# ****Practical: Modeling UML Class Diagrams (Structural & Behavioral Aspects)****

### ****Objective:****

To learn how to model the structure and behavior of a system using UML class diagrams. The project domain is **Brain Disease Prediction**, which uses machine learning techniques to diagnose brain-related diseases.

## ****Project Overview: Brain Disease Prediction System****

This system collects patient data (e.g., symptoms, MRI images), preprocesses it, runs predictions using machine learning models, and provides diagnostic reports to users like doctors and researchers.

## ****UML Class Diagrams: Definition****

A **UML Class Diagram** is a **static (structural) view** of a system, representing its **classes**, **attributes**, **methods**, and the **relationships** between them. It may also reflect **behavioral aspects** through operations (methods) and associations.

## ****Elements of Class Diagram****

| **Element** | **Description** |
| --- | --- |
| **Class** | Blueprint for objects; includes attributes (fields) and operations (methods) |
| **Attributes** | Represent data (e.g., name, age, MRI image) |
| **Methods** | Actions/functions (e.g., uploadImage(), predictDisease()) |
| **Relationships** | Associations, generalization, aggregation, composition |

## ****Identifying Classes and Relationships****

### Main Classes:

| **Class** | **Key Attributes** | **Key Methods** |
| --- | --- | --- |
| **Patient** | patientID, name, age, symptoms, image | uploadData(), viewReport() |
| **Doctor** | doctorID, name, specialization | login(), analyzeReport() |
| **MLModel** | modelType, accuracy, parameters | trainModel(), predictDisease() |
| **Report** | reportID, result, dateGenerated | generateReport(), viewDetails() |
| **Database** | dataType, storageSize | saveData(), retrieveData() |
| **PredictionSystem** | systemID, status | processInput(), displayOutput() |

## ****Relationships in Class Diagram****

| **Relationship** | **Meaning** |
| --- | --- |
| **Association** | A patient uploads data to the system |
| **Aggregation** | A report includes the prediction result (Report → MLModel) |
| **Composition** | A patient has a report (strong life-cycle dependency) |
| **Generalization** | A **User** class can be a base for **Patient** and **Doctor** (inheritance) |

## ****UML Class Diagram: Brain Disease Prediction System****

Here is the generated class diagram:

**Key Highlights from Diagram**:

User is an abstract superclass.

Patient and Doctor inherit from User.

Patient is composed with Report.

PredictionSystem communicates with MLModel.

MLModel is used by Report to generate results.

## ****Structural vs Behavioral Aspects****

| **Aspect** | **UML Representation** | **Example in Diagram** |
| --- | --- | --- |
| **Structural** | Classes, Attributes, Relationships | Patient, Doctor, Report, MLModel |
| **Behavioral** | Methods/Operations | predictDisease(), generateReport(), viewReport() |

## ****Conclusion****

The UML Class Diagram provides a clear representation of both **data structure** and **functionality** in the **Brain Disease Prediction System**. It is helpful in planning the codebase, communication between components, and long-term maintenance of the system.