Jerry Wang
Final Project Report

Throughout the development of my social network project, I adhered to the original plan outlined in HW6, while making some adjustments to the connection functions I used. Based on the social network social network UI from HW4, I added a searchbox using QLineEdit class, a search button using QPushButton, and a searchList using QListView that dynamically displays friends' names that match the entered search criteria. I implemented a search feature in my program to enable users to efficiently find friends by typing even partial names they may recall. To make this functionality work, I first added another function on network.cpp called getAllUserNames(). This functionality returns all user names in the social network document to provide the foundation to display relative names in searchList. I connect my functions and UI using three different connection functionality.

First:

```
  connect(ui->searchBox, &QLineEdit::textChanged, this,
&SocialNetworkWindow::updateSearchResults);
```

The updateSearchResults function provides a live search functionality within a social network program. The search is performed via a QLineEdit input, whereas users type in the names they wish to find, the associated QListView updates in real-time to display a filtered list of friend names that match the entered text, leveraging case-insensitive comparison. The updateSearchResults slot is connected to the textChanged signal of the QLineEdit and is triggered on each keystroke. If the search box is empty, the search result list is cleared and hidden. If there is input text, the code fetches a complete list of usernames from the network's backend storage, filters this list according to the current search query, and updates the view to show only the matching results. The search results are seamlessly displayed in a QListView, utilizing a QStringListModel that dynamically filters usernames from the complete user list obtained from the social network. This setup offers users immediate feedback during their search, thereby enhancing the user experience with efficient and responsive navigation to friends' profiles within the social network.

Second:

```
   connect(ui->searchButton, &QPushButton::clicked, this,
&SocialNetworkWindow::performSearch);
```

This code and corresponding function enable users to search for friends by name. When the search button is clicked, it triggers the performSearch function, which checks if the search input is non-empty. If not empty, it uses the Network object to find a user by the entered name. If a user is found, their profile information is retrieved and displayed, updating the personProfile widget with the user’s name and hiding the searchList. If no user is found, a message box informs the user that the search query did not return any results.

Third:

```
   connect(ui->searchList, &QListView::clicked, this,
&SocialNetworkWindow::onSearchListItemClicked);
```

This code establishes a connection between the clicked signal of the QListView (searchList) and the onSearchListItemClicked slot in the SocialNetworkWindow class. Upon clicking an item in searchList, the onSearchListItemClicked function extracts the username from the selected item and populates it into the QLineEdit (searchBox). This interaction efficiently streamlines user experience by automatically filling in the search box with the selected username and concurrently hiding the searchList.