# ****Practical: Modeling UML Sequence Diagrams****

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

To understand how to model the **dynamic behavior** of a system using **UML Sequence Diagrams**, which capture **interactions between objects over time**.

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

The system allows a **doctor** or **patient** to input medical data, which is processed by an **ML model**. The system then generates and displays a **diagnostic report**.

## ****What is a Sequence Diagram?****

A **UML Sequence Diagram** models the **flow of messages between objects** over **time**. It shows how objects interact in a particular **scenario of a use case**.

## ****Elements of a Sequence Diagram****

| **Element** | **Description** |
| --- | --- |
| **Object (Lifeline)** | An entity (class instance) involved in the interaction (e.g., Patient, System) |
| **Message** | A communication between objects (e.g., request, response) |
| **Activation Bar** | The time duration when an object is performing an action |
| **Lifeline** | A vertical dashed line representing the life span of an object |
| **Return Message** | Dashed line with arrow indicating response |

## ****Identifying Objects****

For the scenario: **"Patient submits data and receives prediction report"**, the key objects are:

Patient

PredictionSystem

MLModel

Report

Database

## ****Sequence Diagram for Prediction Flow****

Below is a diagram showing interactions between the system components when a patient submits data and receives a prediction.

### ****Diagram: Brain Disease Prediction Sequence Diagram****

### ****Step-by-Step Message Flow:****

| **Step** | **Interaction Description** |
| --- | --- |
| 1 | Patient sends medical data to PredictionSystem via submitData() |
| 2 | PredictionSystem stores data in Database |
| 3 | PredictionSystem sends data to MLModel for analysis |
| 4 | MLModel returns predicted result |
| 5 | PredictionSystem generates report using Report class |
| 6 | Report is saved to Database |
| 7 | Patient receives the result using viewReport() |

## ****Use of Sequence Diagrams****

**Analyzes dynamic behavior** of the system.

Identifies the order of interactions.

Helps in validating use cases and refining class designs.

## ****Conclusion****

The **Sequence Diagram** clearly models how various system components interact in the **Brain Disease Prediction** workflow. It is essential in design stages to understand runtime object communication.

Would you like a **collaboration diagram** or **activity diagram** next for behavioral modeling?