**Key Topics:**

1. **Recap: UML Class Diagrams**
2. **Dynamic Modelling: UML Sequence Diagrams**
3. **Components of Sequence Diagrams**
4. **Creating Sequence Diagrams**
5. **Complex Situations and When to Use Sequence Diagrams**

**1. Recap: UML Class Diagrams 100%**

**Key Points:**

* **Purpose**: Describe the static structure of the system, including classes, attributes, and associations.
* **Classes**: Represent features and facts about the problem domain relevant to the system.
* **Attributes and Associations**: Detail class properties and relationships, including roles and multiplicity.

**2. Actors, Objects, and Classes**

**Definitions**:

* **Actor**: An entity outside the system interacting with it (e.g., a "Driver" interacting with a car system).
* **Class**: An abstraction modelling an entity within the problem domain (e.g., "Car" within the system).
* **Object**: A specific instance of a class (e.g., "my white Toyota Corolla hybrid").

**Example**:

* **Actor**: Customer
* **Class**: Booking System
* **Object**: Specific booking instance

**3. Class-Responsibility-Collaborator (CRC) Model**

**Method for Sequence Diagrams drawing**

**Components**:

* **Class Name**: Defines the object’s vocabulary for design discussion.
* **Responsibilities**: Identify the tasks or problems the object is responsible for solving, described with short verb phrases containing active verbs.
* **Collaborators**: Other objects that the primary object interacts with to fulfill its responsibilities.

**CRC Cards**:

* Used as a brainstorming tool for Object-Oriented (OO) design, particularly in Extreme Programming (XP).

**Example Scenario: Cinema Booking System**:

* **Alice** selects the movie "Star Wars" at the Apollo Theatre.
* **Alice** requests the 8 pm showing on 24 March 2020.
* The **booking system** sends the request to Apollo Theatre.
* **Alice** is allocated seat R18.

**CRC Breakdown**:

* **Customer**: Selects theatre, movie, and show date and time.
  + *Collaborators*: Show, Theatre
* **Show**: Receives ticket requests.
  + *Collaborators*: Customer, Theatre
* **Theatre**: Receives booking requests.
  + *Collaborators*: Customer, Show

**4. Dynamic Modelling: UML Sequence Diagrams**

**Definition**:

* Sequence diagrams represent the interaction of objects (horizontally) over time (vertically), using messages to communicate.
* Typically capture the behaviour of a single scenario, showing the messages passed between objects within a use case.

**Purpose**:

* Understand how objects collaborate over time.
* Tie use cases to objects, revealing the flow of events.

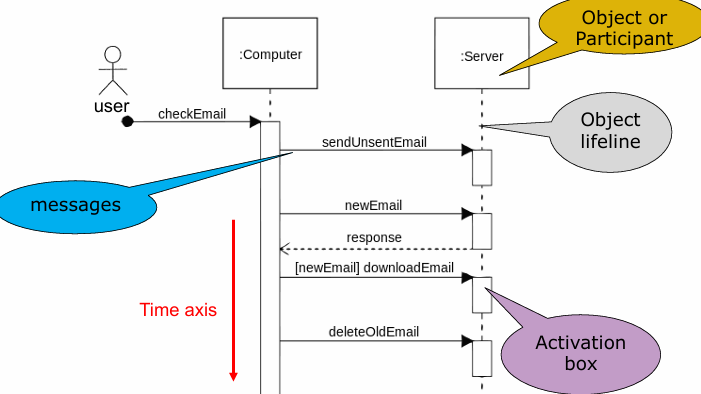
**Case Study: Cinema Booking System**

* The system stores seat bookings for multiple theatres, where each theatre has rows of seats.
* Customers can reserve seats and are given a row number and a seat number, possibly for several adjoining seats.
* Each booking is for a particular show at a specified date and time in a designated theatre.
* The system stores the customer's telephone number.

**5. Components of Sequence Diagrams**

**Interaction Diagrams:**

* **Describe how groups of objects collaborate in some behavior.**
* **Components: Participants (objects), lifelines, messages, and the time axis.**
* **interaction of objects (horizontally) over time (vertically).**

****

**Example: User Checks Email**

* **Participants: User, Email Client, Server**
* **Messages: Check email, Authenticate, Retrieve emails**

**Conventions:**

* **Layout:**
  + **1st column: Actor initiating the use case.**
  + **2nd column: Boundary object the actor interacts with.**
  + **3rd column: Control object managing the use case.**
  + **4th column: Entity object representing long-lived information.**

**Example: Order Pricing Scenario**

* **Order invokes a command to calculate its price by looking at line items and determining their prices based on product pricing rules.**
* **Order computes an overall discount based on customer-specific rules.**

**6. Creating Sequence Diagrams**

**Steps:**

1. **Identify Participants: Determine objects associated with the use case.**
2. **Identify Events: Define the messages sent between objects.**

**Worked Example: Create New Online Library Account**

1. **Librarian requests system to create a new account.**
2. **Librarian selects account type.**
3. **Librarian enters user details.**
4. **System checks details against the Credentials Database.**
5. **New account is created.**
6. **Account details are emailed to the user.**

**Sequence Diagram**:

* Participants: Librarian, System, Credentials Database, Email System
* Messages: Create account request, Select account type, Enter user details, Check details, Create account, Email account details

**5. Complex Situations and When to Use Sequence Diagrams**

**Usage**:

* Sequence diagrams are effective for understanding collaborations among objects in a single use case.
* Not suitable for precise behaviour definition.
* Useful for resolving ambiguity and uncertainty in complex scenarios.

**Example: More Complex Situations**

* Use sequence diagrams to clarify interactions and collaborations in scenarios with multiple alternative interactions.

**When Not to Use**:

* For quickly exploring multiple alternative interactions, CRC cards might be more efficient.

**Detailed Example of a Sequence Diagram**

**Example: Create New Online Library Account**

**Step 1: Identify Participants**

1. Librarian
2. System
3. Credentials Database
4. Email System

**Step 2: Identify Events**

1. Librarian requests to create a new account.
2. Librarian selects account type.
3. Librarian enters user details.
4. System checks details against the Credentials Database.
5. New account is created.
6. Account details are emailed to the user.