```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_USERS 100 // Maximum number of users

// Structure for a message in a doubly linked list
typedef struct MessageNode {
char text[256];
char senderId[50];
struct MessageNode* next;
struct MessageNode* prev;
} MessageNode;

// Structure for the chat history (Doubly linked list of messages)
typedef struct ChatHistory {
MessageNode* head; // Pointer to first message
MessageNode* tail; // Pointer to last message
} ChatHistory;

// Structure for a user
typedef struct UserAccount {
char id[50];
char name[50];
ChatHistory chat; // Chat history
} UserAccount;

// HashMap for storing users
UserAccount* users[MAX_USERS];

// Hash function to calculate the index for user storage
int computeHash(char* id) {
int i, hash = 0;
for (i = 0; id[i] != '\0'; i++) {
hash = (hash + id[i]) % MAX_USERS;
}
return hash;
}

// Function to create a new user
void registerUser(char* id, char* name) {
int index = computeHash(id);
UserAccount* newUser = (UserAccount*)malloc(sizeof(UserAccount));
```

```
strcpy(newUser->id, id);
strcpy(newUser->name, name);

newUser->chat.head = newUser->chat.tail = NULL;
users[index] = newUser;
}

// Function to find a user by ID
UserAccount* searchUser(char* id) {
int index = computeHash(id);
return users[index];
}

// Function to create a new message
MessageNode* composeMessage(char* text, char* senderId) {
MessageNode* newMessage = (MessageNode*)malloc(sizeof(MessageNode));
strcpy(newMessage->text, text);
strcpy(newMessage->senderId, senderId);
newMessage->next = NULL;
newMessage->prev = NULL;

return newMessage;
}

// Function to send a message (add to chat history)
void addMessage(UserAccount* user, char* text, char* senderId) {
MessageNode* newMessage = composeMessage(text, senderId);
if (user->chat.tail == NULL) {
user->chat.head = user->chat.tail = newMessage;
} else {
user->chat.tail->next = newMessage;
newMessage->prev = user->chat.tail;
user->chat.tail = newMessage;
}
}

// Function to display all messages in a chat
void displayChatHistory(UserAccount* user) {
MessageNode* current = user->chat.head;

if (current == NULL) {
printf("No messages in chat.\n");
return;
```

```
}
printf("Chat history:\n");

while (current != NULL) {
printf("%s: %s\n", current->senderId, current->text);
current = current->next;
}
}

int main() {
int i;
// Initialize user directory
for (i = 0; i < MAX_USERS; i++) {
users[i] = NULL;
}

int choice, numUsers, numMessages;
char userId[50], userName[50], senderId[50], messageText[256];

// Dynamic input for creating users
printf("Enter number of users to create: ");
scanf("%d", &numUsers);

for (i = 0; i < numUsers; i++) {
printf("\nEnter User ID: ");
scanf("%s", userId);
printf("Enter Username: ");
scanf("%s", userName);

registerUser(userId, userName);
}

// Dynamic input for sending messages
printf("\nEnter number of messages to send: ");
scanf("%d", &numMessages);

for (i = 0; i < numMessages; i++) {
printf("\nEnter Sender ID: ");
scanf("%s", senderId);
printf("Enter Message Content: ");
getchar(); // Clear the newline left in the input buffer
fgets(messageText, sizeof(messageText), stdin);
```

```
messageText[strcspn(messageText, "\n")] = 0;

UserAccount* sender = searchUser(senderId);
if (sender == NULL) {
printf("User not found.\n");
continue;
}

addMessage(sender, messageText, senderId);
}

// Dynamic input for displaying messages
printf("\nEnter User ID to display chat history: ");
scanf("%s", userId);

UserAccount* user = searchUser(userId);
if (user != NULL) {
displayChatHistory(user);
} else {
printf("User not found.\n");
}

return 0;
}
```

```
rtk5@penguin:~/PES/DSA-sem-3/lab$ gcc orange\ problem\ -\ Chat\ messenger.c
rtk5@penguin:~/PES/DSA-sem-3/lab$ ./a.out
Enter number of users to create: 3

Enter User ID: 1
Enter Username: rtk

Enter User ID: 2
Enter Username: rishil

Enter User ID: 3
Enter Username: sam

Enter number of messages to send: 4

Enter Sender ID: 2
Enter Message Content: hi there!!

Enter Sender ID: 3
Enter Message Content: heyyy

Enter Sender ID: 1
Enter Message Content: long time

Enter Sender ID: 2
Enter Message Content: ikr

Enter User ID to display chat history: 1
Chat history:
1: long time
```