Java Code Documentation

# Introduction

This document provides an overview and explanation of the Java program that implements a basic messaging system. The application supports features like sending messages, viewing message history, adding and deleting contacts, and managing message statuses. Below is a breakdown of the program's structure and the functionality of the different classes and methods.

# DefaultMessages.java

The `DefaultMessages` class generates random default messages for the system. This class contains the method `originalMessages` that takes a `MsgSystem` object and populates it with random messages. The random messages are created using the Random class to determine the status of each message.

# Demo.java

The `Demo` class contains the main method that drives the application. It initializes the message system (`MsgSystem`), generates default messages using `DefaultMessages`, and provides a menu for the user to interact with the system.

## Main Menu Options

The following options are available in the menu:

1. \*\*View Contact List\*\*: Displays the list of contacts.

A screen shot of a computer

Description automatically generated

1. \*\*Send Message\*\*: Allows the user to send a message to a specified contact.

A screenshot of a computer

Description automatically generated

1. \*\*View Messages\*\*: Displays all the messages sent to a specific contact.



1. \*\*View Status-based Messages\*\*: Shows seen/unseen messages.

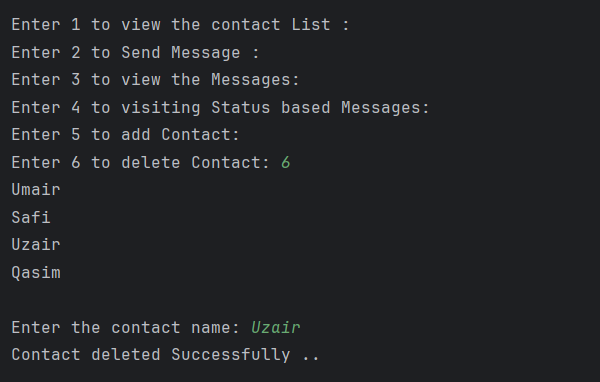


1. \*\*Add Contact\*\*: Allows the user to add a new contact.

A screen shot of a computer

Description automatically generated

6. \*\*Delete Contact\*\*: Allows the user to delete a contact.



# Message.java

The `Message` class represents an individual message in the system. Each message has a receiver, content, unique message ID, status (seen/unseen), and a timestamp. This class also provides getters and setters for message properties.

# MsgSystem.java

The `MsgSystem` class handles the core functionalities of the messaging system. It manages the contact list, sends messages, and handles the logic for displaying messages based on their status (seen or unseen).

## Key Methods in MsgSystem

### sendMsg()

The `sendMsg` method is used to send a message to a specific contact. It takes the receiver's name, the message content, and the message status as parameters. The message is stored in the system if the receiver is found in the contact list.

### receiverMessages()

The `receiverMessages` method is responsible for displaying the messages received by a specific contact. It allows the user to view all messages or select a specific receiver's messages. When messages are displayed, their status is updated to 'seen'.

### addContact()

The `addContact` method allows the user to add a new contact to the system. It dynamically resizes the contact list and message arrays to accommodate the new contact.

### deleteContact()

The `deleteContact` method removes a contact from the contact list. It also removes all associated messages of that contact.

# Screenshots of Compiler Output

The following are screenshots from the compiler showing the execution of various functionalities within the program.