Green University

Social Media Friend Recommendation System

Name: Ashab Uddin
ID: 232002274
Dept. of CSE
Green University of Bangladesh

Presented To
Farjana Akter Jui
Lecturer
Dept. of CSE
Green University of Bangladesh

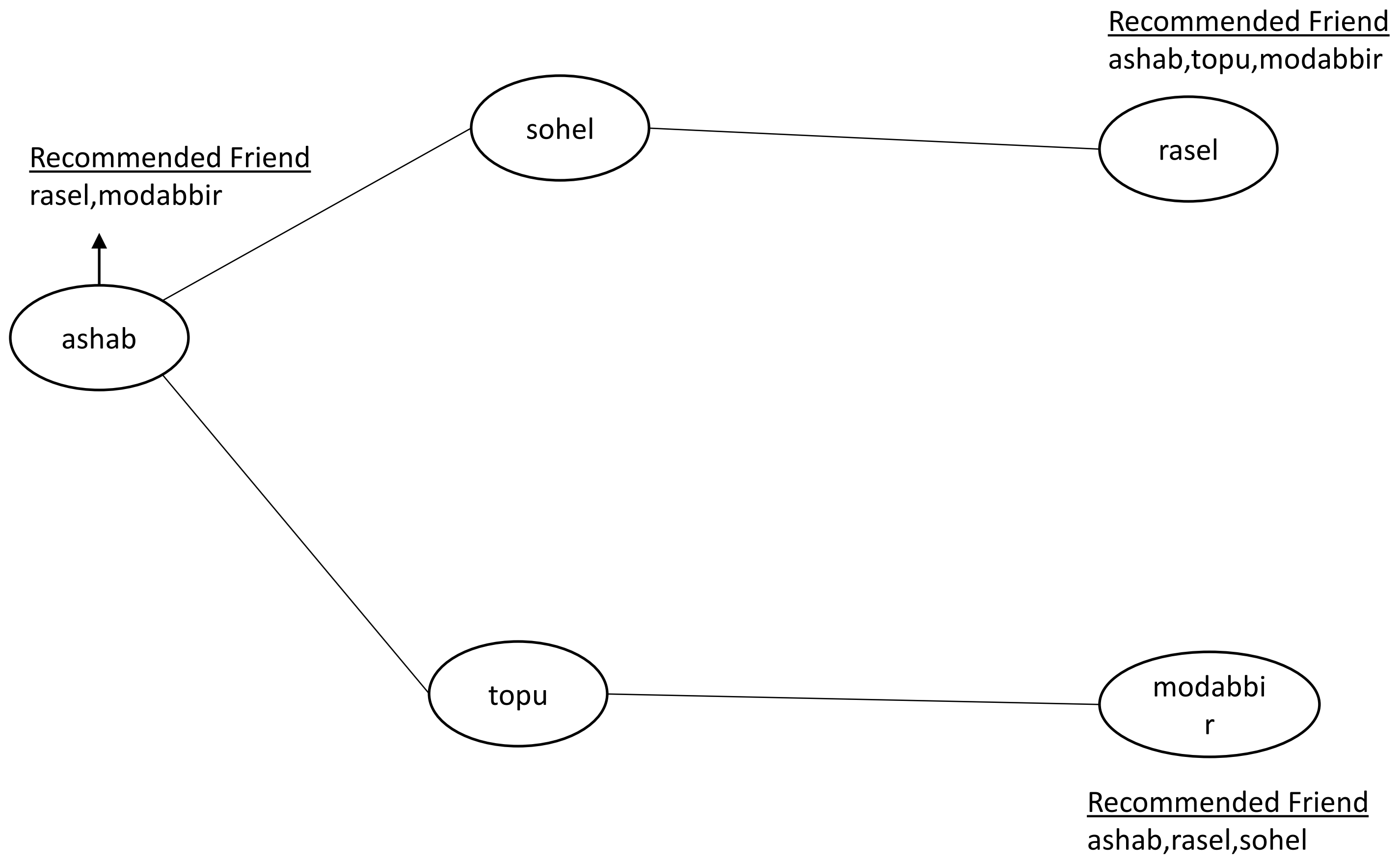
Key Features Implemented

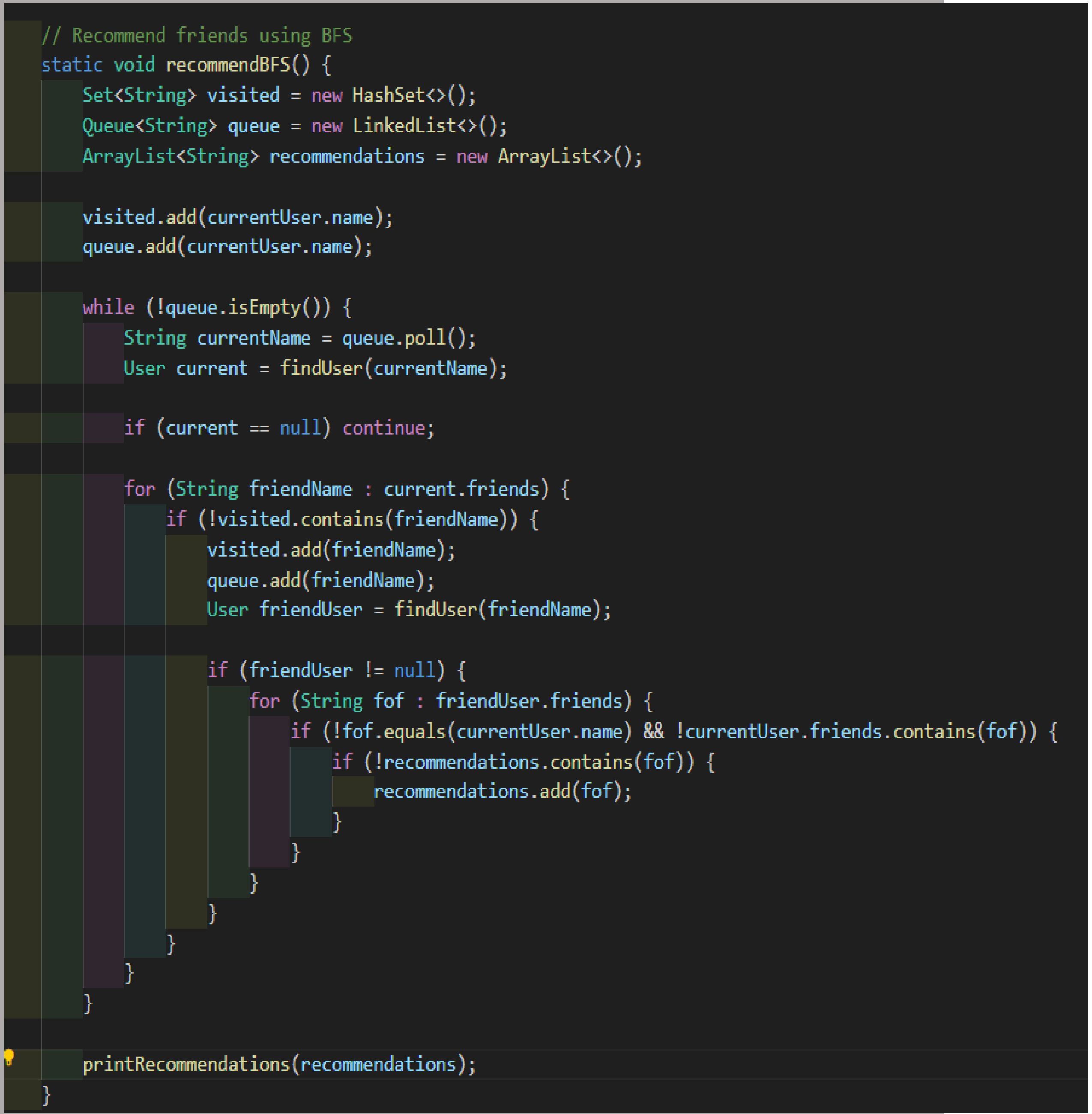
- User Login
- Send Friend Requests
- Accept Friend Requests
- Show all Friends
- Recommend Friends using BFS
- Recommend Friends using DFS

Algorithms Used

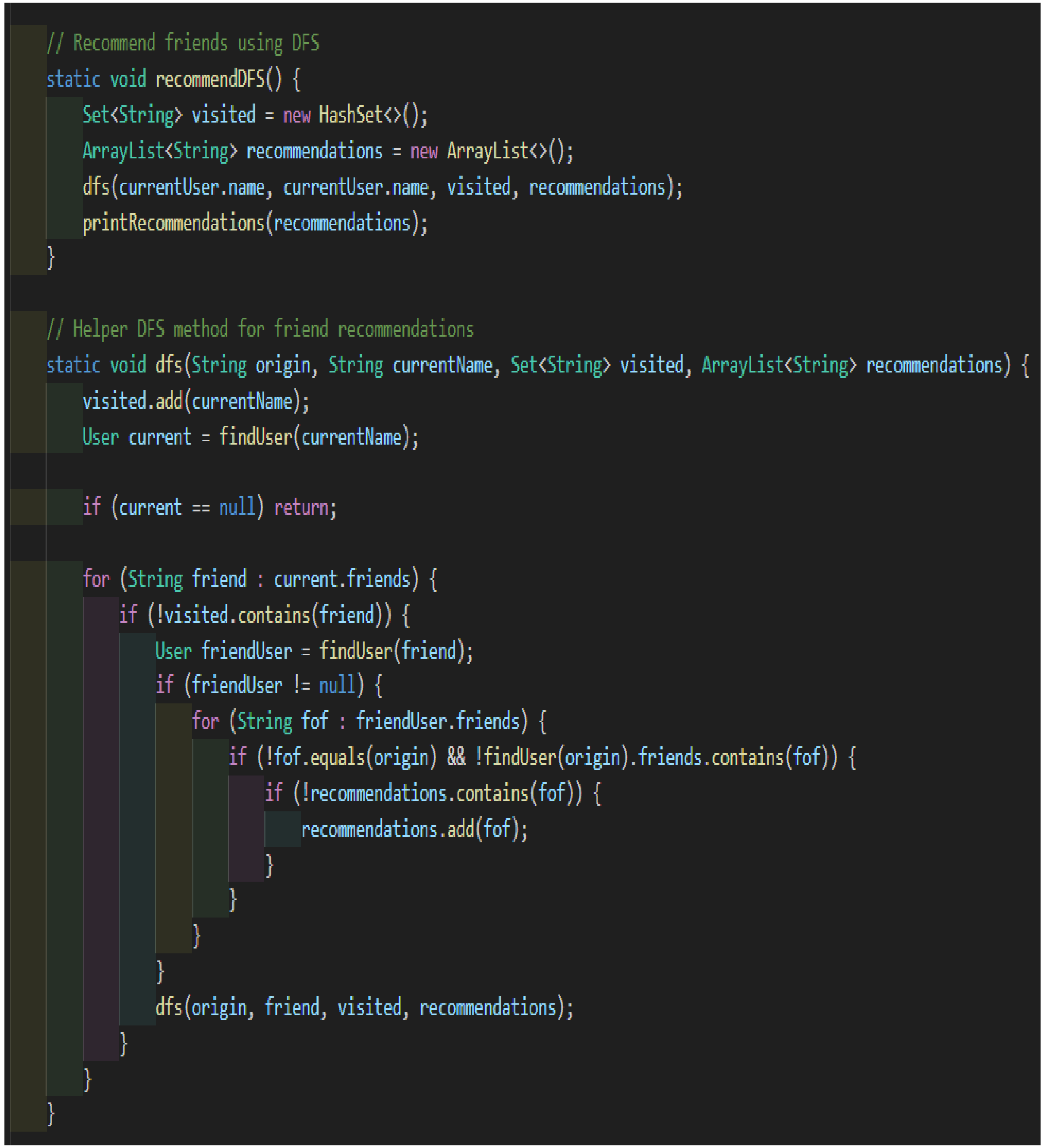
- ❑ **Friend Recommendation (BFS):** Suggests friends using Breadth-First Search (BFS).
- ❑ **Friend Recommendation (DFS):** Suggests friends using Depth-First Search (DFS).

Implemented Concept

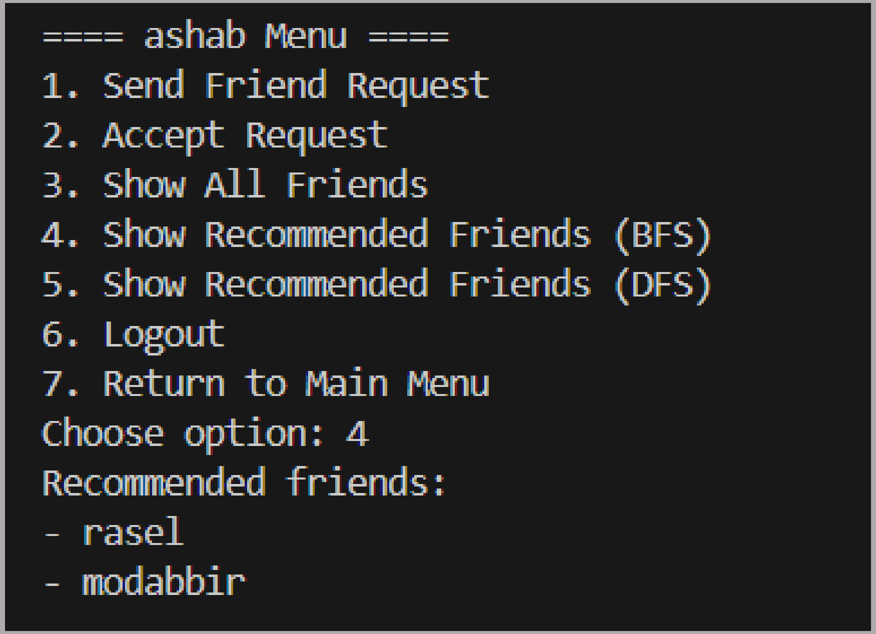




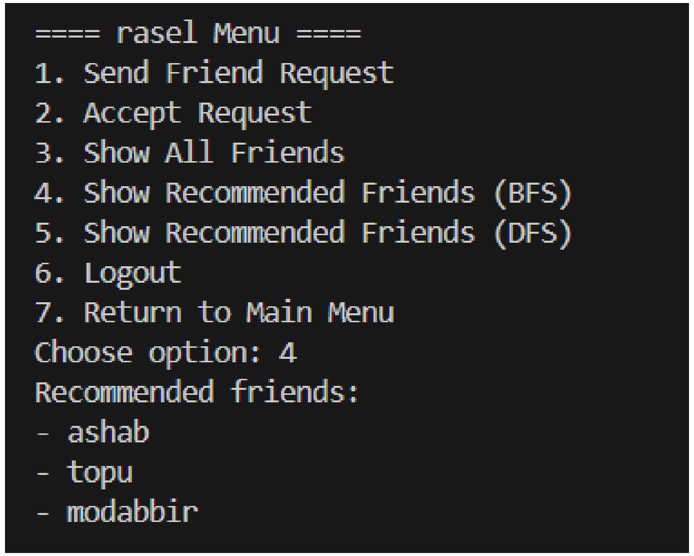
```java
// Recommend friends using BFS
static void recommendBFS() {
    Set<String> visited = new HashSet<>();
    Queue<String> queue = new LinkedList<>();
    ArrayList<String> recommendations = new ArrayList<>();

    visited.add(currentUser.name);
    queue.add(currentUser.name);

    while (!queue.isEmpty()) {
        String currentName = queue.poll();
        User current = findUser(currentName);

        if (current == null) continue;

        for (String friendName : current.friends) {
            if (!visited.contains(friendName)) {
                visited.add(friendName);
                queue.add(friendName);
                User friendUser = findUser(friendName);

                if (friendUser != null) {
                    for (String fof : friendUser.friends) {
                        if (!fof.equals(currentUser.name) && !currentUser.friends.contains(fof)) {
                            if (!recommendations.contains(fof)) {
                                recommendations.add(fof);
                            }
                        }
                    }
                }
            }
        }
    }

    printRecommendations(recommendations);
}
```



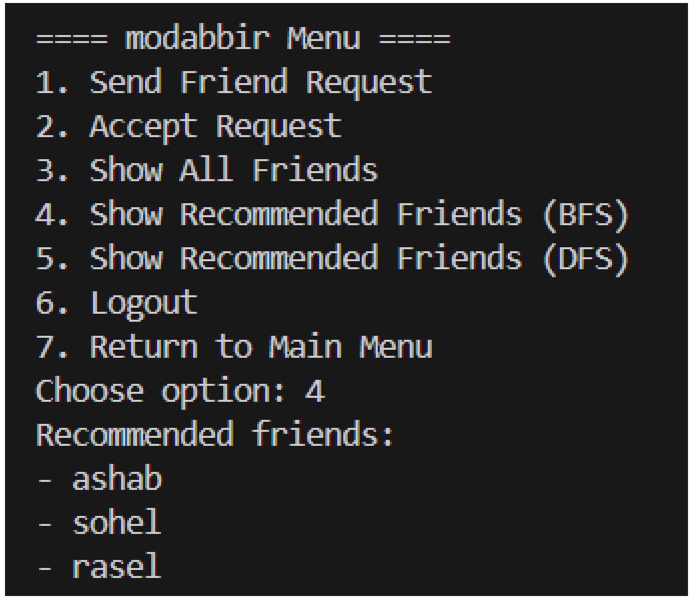
```java
// Recommend friends using DFS
static void recommendDFS() {
    Set<String> visited = new HashSet<>();
    ArrayList<String> recommendations = new ArrayList<>();
    dfs(currentUser.name, currentUser.name, visited, recommendations);
    printRecommendations(recommendations);
}

// Helper DFS method for friend recommendations
static void dfs(String origin, String currentName, Set<String> visited, ArrayList<String> recommendations) {
    visited.add(currentName);
    User current = findUser(currentName);

    if (current == null) return;

    for (String friend : current.friends) {
        if (!visited.contains(friend)) {
            User friendUser = findUser(friend);
            if (friendUser != null) {
                for (String fof : friendUser.friends) {
                    if (!fof.equals(origin) && !findUser(origin).friends.contains(fof)) {
                        if (!recommendations.contains(fof)) {
                            recommendations.add(fof);
                        }
                    }
                }
            }
            dfs(origin, friend, visited, recommendations);
        }
    }
}
```



```
==== ashab Menu ====
1. Send Friend Request
2. Accept Request
3. Show All Friends
4. Show Recommended Friends (BFS)
5. Show Recommended Friends (DFS)
6. Logout
7. Return to Main Menu
Choose option: 4
Recommended friends:
- rasel
- modabbir
```



```
==== rasel Menu ====
1. Send Friend Request
2. Accept Request
3. Show All Friends
4. Show Recommended Friends (BFS)
5. Show Recommended Friends (DFS)
6. Logout
7. Return to Main Menu
Choose option: 4
Recommended friends:
- ashab
- topu
- modabbir
```



```
==== modabbir Menu ====
1. Send Friend Request
2. Accept Request
3. Show All Friends
4. Show Recommended Friends (BFS)
5. Show Recommended Friends (DFS)
6. Logout
7. Return to Main Menu
Choose option: 4
Recommended friends:
- ashab
- sohel
- rasel
```

Future Improvements & Conclusion

❑ Future Improvements:

- Create a Graphical User Interface (GUI).
- Implement Database for data persistence.
- Suggesting Friend base on Interests.

Thank You