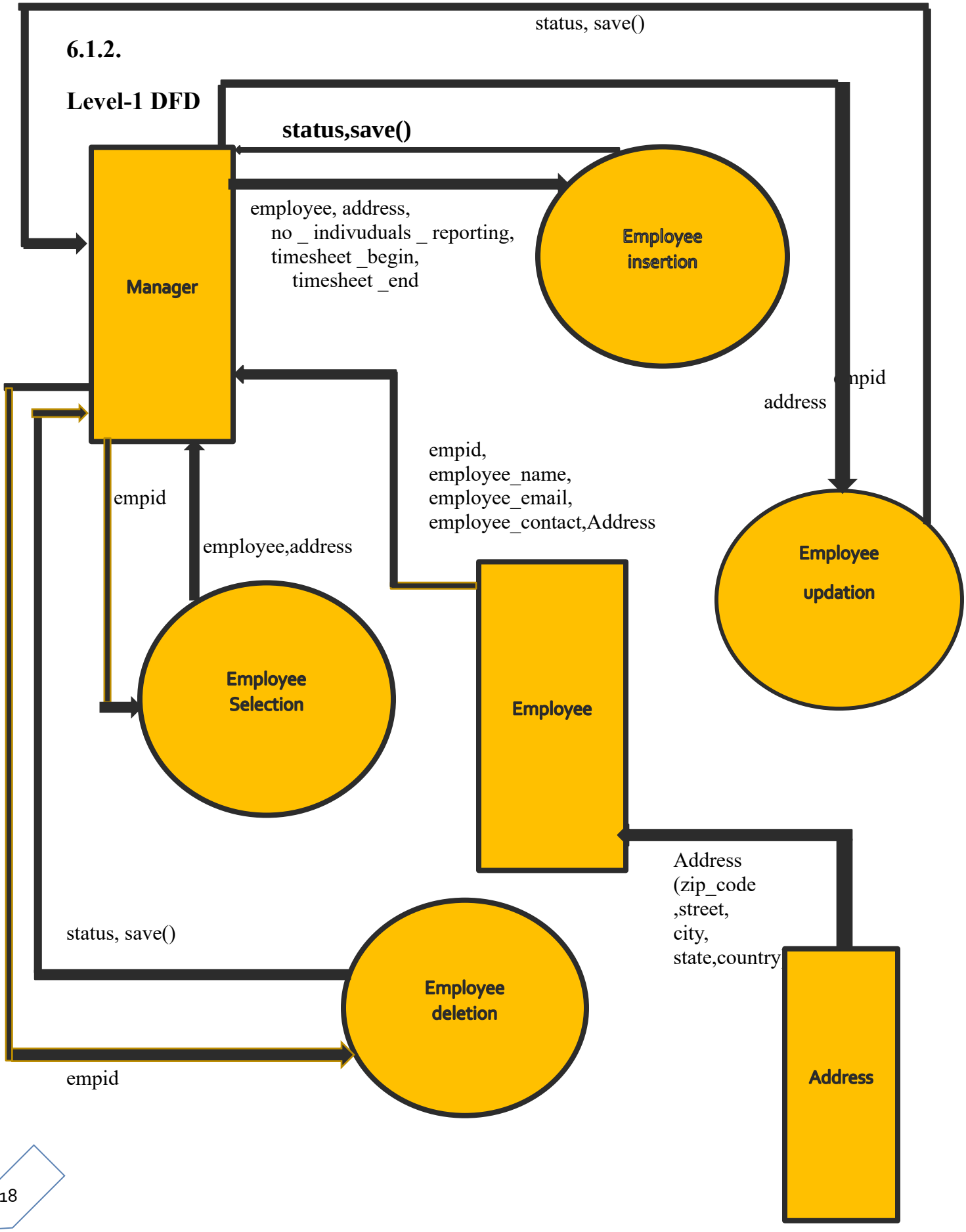
6.1. DATA FLOW DIAGRAMS (DFD): Drawn to show the flow of data between processes and entities of a System. It has no control flow. It is a mapping to demonstrate the flowof input and output from a process or entity.

6.1.2. CONTEXT LEVEL (ZERO LEVEL) DFD OF THE SYSTEM



6.1.2.

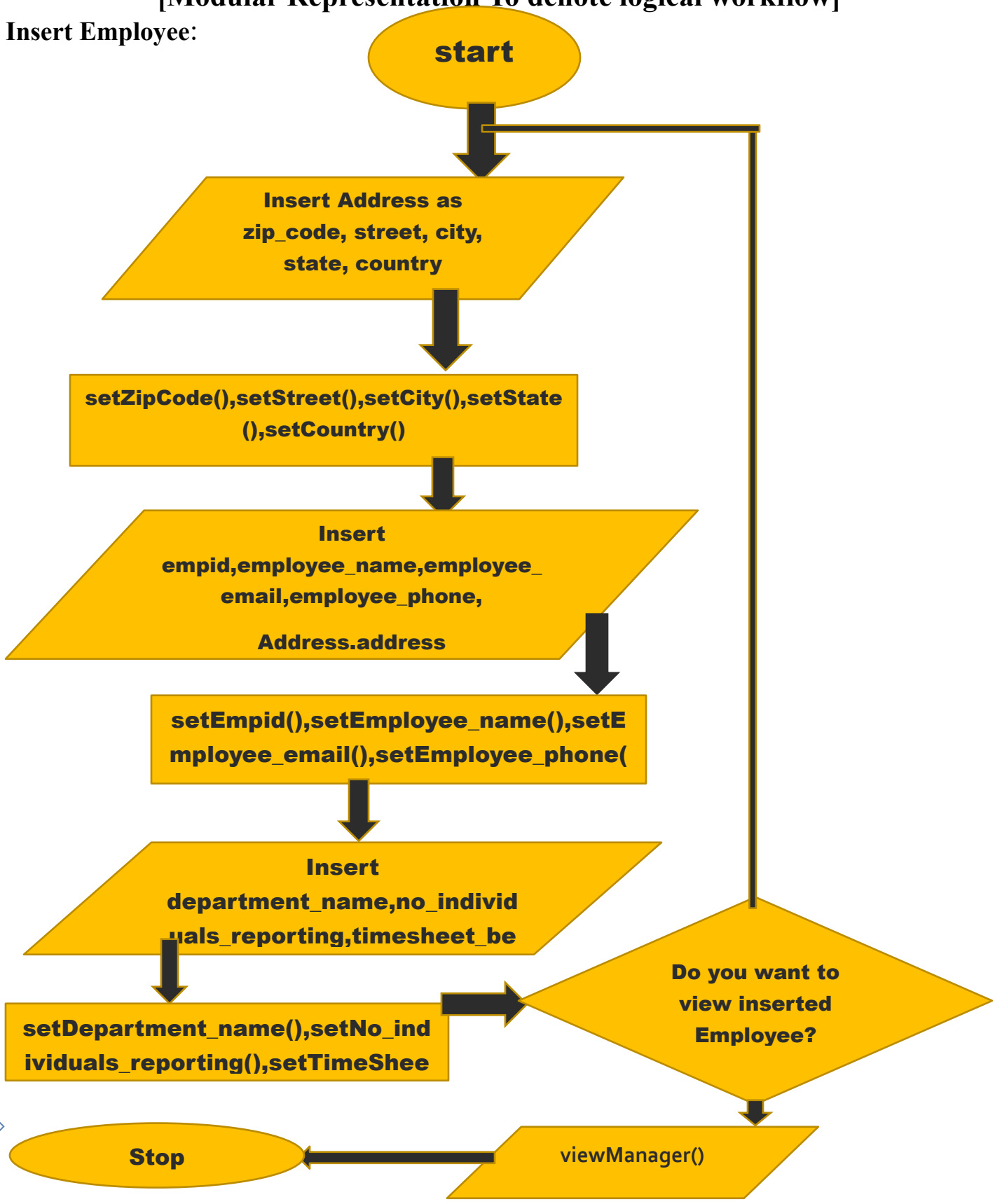
Level-1 DFD



6.2. FLOW CHART

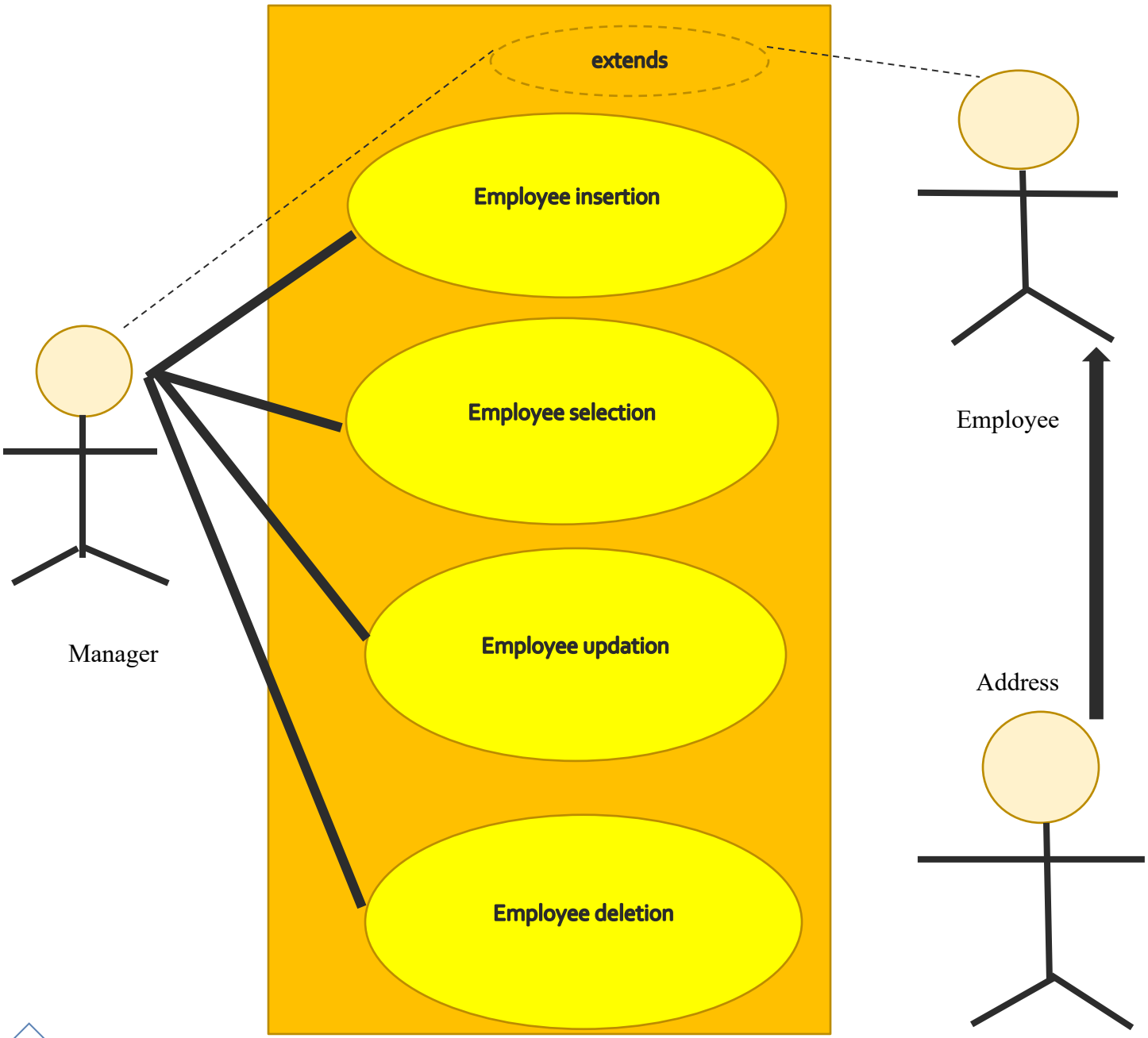
[Modular Representation To denote logical workflow]

Insert Employee:

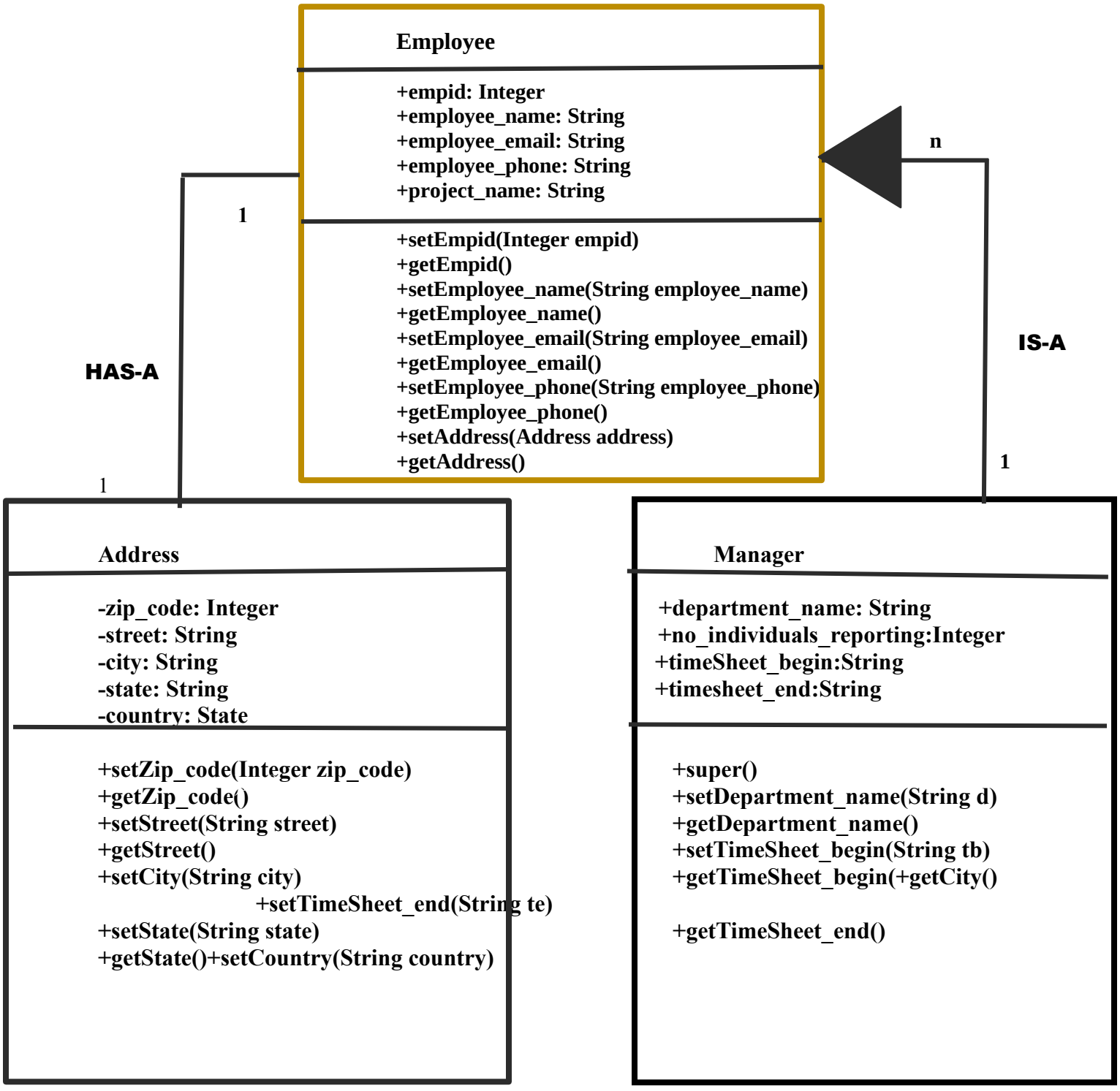


7. SYSTEM DESIGN: The design phase focuses on the detailed implementation ofthe system recommended in the feasibility study. Emphasis is on translating performance specifications into design specification. The design phase is a transition from a user- oriented document (system proposal) to a document oriented to the programmers or data base personnel.

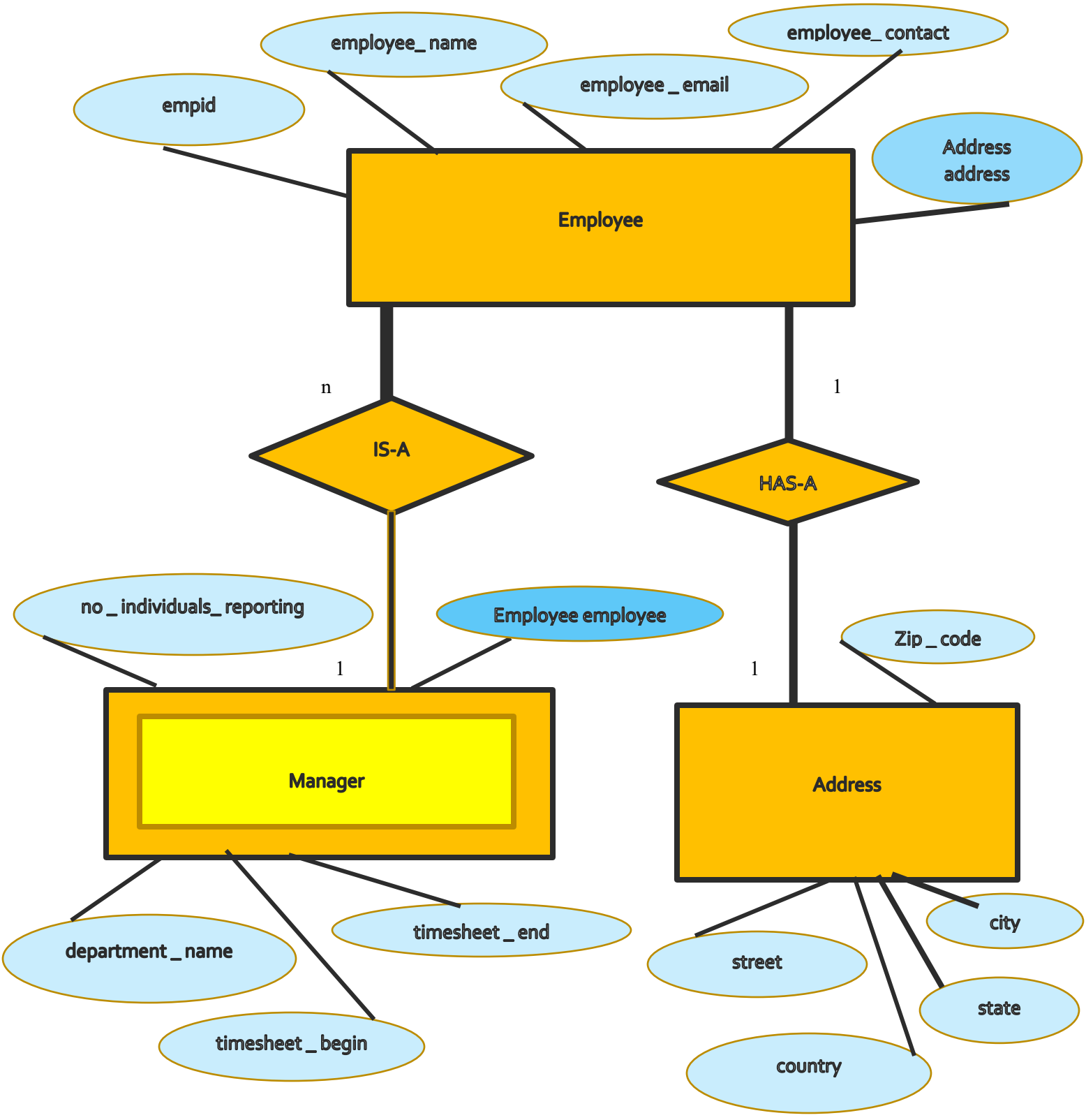
7.1. Use case diagram :



7.2. CLASS DIAGRAM: It is used as a mapping to design systems in Object Oriented languages. It is a static representation of each class, interface, association and constraint involvedin designing the system.



7.3. ENTITY RELATIOSHIP DIACRAM



8. SYSTEM TESTING

Software testing is a crucial element and it represents the ultimate review of specification design & coding. There are two types of test approaches. They are-

- Black Box Testig
- White box testing

When computer software is considered, *black-box testing* alludes to tests that are conducted at the software interface. Black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information is maintained.

Other Testing methods are:

- Unit testing:

Unit testing focuses verification effort on the smallest unit of software design—the software component or module. Here each module is tested individually and it was ensured that all the modules function as required.

Integrated testing:

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

The objective is to take unit tested components and build a program structure

that has been dictated by design. Here the tested modules is combined together and were tested to work properly as a group of interactive modules. The main purpose of this testing is to check the interface between the modules.

9. FUTURE SCOPE:

- **Hibernate integration**
- **Database connectivity**
- **Connecting jars in build path of project**

PROJECT AVAILABLTITY:

Project-code url in github repository:

https://github.com/paramita22/EmployeeTrackingSystem/tree/main/EmployeeTrackingSystem

Project post in Linked in:

https://www.linkedin.com/feed/update/urn:li:activity:7055241634241785856/

Project demo video posted in linkedin:

https://www.linkedin.com/feed/update/urn:li:activity:7055237843085885440/