EGE UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

ADVANCE OBJECT ORİENTED PROGRAMMING 2023-2024 SPRING TERM PROJECT REPORT

PROJECT TEAM:

ALİ EKREM YAPICI / 05220001018

ÖZGÜR SAY / 05220001004

RAHİM CAN ZARARSIZ / 05220000958

UML Class Diagram

The UML class diagram is as follows:

- Classes and Interfaces:
    - **AddMemberCommand**
    - **RemoveMemberCommand**
    - **Command (Interface)**
    - **Group**
    - **SubGroup**
    - **User**
    - **UserFactory**
    - **UserManager**
    - **FriendIterator**
    - **Iterator (Interface)**
    - **Wall**
    - **Gui**
    - **GroupManager**
    - **GroupComponent (Interface)**

Relationships between classes:

- **AddMemberCommand** and **RemoveMemberCommand** classes implement the **Command** interface.
- **Group** and **SubGroup** classes implement the **GroupComponent** interface.
- The **User** class is created via the **UserFactory** class.
- The **UserManager** class follows the Singleton pattern, having a single instance that manages **User** objects.
- The **FriendIterator** class implements the **Iterator** interface.

- The **GroupManager** class uses the **Command** interface to implement the command pattern.

Design Patterns and Their Usage

1. **Command Pattern**
    - **Purpose**: The Command Pattern is used to encapsulate a request as an object, thereby allowing for parameterization of clients with queues, requests, and operations.
    - **Usage**:
        - **AddMemberCommand** and **RemoveMemberCommand**: These classes implement the **Command** interface and encapsulate the actions of adding and removing a member from a group, respectively. The commands can be executed and undone.
        - **GroupManager**: This class acts as the invoker, maintaining a reference to a **Command** object and invoking its **execute()** and **undo()** methods.
2. **Iterator Pattern**
    - **Purpose**: The Iterator Pattern provides a way to access the elements of an aggregate object sequentially without exposing its underlying representation.
    - **Usage**:
        - **FriendIterator**: This class implements the **Iterator** interface and provides a way to iterate over a list of friends (**List<User>**).
3. **Singleton Pattern**
    - **Purpose**: The Singleton Pattern ensures a class has only one instance and provides a global point of access to it.
    - **Usage**:
        - **UserManager**: This class is implemented as a singleton to manage the list of users in the application. It provides methods to add, remove, and check the existence of users.
4. **Factory Pattern**
    - **Purpose**: The Factory Pattern is used to create objects without specifying the exact class of object that will be created.
    - **Usage**:

- **UserFactory**: This class provides a static method to create instances of **User**. It abstracts the instantiation logic from the client.

5. **Composite Pattern**

    - **Purpose**: The Composite Pattern allows you to compose objects into tree structures to represent part-whole hierarchies. It lets clients treat individual objects and compositions of objects uniformly.
    - **Usage**:
        - **Group** and **SubGroup**: These classes implement the **GroupComponent** interface, allowing both groups and sub-groups to be treated uniformly. Groups can contain other groups as well as members.

Summary

The provided code uses several design patterns to structure and manage the functionality of a simplified social networking application. These patterns help in organizing the code, promoting reusability, and making the system more flexible and easier to maintain. The UML class diagram provides a visual representation of the relationships between the different classes and interfaces, while the design pattern documentation explains the purpose and usage of each pattern in the system.

User Guide for Simple Facebook Application

Welcome to the Simple Facebook Application! This guide will walk you through the features and functionalities of the application, ensuring you can make the most of your experience.

Table of Contents

Getting Started

1. **Launch the Application:**
    - Run the **Main** class to start the application.
    - The login window will appear.

Logging In

1. **Enter Your Credentials:**
    - Enter your ID and Password in the respective fields.
    - Click the "Login" button to proceed to the Home Page.

Home Page

1. **Main Interface:**

- The home page consists of a top panel for searching, a left panel for groups, and a center panel for your wall.
- You will see a welcome message and your current posts on the wall.

Posting on Your Wall

1. **Add a New Post:**
    - In the center panel, you will find a text field to write your post.
    - Enter your post content in the text field.
    - Click the "Post" button to add it to your wall.
    - The new post will appear below the text field.

Searching for Users

1. **Search for a User:**
    - Enter the name of the user you are looking for in the search field at the top panel.
    - Click the "Search" button.
    - A new window will open displaying search results.
2. **View User Profiles:**
    - If the searched user is found and their profile is visible, their name will appear as a button.
    - Click on the user's name to view their profile.

Viewing and Managing Groups

1. **View Your Groups:**
    - The left panel shows a list of groups you are a member of.
    - Double-click on a group name to open the group's page.
2. **Group Page:**
    - The group's page displays its members and sub-groups.
    - You can add or remove members if you have the necessary permissions.

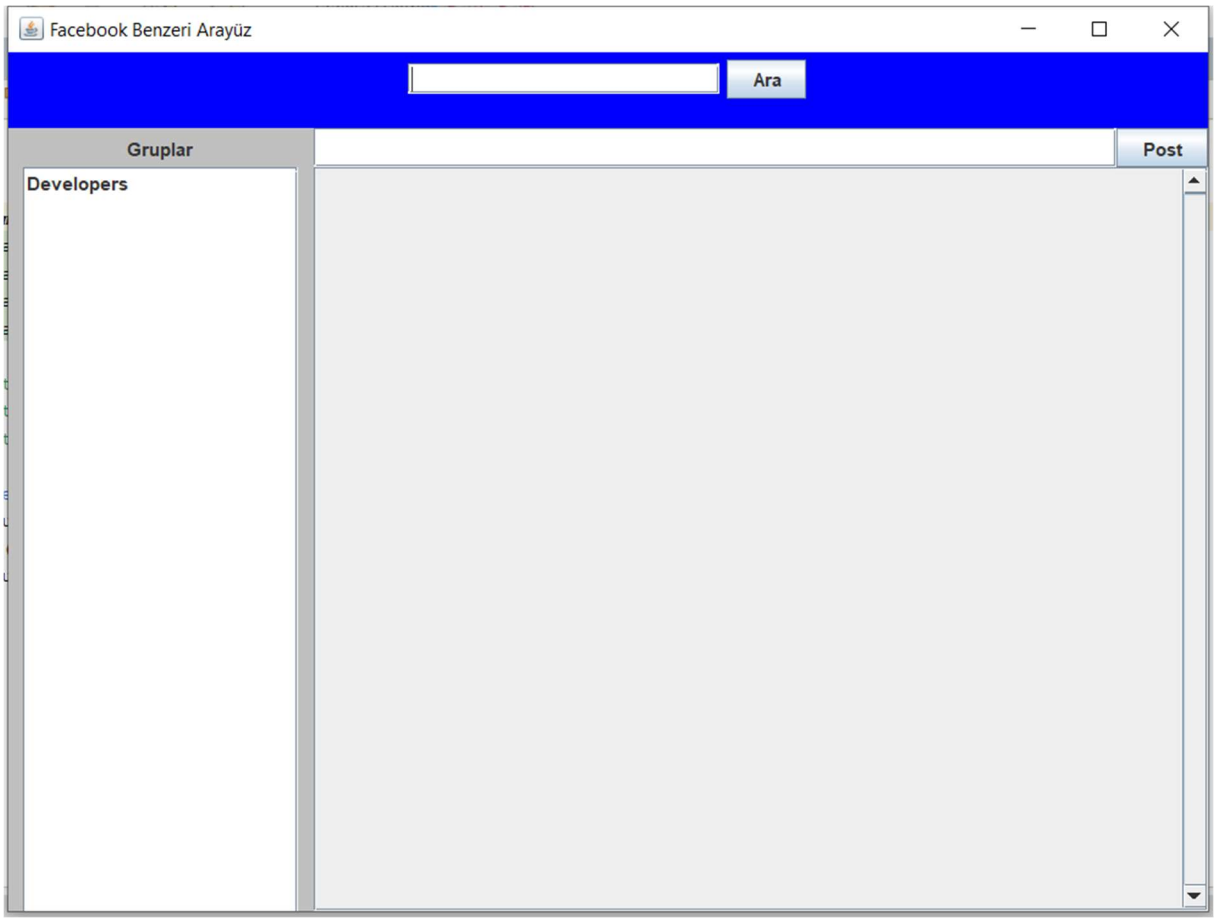
Managing Friends

1. **Viewing Friends:**
    - Navigate to a user's profile to view their friend list.
    - Use the "Add Friend" or "Remove Friend" button to manage friendships.
2. **Sending Friend Requests:**
    - Click on the "Add Friend" button on a user's profile to send a friend request.
    - The button will update to "Remove Friend" once the request is sent and accepted.

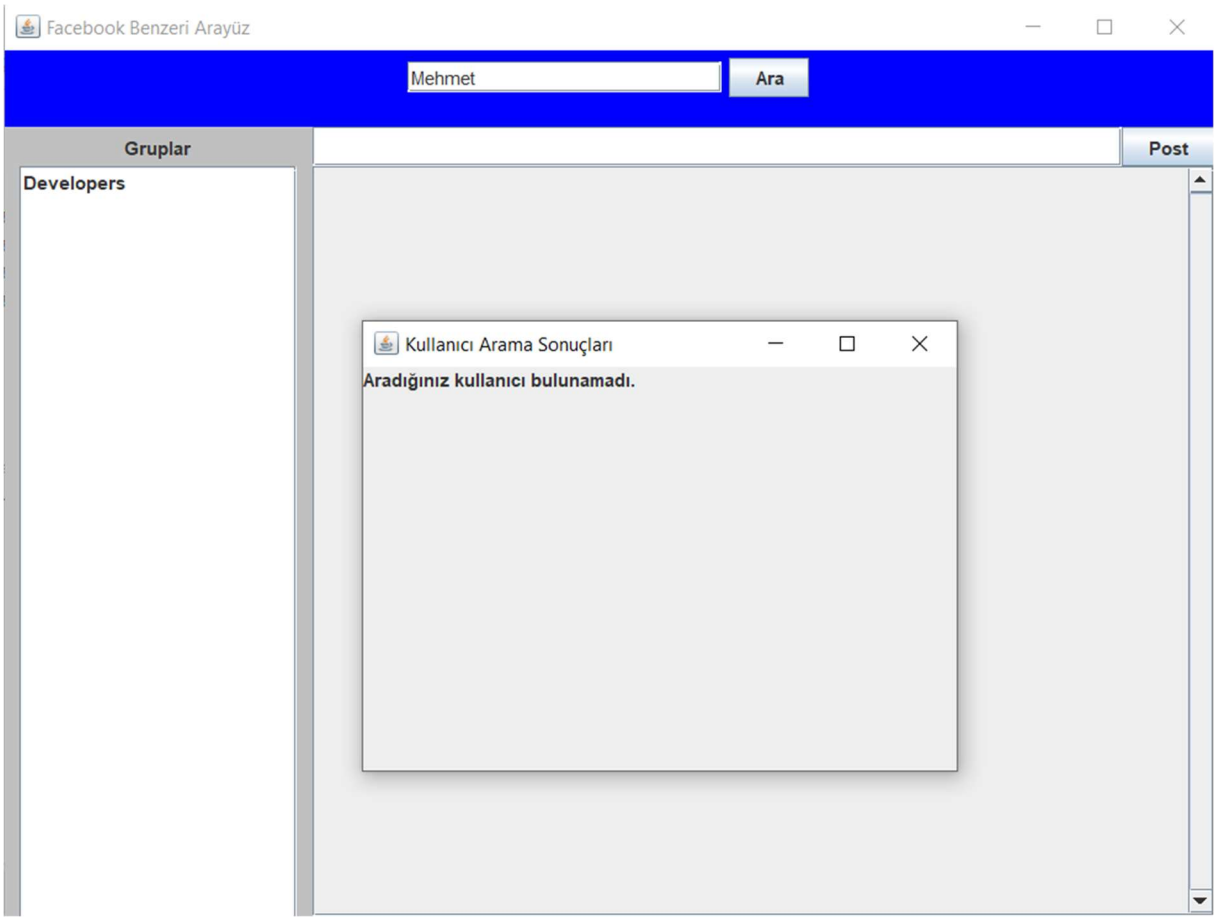
Group Management

1. **Adding Members:**
    - On a group's page, enter the name of the user you wish to add in the text field.
    - Click the "Add Member" button to add the user to the group.
    - The user will appear in the member list if added successfully.
2. **Removing Members:**
    - Enter the name of the user you wish to remove in the text field.
    - Click the "Remove Member" button to remove the user from the group.
    - The user will be removed from the member list if the operation is successful.
3. **Managing Sub-Groups:**
    - Sub-groups are listed in the group's page.
    - Click on a sub-group name to open its page and manage members similarly to the main group.

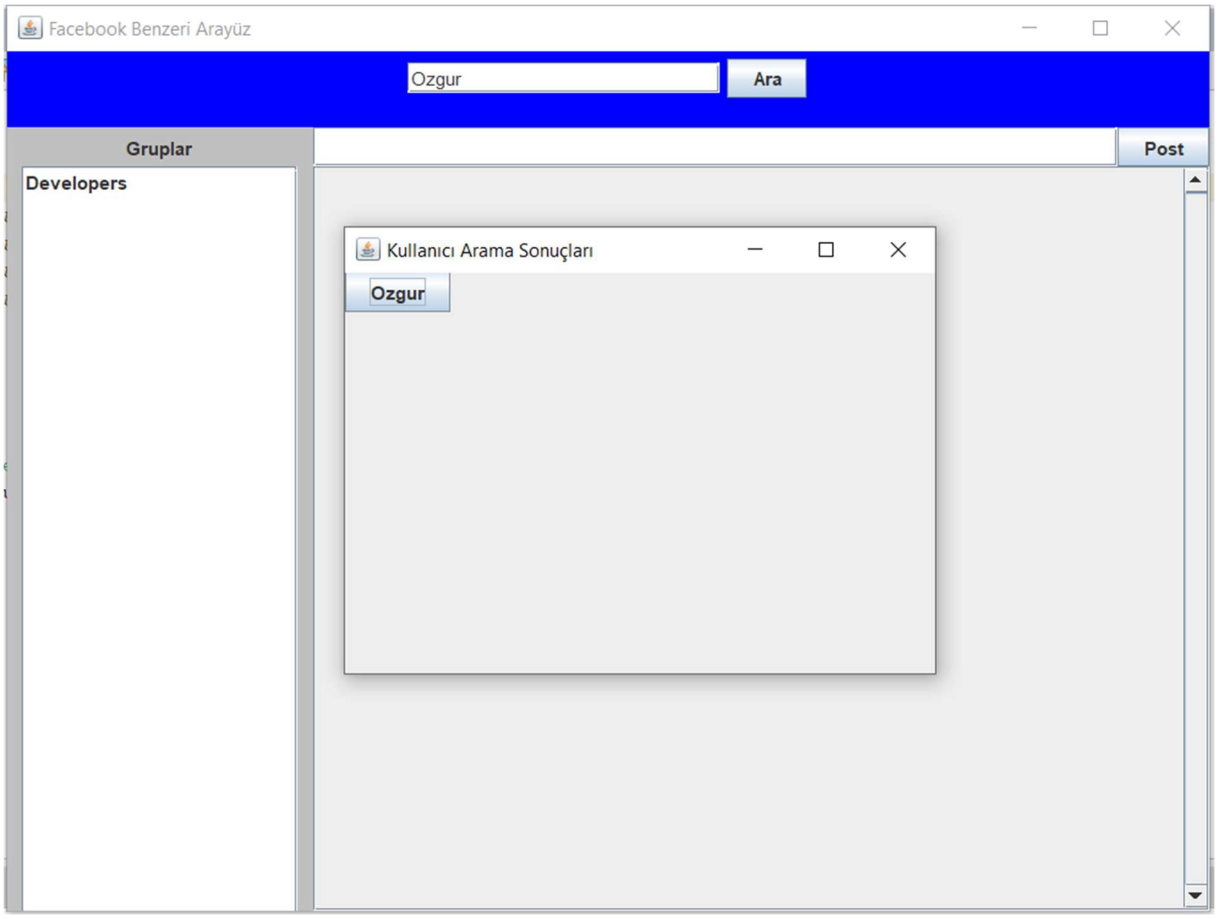
Screenshots:



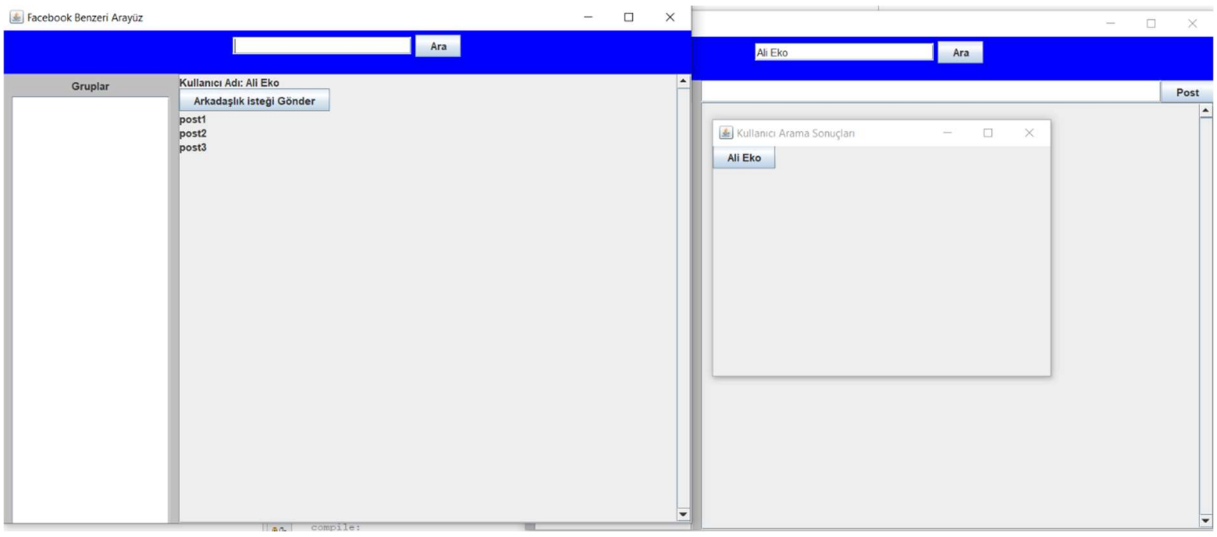
Search and post buttons.



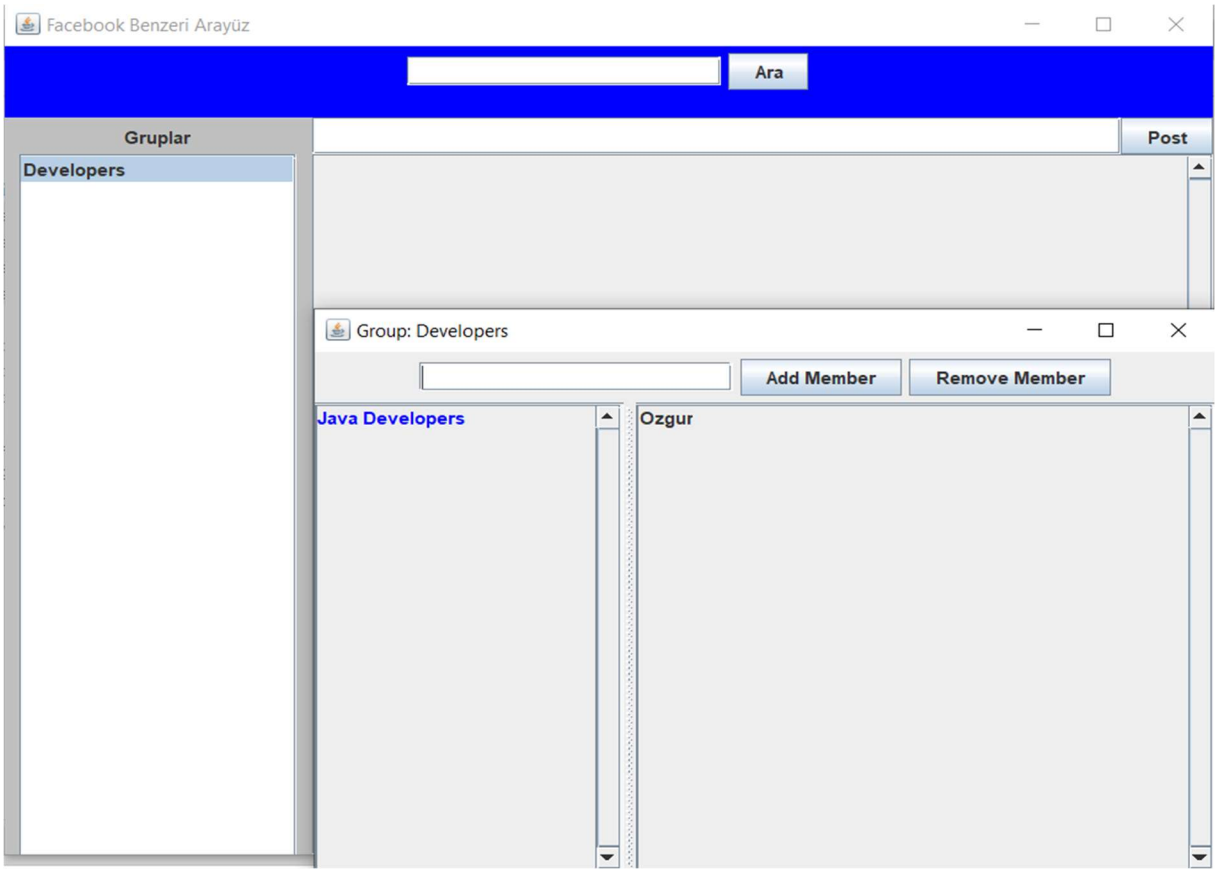
Error when searching for invalid user

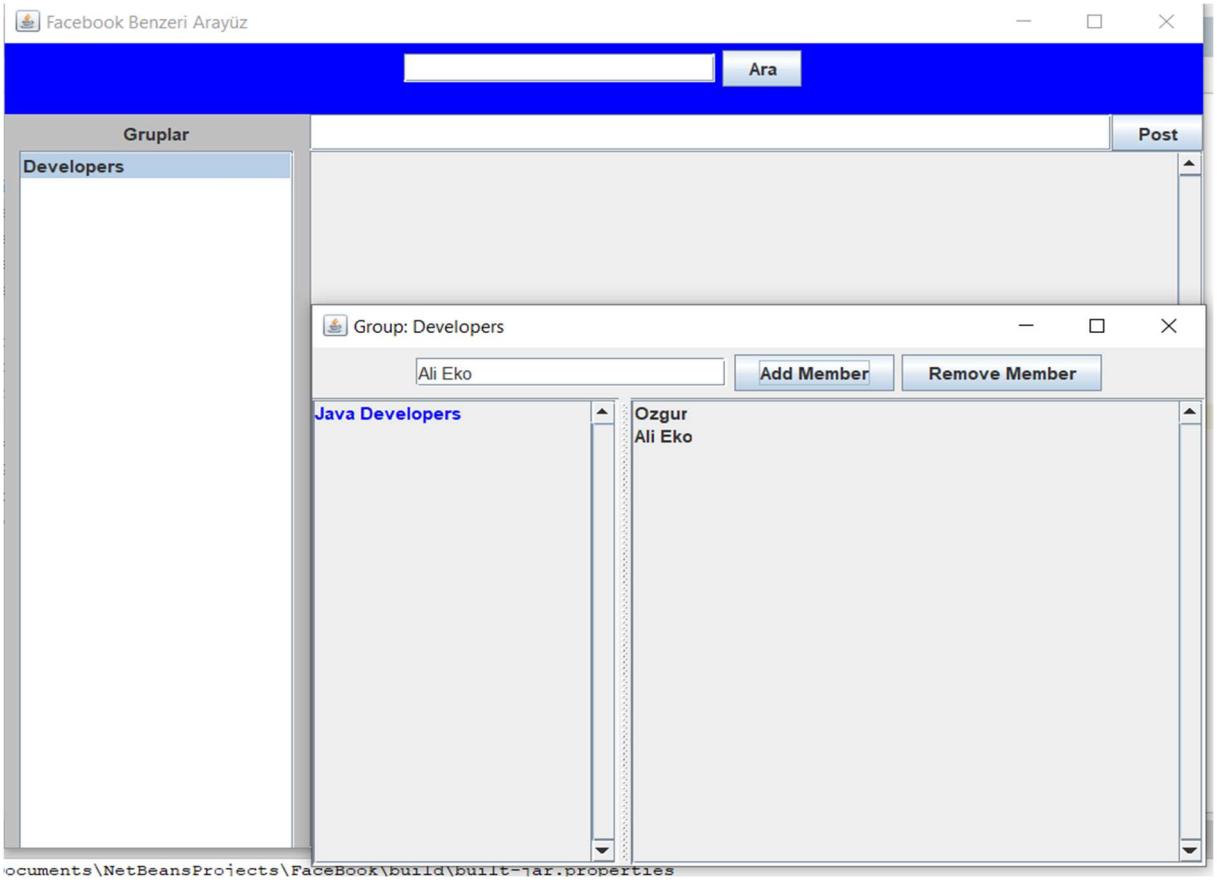


Searching for current user

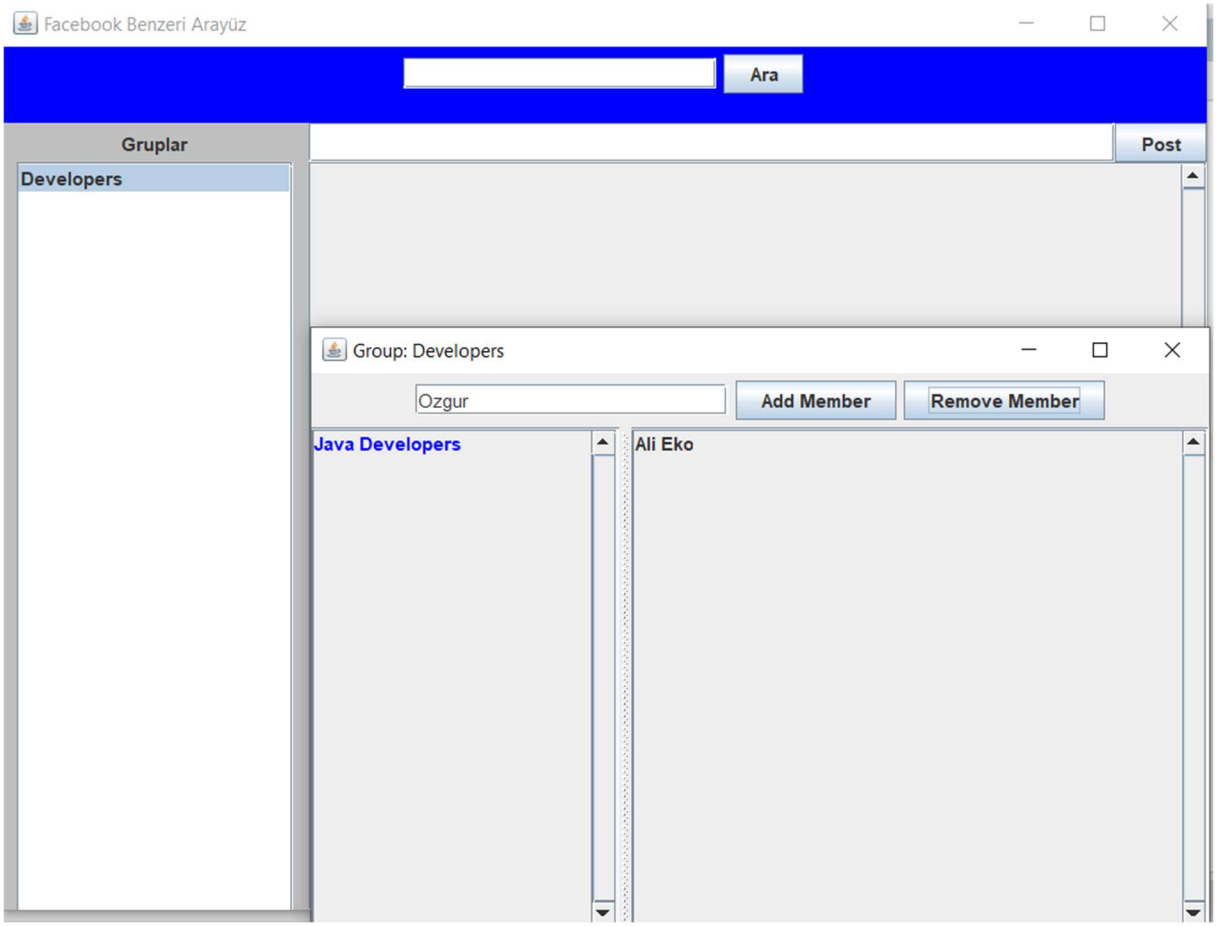


Send a friend request to the current user

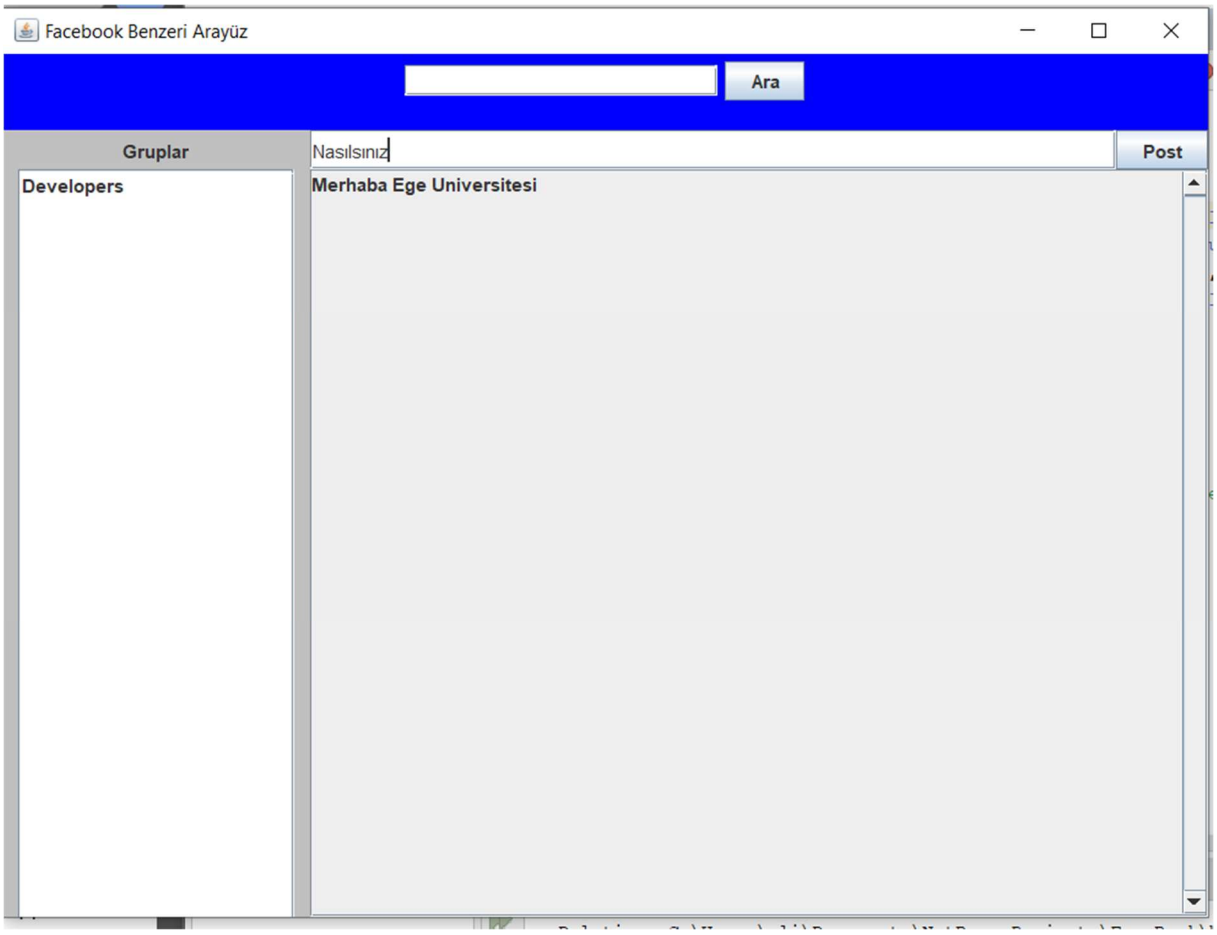




Adding users to the group(Ali Eko)



Delete a user from a group(Ozgur)


Facebook Benzeri Arayüz
Ara
Gruplar
Developers
Nasılsınız
Post
Merhaba Ege Universitesi

Share a post