****

**Assignment 1**

**Title: *Messaging App***

|  |  |
| --- | --- |
| Submitted by: | Aleesha Batool |
| Registration No: | SP24-BSE-131-B |
| Course Title: | Object Oriented Programming |
| Submitted To: | Sir M. Shahid Bhatti |
| Date: | 11-October-2024 |

**1. Class: MessagingApp**

**Attributes:**

* **Receiver[][] receivers**:
  + A 2D array of Receiver objects to store messages for different receivers.
* **int receiversCounter**:
  + Keeps track of how many unique receivers are stored in the system.
* **Block**:
  + Initializes the receivers array with some sample Receiver objects and their associated messages for testing purposes.

**Methods:**

* **sendMessage()**:
  + Takes user input for the receiver's name, phone number, and message content.
  + If the receiver already exists in the system, the new message is added to that receiver's record.
  + If the receiver does not exist, a new Receiver object is created, and the message is associated with that receiver.
* **readAllMessages()**:
  + Retrieves and displays all messages from all receivers.
  + Marks all messages as "seen" and sorts them by date and time using the sortMessages() method.
* **readReceiverMessages()**:
  + Prompts the user for the receiver's name and phone number.
  + Displays all messages associated with the receiver and marks them as "seen."
* **deleteMessages()**:
  + Allows the user to delete a specific message by providing the receiver's name, phone number, and message ID.
  + Deletes the message if the provided ID matches.
* **searchMessage()**:
  + Prompts the user for a message ID.
  + Searches through all stored messages to find the one with the matching ID and displays it if found.
* **displayContacts()**:
  + Displays the names and phone numbers of all receivers stored in the system.
* **sortMessages()**:
  + Collects all messages from the receivers array and sorts them by date in descending order.
  + Displays the sorted messages.

**2. Class: Message**

**Attributes:**

* **int messageId**:
  + A unique identifier for each message.
* **String content**:
  + The text content of the message.
* **String dateTime**:
  + The date and time the message was created.
* **String status**:
  + The status of the message (e.g., "unseen", "seen").

**Methods:**

* **toString()**:
  + Returns a formatted string containing the message details, including the ID, content, status, and date/time.

**3. Class: Receiver**

**Attributes:**

* **String name**:
  + The name of the receiver.
* **String number**:
  + The phone number of the receiver.
* **Message message**:
  + The message associated with the receiver.

**Methods:**

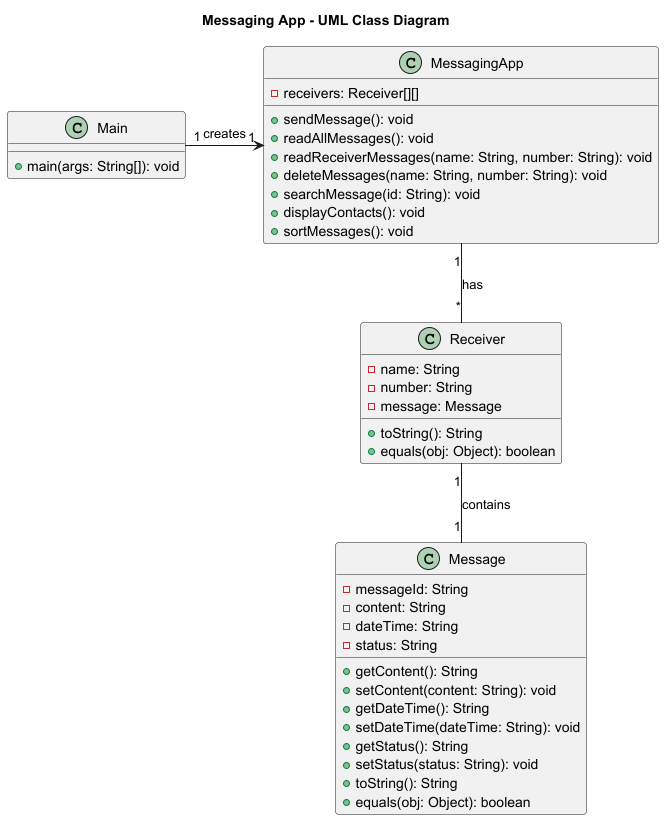
* **Receiver(String name, String number, Message message)**:
  + A constructor that initializes a Receiver object with the receiver's name, phone number, and message.
* **getName()**:
  + Returns the name of the receiver.
* **getNumber()**:
  + Returns the phone number of the receiver.
* **getMessage()**:
  + Returns the Message object associated with the receiver.
* **toString()**:
  + Returns a formatted string containing the receiver's details, including the name, phone number, and message.
* **equals(Object obj)**:
  + Compares two Receiver objects based on their name and phone number to determine if they are equal.

**4. Class: Main**

**Method:**

* **main(String[] args)**:
  + The entry point of the program.
  + Creates an instance of the MessagingApp class.
  + Displays a menu that allows the user to select various actions such as sending messages, reading messages, deleting messages, searching for messages, and displaying contacts.
  + Continuously loops to accept user input and perform the corresponding actions

**UML Diagram**:



**Menu Screen:** 