```
package main

import (
	"bufio"
	"fmt"
	"math/rand"
	"os"
	"strings"
	"time"
)

type User struct {
	ID       int
	Messages []string
}

var users map[int]*User

func main() {
	users = make(map[int]*User)

	for {
		fmt.Println("1. Send Message between two users")
		fmt.Println("2. Broadcast message to all users")
		fmt.Println("3. View message log of a user")
		fmt.Println("4. Exit")
		fmt.Print("Enter your choice: ")

		scanner := bufio.NewScanner(os.Stdin)
		scanner.Scan()
		choice := strings.TrimSpace(scanner.Text())

		switch choice {
		case "1":
			sendMessage()
		case "2":
			broadcastMessage()
		case "3":
			viewMessageLog()
		case "4":
			fmt.Println("Exiting the application. Goodbye!")
			return
		default:
			fmt.Println("Invalid choice. Please select a valid option.")
		}
```

```
        }
}

func sendMessage() {
        fmt.Print("Enter sender ID: ")
        senderID := getUserID()

        fmt.Print("Enter receiver ID: ")
        receiverID := getUserID()

        fmt.Print("Enter message content: ")
        message := strings.TrimSpace(getInput())
    if message == "" {
                randomFact := getRandomFact()
                users[senderID].Messages = append(users[senderID].Messages,
fmt.Sprintf("To %d (Random Fact): %s", receiverID, randomFact))
        } else {
                users[senderID].Messages = append(users[senderID].Messages,
fmt.Sprintf("To %d: %s", receiverID, message))
                users[receiverID].Messages = append(users[receiverID].Messages,
fmt.Sprintf("From %d: %s", senderID, message))
        }
}

func broadcastMessage() {
        fmt.Print("Enter message content: ")
        message := strings.TrimSpace(getInput())

        for _, user := range users {
                user.Messages = append(user.Messages, fmt.Sprintf("Broadcast:
%s", message))
        }
}

func viewMessageLog() {
        fmt.Print("Enter user ID: ")
        userID := getUserID()

        user, ok := users[userID]
        if !ok {
                fmt.Println("User not found.")
                return
        }

        fmt.Printf("Message log for User %d:\n", userID)
```

```
	for _, msg := range user.Messages {
		fmt.Println(msg)
	}
}

func getUserID() int {
	scanner := bufio.NewScanner(os.Stdin)
	scanner.Scan()
	input := strings.TrimSpace(scanner.Text())
	userID, err := strconv.Atoi(input)
	if err != nil {
		fmt.Println("Invalid user ID. Please enter a valid integer.")
		return getUserID()
	}
	return userID
}

func getInput() string {
	scanner := bufio.NewScanner(os.Stdin)
	scanner.Scan()
	return scanner.Text()
}
func getRandomFact() string {
	facts := []string{
		"cats jump six times their length",
		"bookkeeper word with three consecutive double letters.",
		"purple no English word rhymes with it",
	}
	rand.Seed(time.Now().UnixNano())
	return facts[rand.Intn(len(facts))]
}
```