

ASSIGNMENT NO. 5

Title : Design And Implementation Design Model From Analysis Model.

Objectives :

- To Identify Design level Classes.
- To Draw Design level class Model using analysis model.
- To Implement Design Model-class diagram

Problem Statement :

- Prepare a Design Model from Analysis Model
- Study in detail working on systems/projects.
- Identify Design classes/ Evolve Analysis Model. Use advanced relationships. Draw Design class Model using OCL and UML2.0 Notations. Implement the design model with a suitable object-oriented language.

Theory :

The analysis model is refined and formalized to get a design model. Design modeling, means refinement to analysis level models to adapt to the actual implementation environment. In the design space, yet another new dimension has been added to the analysis space to include the implementation environment. This means that we want to adopt our analysis model to fit in the implementation model at the same time as we refine it.

1. Creating Design level Class Diagrams

Class diagrams model the static structure of a package or of a complete system. As the blueprints of the system, class diagrams model the objects that make up the system, allowing to display the relationships among those objects and to describe what the objects can do and the services they can provide.

2. Class diagrams

In UML, class diagrams are one of six types of structural diagram. Class diagrams are fundamental to the object modeling process and model the static structure of a system. Depending on the complexity of a system, you can use a single class diagram to model an entire system, or you can use several class diagrams to model the components of a system.

Relationships in class diagram:

1. Abstraction relationships: It is a dependency between model elements that represent the same concept at different levels in abstraction. In this, the derived models are more refined than the parent models.
2. Aggregation relationships: It is used to define a 'has-a' relationship. Basically, it means, several classes come together to create a single class. They 'aggregate' together to create a new class. For e.g. in a person class, address, education etc. are aggregated to create the person class.
3. Generalization relationships: Generalization relationship is the one where one model element (child) is based on another model (the parent). We can add the generalization relationships to capture attributes, operations and relationships.
4. Composition association relationships: It represents a whole-part relationship, and it is a form of aggregation.

Steps to make the design model:

It is an object model which describes the realization of use cases, and serves as an abstraction of the implementation model and its source code. Unlike a class diagram, design model contains actual classes of the system (which will be implemented).

1. Referring to the actual class diagram, we can get a basic idea of the model.
2. As per actual code implementation, additional classes are created, for e.g. databases access classes, classes for OOPS concepts, etc.
3. The final design model is created which mimics the actual software classes, which are needed to be created while actual coding.

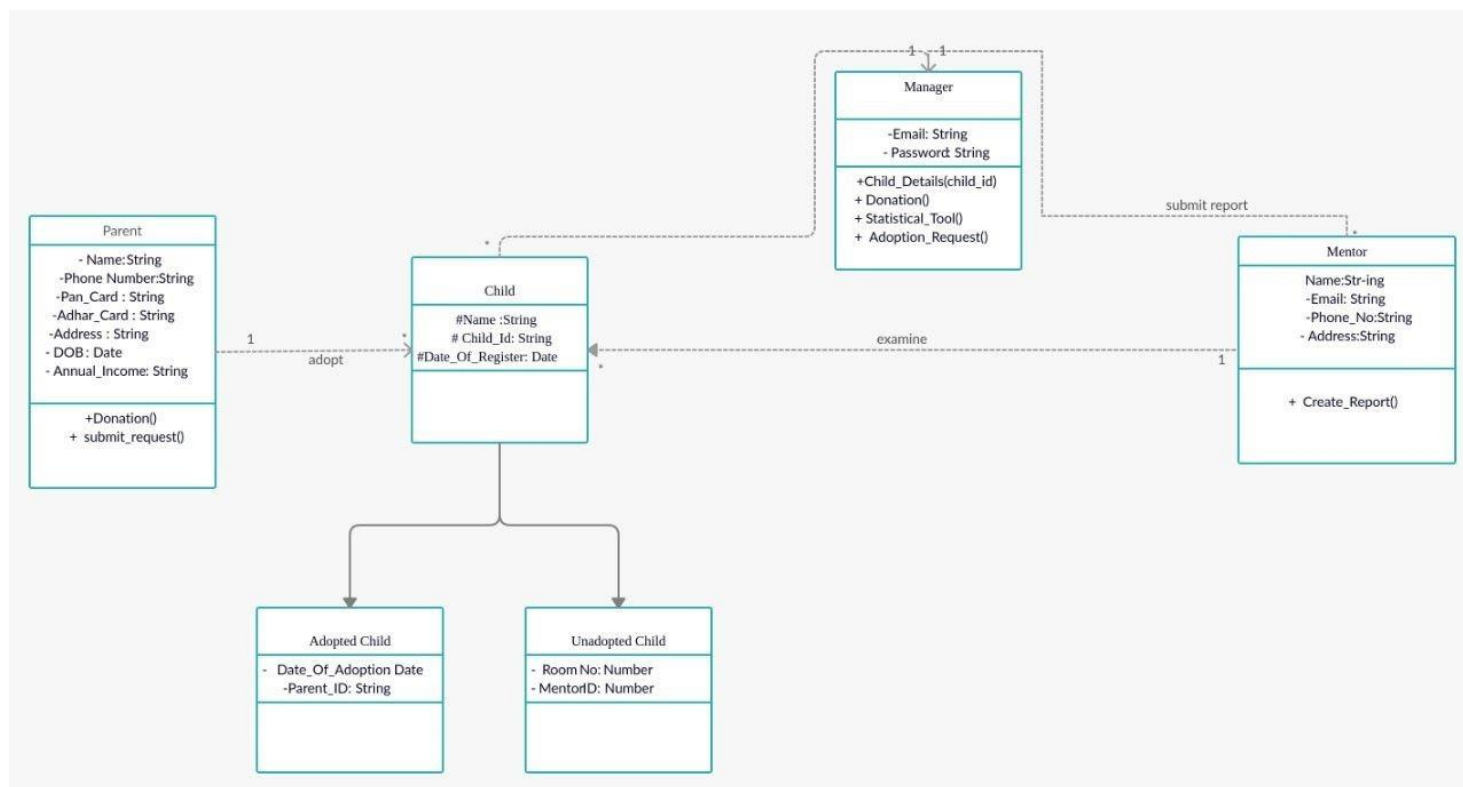
Analysis:

Nouns: Software, Orphanage Management System, Admin, orphan, room, attendance, payment, late, Mentor, account, registration ID, password, Aadhaar number, doctor, parents, donor, school,

Verbs: view, login, access, verification, allot, see, create, add, update, delete, logout, set an appointment, donate

Classes: Child, Mentor, Admin, Adopted Child, Unadopted Child

Design model from the analysis model:



Conclusion :

We understood how to identify design level classes in a project. We have thus implemented accepted standards and procedures to develop a Design model from analysis model for Orphanage Management System.