**OOP Messaging App System**

**Theory Assignment-1**

**Prepared by:**  
[Sara Riaz]

**Student ID:**  
[SP24-BSE-110-B]

**Course Name:**  
[Object Oriented Programming]

**Submitted TO:**  
[Muhammad Shahid Bhatti]

**Submission Date:**  
[10/11/2024]

**Institution:**  
[Comsats University Islamabad, Lahore Campus]

### Introduction

* This document provides an overview of the OOP Messaging App System, detailing its classes, methods, attributes, and functionalities.

### Class: Message

* **Purpose**: Represents an individual message in the messaging system.

**Attributes:**

* static int idCounter: Static counter for unique message IDs.
* int messageId: Unique ID for each message.
* String sender: Sender of the message.
* String receiver: Receiver of the message.
* String content: Content of the message.
* LocalDateTime timestamp: Timestamp when the message is created.
* boolean seen: Status of the message (seen or unseen).
* boolean isDeleted: Status of the message (deleted or not).

**Methods:**

* **Constructor**: Initializes a new message with sender, receiver, and content.
* **Getters**: Provides access to message attributes.
* **markAsSeen()**: Marks the message as seen.
* **markAsDeleted()**: Marks the message as deleted.
* **toString()**: Returns a string representation of the message.

### Class: MessagingApp

* **Purpose**: Manages the messaging system, including sending, deleting, searching, and reading messages.

**Attributes:**

* Message[][][] messages: 3D array to store messages between senders and receivers.
* int[][] messageCount: 2D array to track the number of messages between sender-receiver pairs.

**Methods:**

* **sendMessage(String sender, String receiver, String content)**: Sends a message and increments the message count.
* **deleteMessage(int id)**: Marks a message as deleted based on its ID.
* **searchMessageById(int id)**: Searches for a message by its ID and prints it if found.
* **readUnseenMessages(String receiver)**: Reads and marks unseen messages for a specific receiver.
* **readAllMessages(String receiver)**: Reads all messages for a specific receiver.
* **readMessagesFromSender(String sender)**: Reads all messages sent by a specific sender.
* **getUserIndex(String username)**: Returns the index of the specified user in the messaging system.

### Encapsulation

* Attributes of classes are kept private to protect data integrity.
* Public methods (getters and setters) provide controlled access to class attributes, ensuring that the internal state is not directly manipulated.

### Composition

* The MessagingApp class is composed of multiple Message objects, establishing a relationship where messages are integral to the functionality of the messaging system.

### Class: MainApp

* Making object of messaging app class
* Using do while loop
* Implement menu

Example OUTPUT:

**Menu Output**

Messaging App Menu:  
1. Send a message  
2. Delete a message  
3. Search message by ID  
4. Read unseen messages  
5. Read all messages of a receiver  
6. Read all messages from a sender  
7. Exit  
Enter your choice:

1.Send a Message :

OUTPUT:

Enter your choice: 4

Enter receiver name to read unseen messages: Jack

From: Sara To: Jack Message: Hello WOrld! (ID: 14) Sent at: 2024-10-11T11:44:23.832211200

2.Delete a Message :

OUTPUT:

Enter your choice: 2

Enter message ID to delete: 14

Message with ID 14 deleted.

3.Search a Message :

OUTPUT:

Enter your choice: 3

Enter message ID to search: 14

Message deleted

4.Read unseen Messages :

(If I did not delete that message and then read unseen message of recever jack then output will be:)

OUTPUT:

Enter your choice: 4

Enter receiver name to read unseen messages: Jack

From: Sara To: Jack Message: Hello WOrld! (ID: 14) Sent at: 2024-10-11T11:44:23.832211200

5. Read all messages of a receiver:

OUTPUT:

Enter your choice: 5

Enter receiver name to read all messages: Jack

From: Sara To: Jack Message: Hi Jack! I'm good. You? (ID: 5) Sent at: 2024-10-11T11:56:36.209798300

From: Sara To: Jack Message: Yes, I am! What do you have in mind? (ID: 7) Sent at: 2024-10-11T11:56:36.209798300

6. Read all messages from a sender:

OUTPUT:

Enter your choice: 6

Enter sender name to read all messages: Sara

From: Sara To: Jack Message: Hi Jack! I'm good. You? (ID: 5) Sent at: 2024-10-11T11:56:36.209798300

From: Sara To: Jack Message: Yes, I am! What do you have in mind? (ID: 7) Sent at: 2024-10-11T11:56:36.209798300