**NATIONAL UNIVERSITY OF COMPUTER AND**

**EMERGING SCIENCES**

**SL-2002 – Software Design & Architecture Lab**

**Lab Instructor: Zarnain Maryam Awan**

**Lab 07**

**Sequence Diagram:**

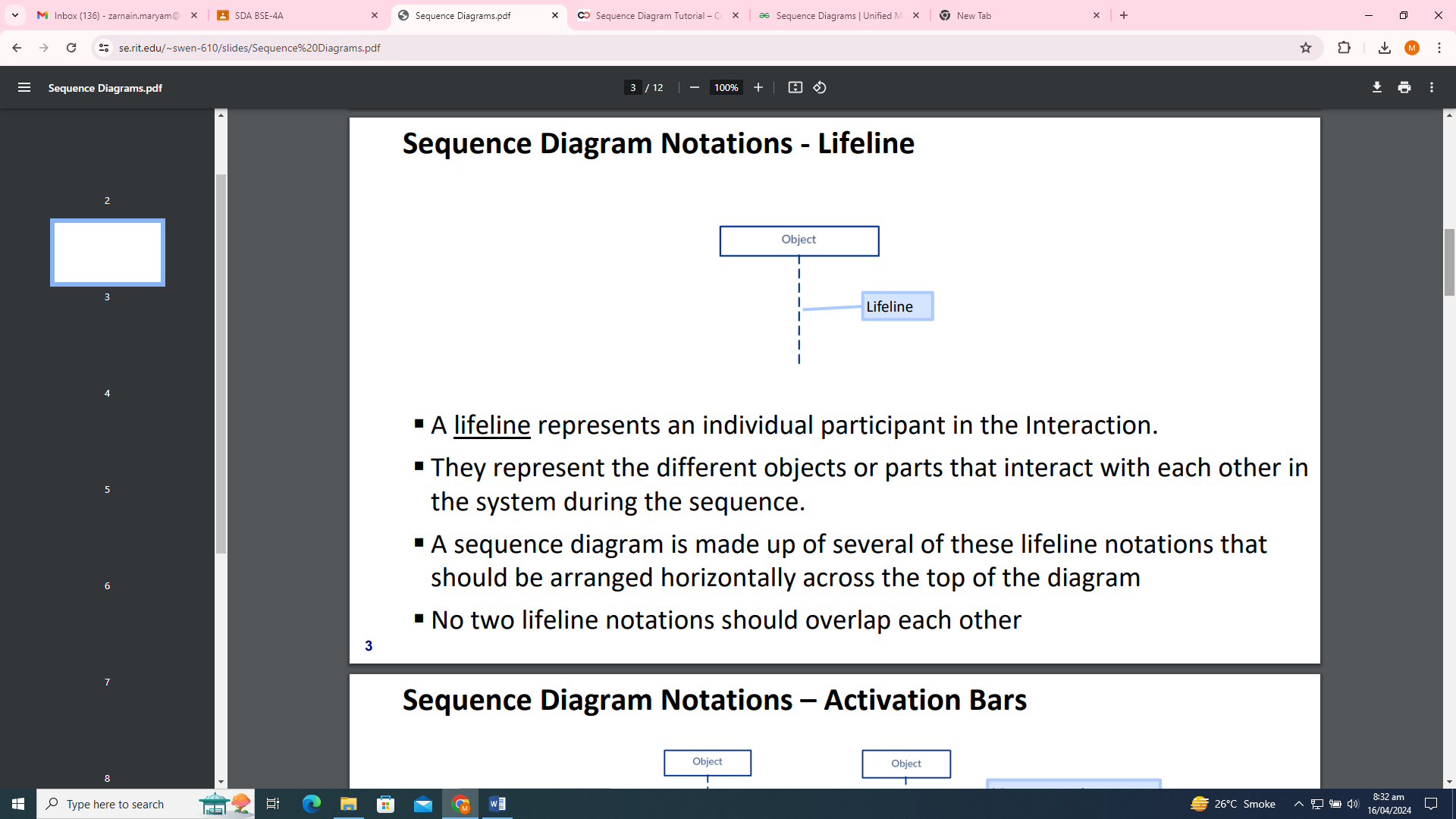
A sequence diagram is a type of interaction diagram in UML (Unified Modeling Language) used to visualize the interactions between objects or participants over time within a system. It illustrates how objects or components interact in a particular scenario or use case by showing the sequence of messages exchanged between them.

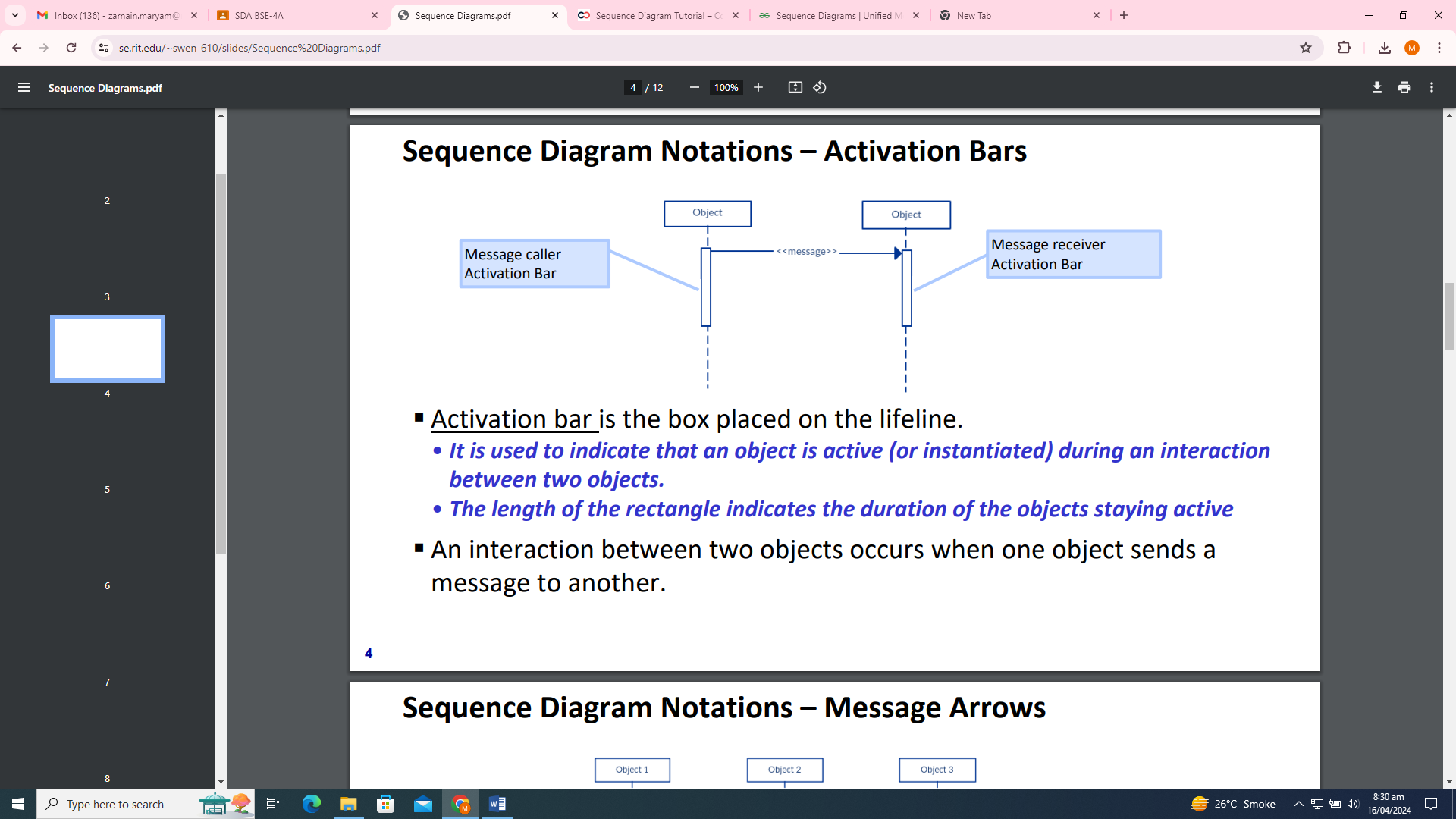
In a sequence diagram:

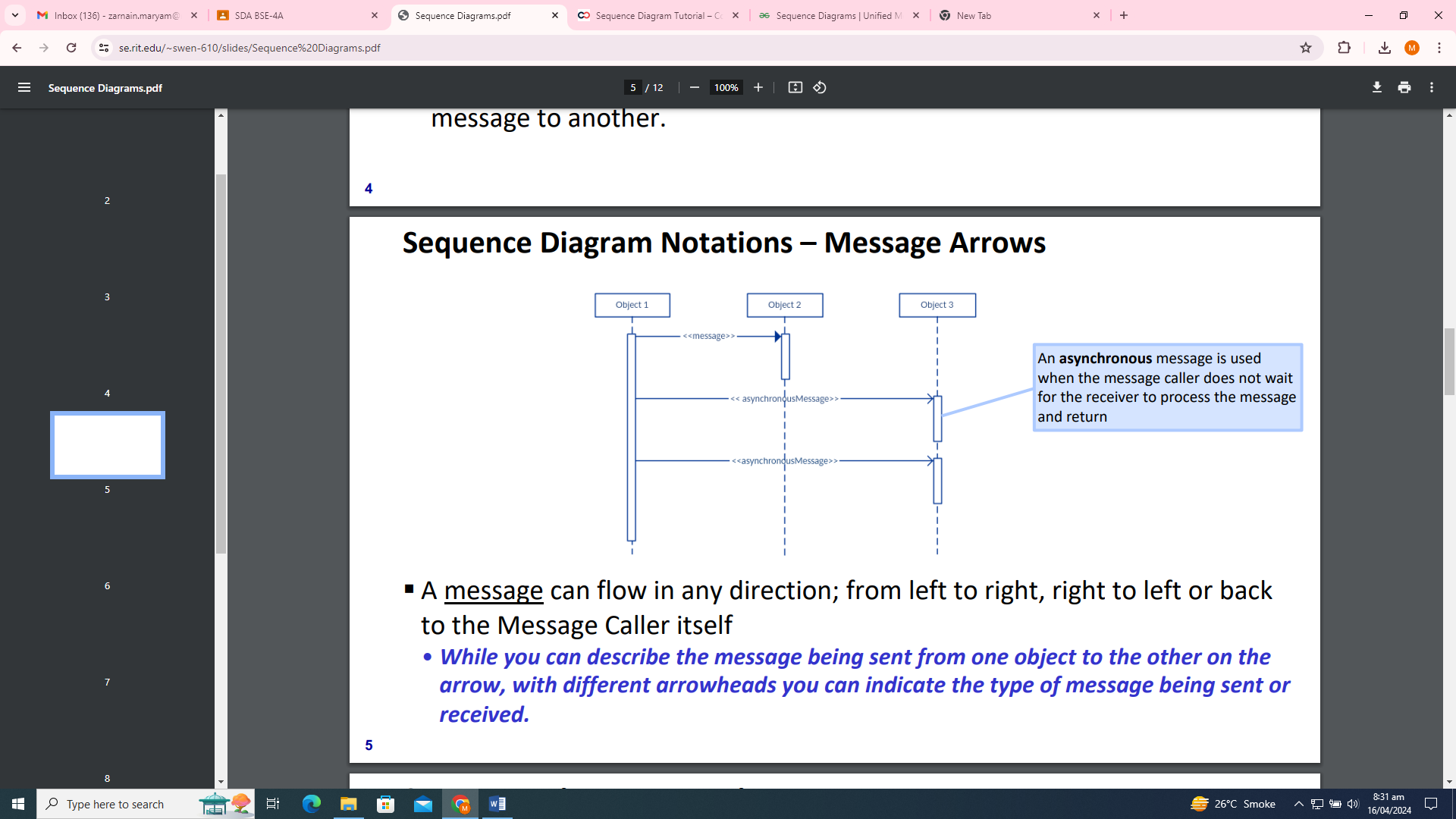
1. **Participants/Objects**: Each participant or object involved in the interaction is represented as a vertical line called a lifeline.
2. **Messages**: Messages exchanged between participants are represented by arrows or lines connecting the lifelines. These messages can indicate method calls, signals, or other types of communication.
3. **Timeline**: The vertical dimension of the diagram represents time, with interactions occurring from top to bottom along the lifelines.
4. **Sequence Numbers**: Messages are typically numbered to indicate the order in which they occur, helping to visualize the chronological sequence of interactions.
5. **Optional Elements**: Sequence diagrams may include optional elements such as activation bars, which represent the period of time during which an object is performing an action.

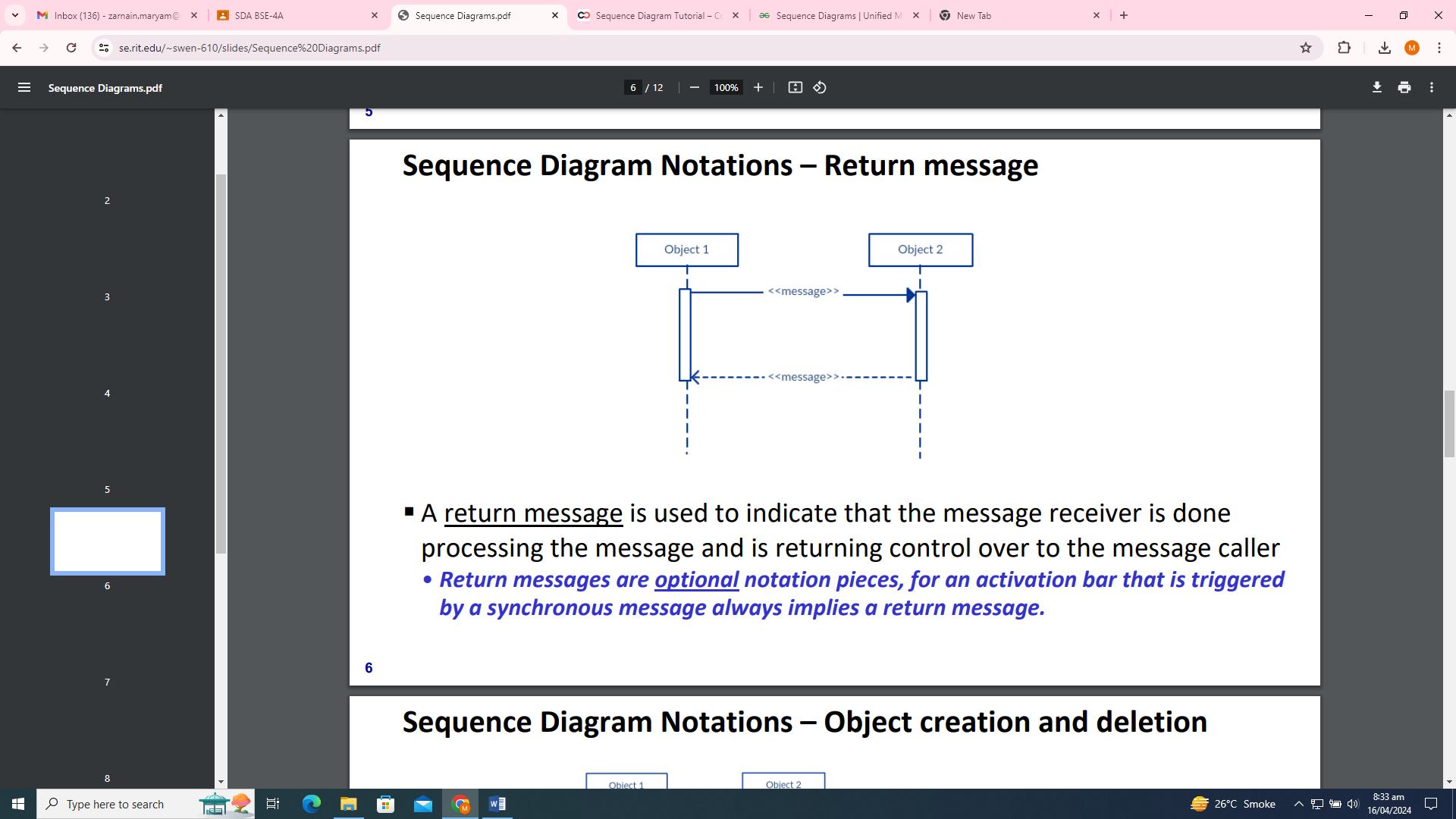
Sequence diagrams are valuable for understanding the flow of interactions in a system, identifying dependencies between objects, and communicating the behavior of a system to stakeholders, designers, and developers. They are commonly used during the design and documentation phases of software development projects.

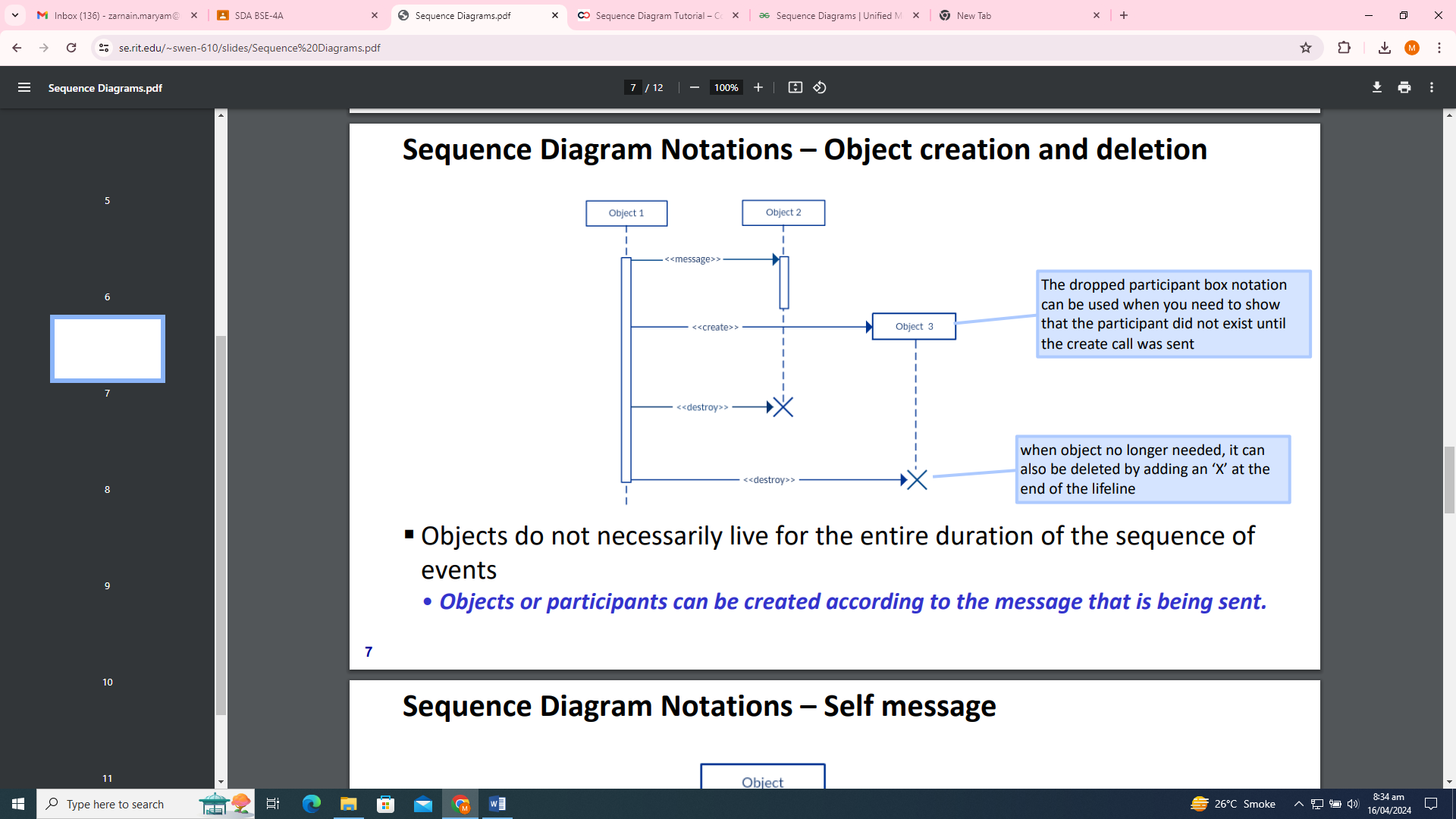
**Sequence Diagram Symbols:**

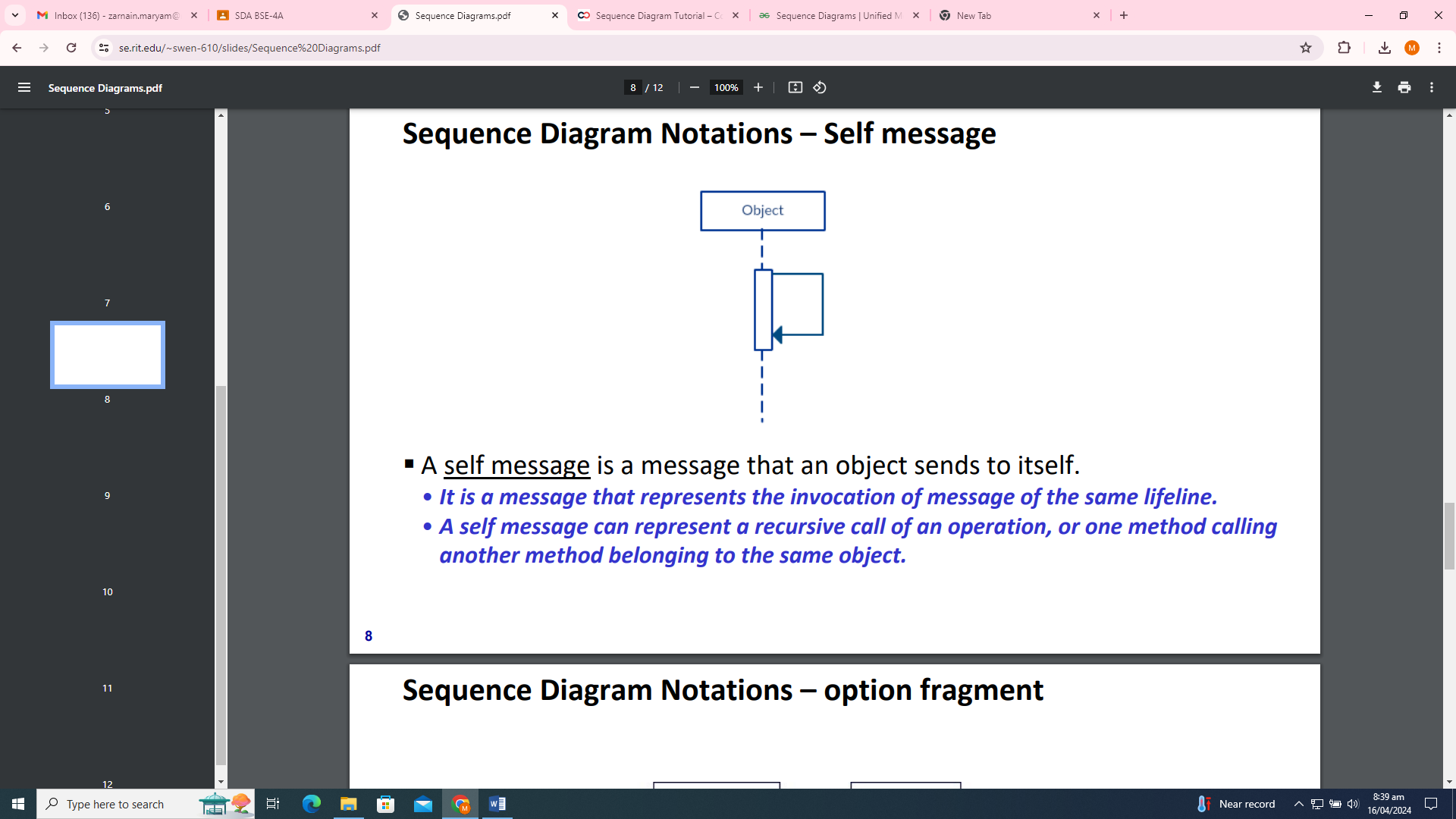


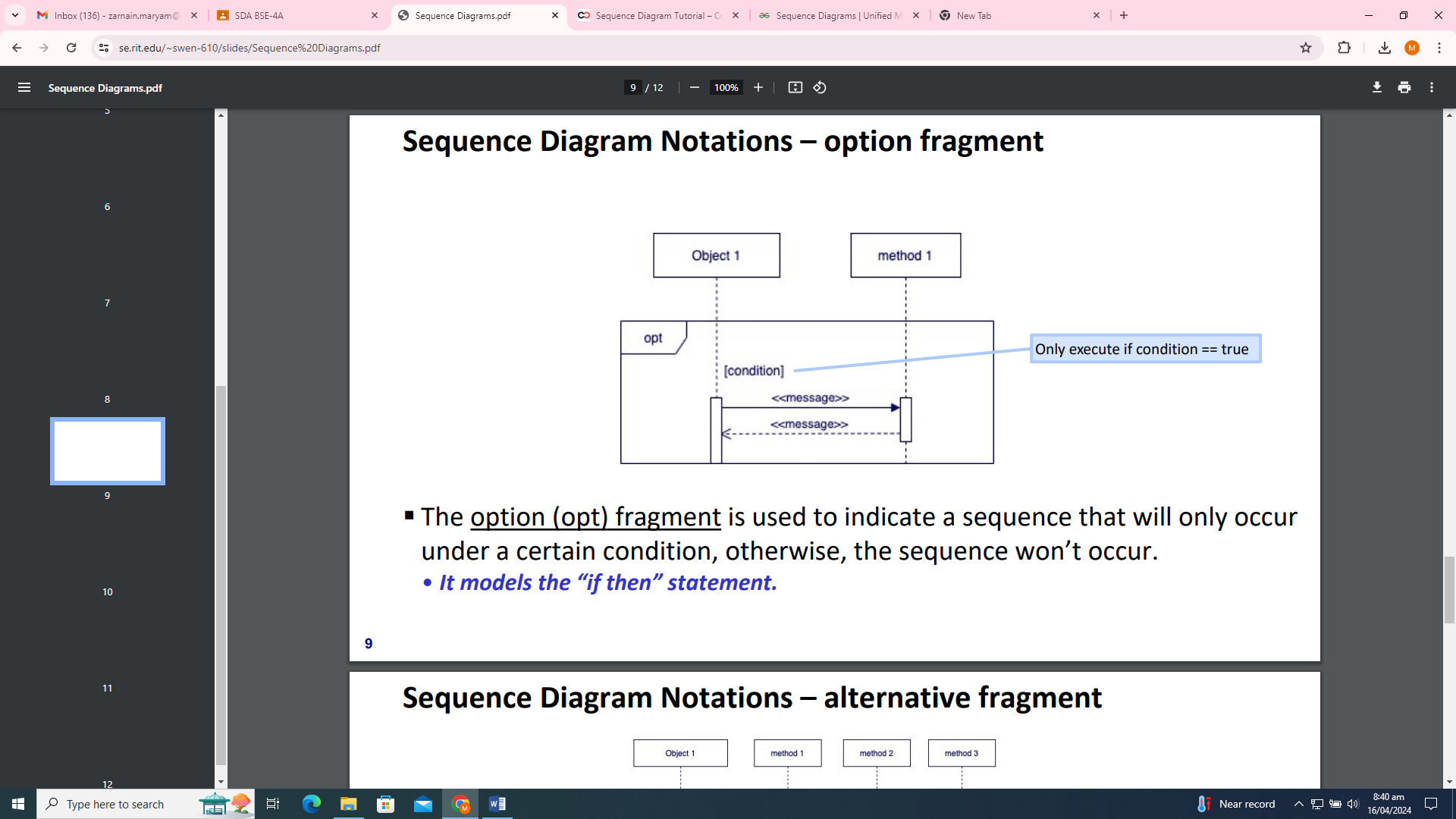


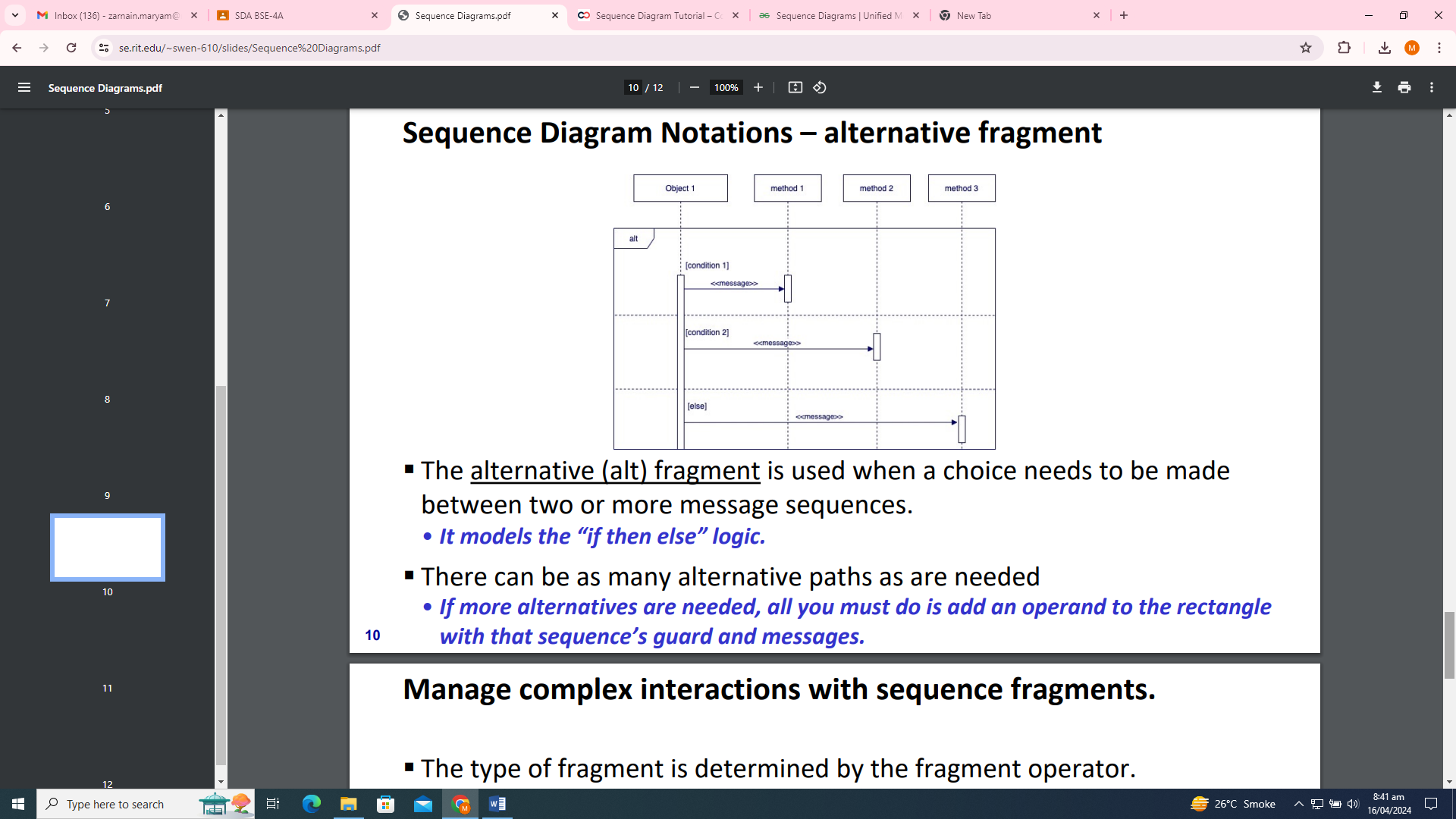


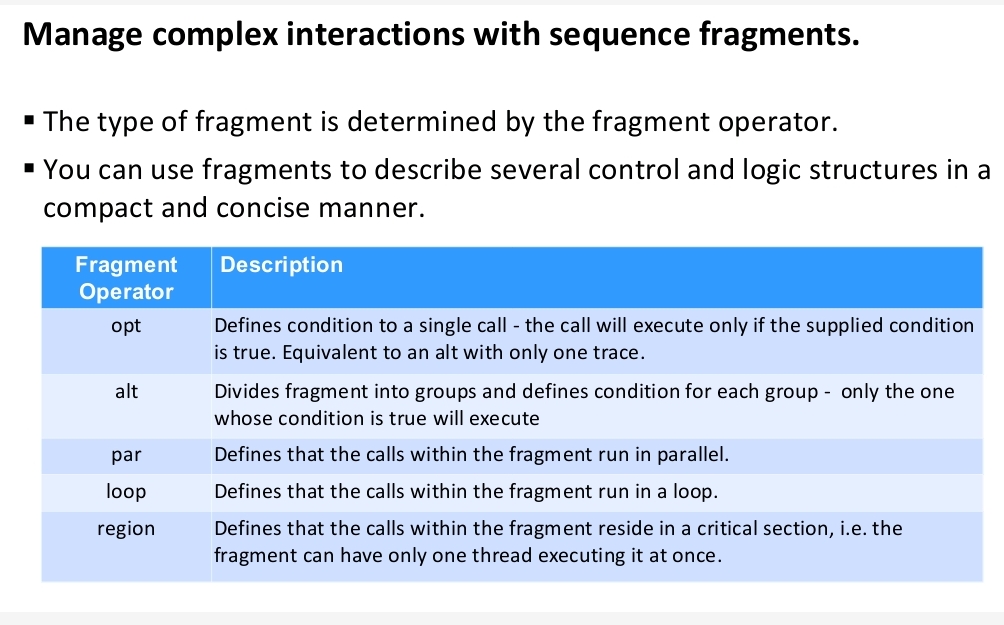


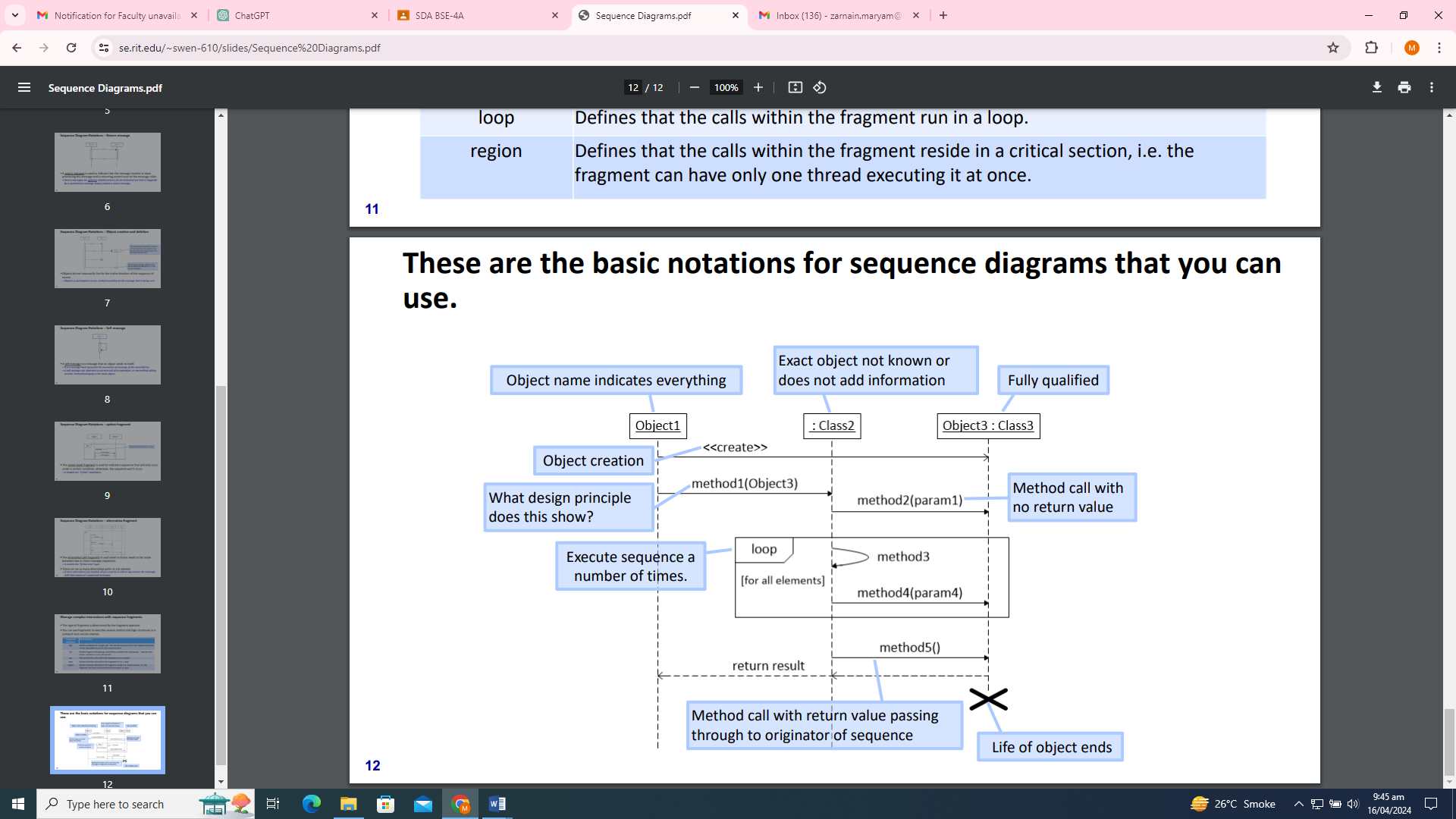








****



**Steps for Creating a Sequence Diagram:**

Creating a sequence diagram involves several steps, but it's generally straightforward. Here's a step-by-step guide:

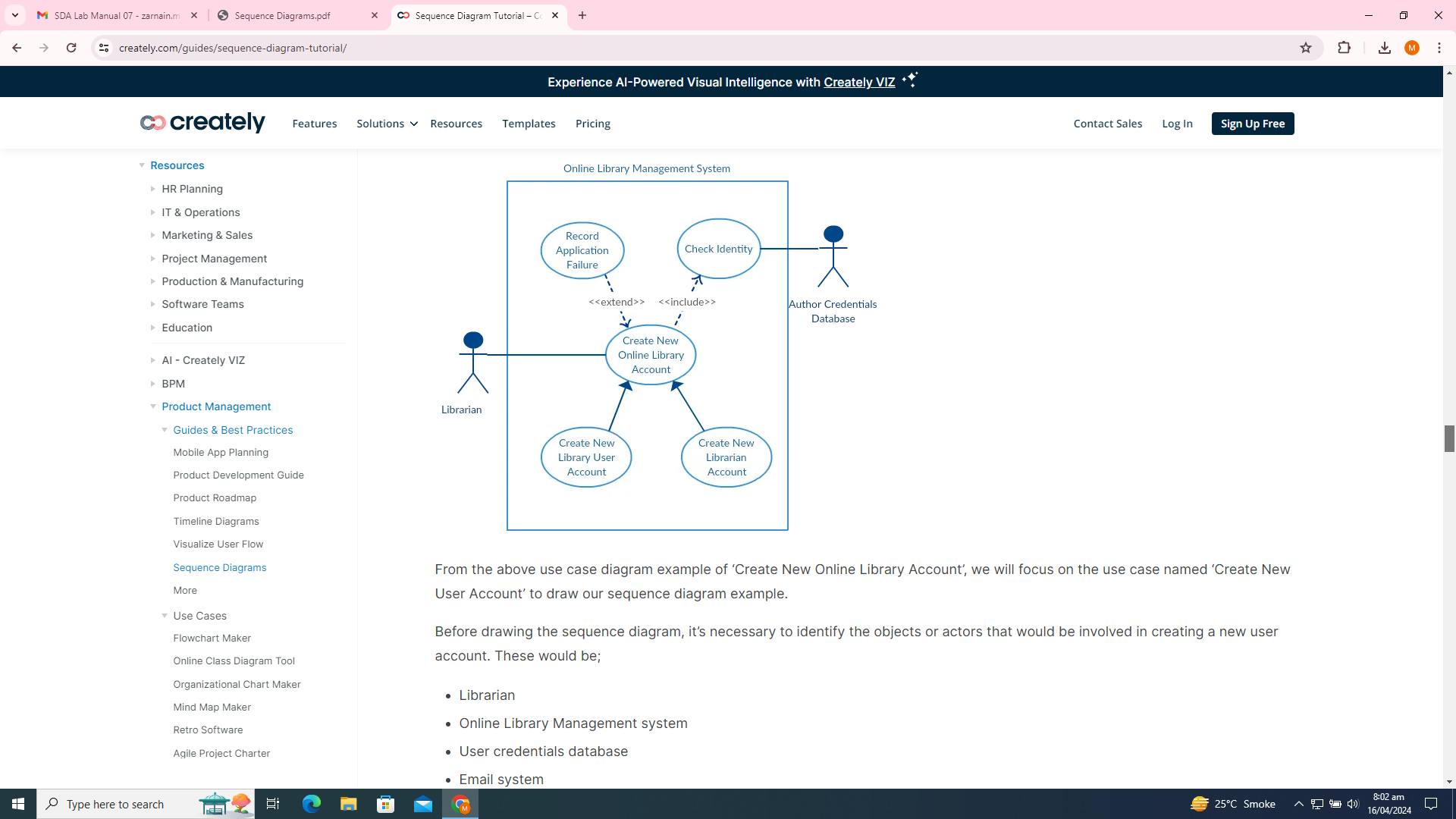
1. **Identify Participants/Objects:** Determine the entities (actors, objects, or components) involved in the sequence of interactions. These could be users, systems, or external entities.
2. **Define the Use Case/Scenario:** Understand the specific scenario or use case you want to represent in the sequence diagram. What actions are being performed? What interactions between participants are occurring?
3. **Draft the Timeline:** Sketch out a rough timeline of the interactions between the participants. This timeline will form the backbone of your sequence diagram.
4. **Identify Messages/Interactions:** Determine the messages passed between the participants during the scenario. Messages typically represent method calls, signals, or communications between objects.
5. **Drawing the Diagram:** Use a tool or draw the sequence diagram by hand. Represent the participants as vertical lines (also called lifelines) running down the page. Place the messages between the lifelines to indicate communication.
6. **Adding Details:** Add additional details to the diagram as necessary. This could include method parameters, return values, or conditions.
7. **Review and Refinement:** Review the diagram to ensure it accurately represents the intended scenario. Make adjustments as needed for clarity and accuracy.
8. **Finalize and Share:** Once satisfied with the diagram, finalize it and share it with stakeholders or team members as needed.
9. **Documentation:** Optionally, document the sequence diagram, providing explanations for each step and clarifying any ambiguities.

Following these steps should help you create a clear and understandable sequence diagram for your scenario.

**How to Draw a Sequence Diagram**

A sequence diagram represents the scenario or flow of events in one single use case. The message flow of the sequence diagram is based on the narrative of the particular use case.

Then, before you start drawing the sequence diagram or decide what interactions should be included in it, you need to draw the use case diagram and ready a comprehensive description of what the particular use case does.



From the above use case diagram example of ‘Create New Online Library Account’, we will focus on the use case named ‘Create New User Account’ to draw our sequence diagram example.

Before drawing the sequence diagram, it’s necessary to identify the objects or actors that would be involved in creating a new user account. These would be;

* Librarian
* Online Library Management system
* User credentials database
* Email system

Once you identify the objects, it is then important to write a detailed description of what the use case does. From this description, you can easily figure out the interactions (that should go in the sequence diagram) that would occur between the objects above, once the use case is executed.

Here are the steps that occur in the use case named ‘Create New Library User Account’.

* The librarian requests the system to create a new online library account
* The librarian then selects the library user account type
* The librarian enters the user’s details
* The user’s details are checked by the system using the user Credentials Database
* The new library user account is created by the system
* A summary of the new account’s details is then emailed to the user by the system

From each of these steps, you can easily specify what messages should be exchanged between the objects in the sequence diagram. Once it’s clear, you can go ahead and start drawing the sequence diagram.

The sequence diagram below shows how the objects in the online library management system interact with each other to perform the function ‘Create New Library User Account’.

