# Ateam 8

Project name: Social time

Zeyu Tan, Lec 01, x-team 21
Samuel Weng, Lec 01, x-team 117
Yuxin He, Lec 01, x-team 14
Shouzhe Li, Lec 01,xteam 21
Boya Zeng, Lec 02, xteam 98

- ## Class Summary

| ***enum, interface, class, abstract class*** | ***Name of the type*** | ***Description of use or purpose of this type*** |
|---|---|---|
| interface | GraphADT | Provide methods for Graph class to implement |
| class | Person | Represents a Person, which is a vertex in Graph, with his or her name, and friendsList. |
| class | Graph | Implements GraphADT and visualize friendship in the undirected graph |
| class | FriendsManager | Process person information input files and provide function that make information available to other users. |
| class | Main | Create the interactive window for program, including buttons that are clickable, and an input textbox for users. |
| class | PersonNotFoundException | The exception occurs when user wants to get the relationship of a not existed person |
| class | DuplicatePersonException | The exception occurs when user trying to add a duplicate person |

## Class Diagram

| Person |
|---|
| -name: String<br>-friends: List<String> |
| Person ()<br>Person (String name)<br>Person (String name, List<String> friends)<br>getName(): String<br>SetName(name : String) : void<br>getFriends(): List<String><br>setFriends(friends: List<String>): void |

<table>
<tr><td colspan="4">Person</td></tr>
<tr><td colspan="4"><b>Field</b></td></tr>
<tr><td>Field Type</td><td>Field Name</td><td colspan="2">Valid Range</td></tr>
<tr><td>String</td><td>name</td><td colspan="2">The names may contain any letters {A-Z}, space, digits {0-9}, underscore {_}, or apostrophe {'}</td></tr>
<tr><td>List<String></td><td>friends</td><td colspan="2">Friends name may contain any letters {A-Z}, space, digits {0-9}, underscore {_}, or apostrophe {'}</td></tr>
<tr><td colspan="4"><b>Constructor</b></td></tr>
<tr><td colspan="2">Constructor Names</td><td colspan="2">Parameter Types</td></tr>
<tr><td colspan="2">Person (default constructor)</td><td colspan="2">none</td></tr>
<tr><td colspan="2">Person</td><td colspan="2">String name</td></tr>
<tr><td colspan="2">Person</td><td colspan="2">String name, List<String> friends</td></tr>
<tr><td colspan="4"><b>Methods</b></td></tr>
<tr><td>Return Type</td><td>Method Name</td><td>Parameter List</td><td>Brief Description</td></tr>
<tr><td>String</td><td>getName</td><td>none</td><td>Get the name of the person</td></tr>
</table>

| | | | |
|---|---|---|---|
| void | setName | String name | Set the name of the person with the given name |
| List<String> | getFriends | none | Get the friends list of the person |
| void | setFriends | List<String> friends | Set the friends list of this person with the given friends list |

| Graph |
|---|
| personList: List<String> |
| Graph()<br>addPerson(String name): void<br>removePerson(): void<br>addRelationship(String name1, String name2):void<br>removeRelationship(String name1, String name2): void<br>getAllPerson(): List<String><br>getFriendsOfPerson(String name): List<String><br>removeAll(): void |

| Graph | | | |
|---|---|---|---|
| **Field** | | | |
| Field Type | Field Name | Valid Range | |
| List<String> | personList | none | |
| **Constructor** | | | |
| Constructor Names | | Parameter Types | |
| Graph (default constructor) | | none | |
| **Methods** | | | |
| Return Type | Method Name | Parameter List | Brief Description |
| void | addPerson | String name | Add a new person to this graph with the given name; Throw |

| | | | |
|---|---|---|---|
| | | | duplicatePersonException if the person is already existed |
| void | removePerson | none | Remove the person from the graph with the given game; Throw PersonNotFoundException if the person is not in the graph |
| void | addRelationship | String name1, name2 | Add the relationship between the two persons with the given two names. |
| void | removeRelationship | String name1, name2 | Remove the relationship between the two persons with the given two names. |
| List<String> | getAllPerson | none | Return all person names in the graph |
| List<String> | getFriendsOfPerson | String name | Return the friends list of the person with the given name(see adj list) |
| void | removeAll | none | Remove all person and corresponding friends within the graph |

| FriendsManager |
|---|
| sc: Scanner<br>data: File<br>graph: Graph |
| FriendManager ()<br>loadFile(File input): void<br>exportFile(): void |

```
constructGraph(String FileName) : Graph
searchPeople(String name): void
showFriends(String currentPerson): List<String>
showFriendsBtw(String Person1, Person 2): List<String>
showShortestPathBtw(String Person1, String Person2): List<String>
showGroupsNum(): int
```

<table>
<tr><td colspan="4">FriendsManager</td></tr>
<tr><td colspan="4"><b>Field</b></td></tr>
<tr><td>Field Type</td><td>Field Name</td><td colspan="2">Valid Range</td></tr>
<tr><td>Scanner</td><td>sc</td><td colspan="2">none</td></tr>
<tr><td>File</td><td>data</td><td colspan="2">File should be existed, or it will throw exception</td></tr>
<tr><td>Graph</td><td>graph</td><td colspan="2">none</td></tr>
<tr><td colspan="4"><b>Constructor</b></td></tr>
<tr><td colspan="2">Constructor Names</td><td colspan="2">Parameter Types</td></tr>
<tr><td colspan="2">FriendManager (default constructor)</td><td colspan="2">none</td></tr>
<tr><td colspan="4"><b>Methods</b></td></tr>
<tr><td>Return Type</td><td>Method Name</td><td>Parameter List</td><td>Brief Description</td></tr>
<tr><td>void</td><td>loadFile</td><td>File input</td><td>load a file with the person and relationships, then add them to the general graph.</td></tr>
<tr><td>void</td><td>exportFile</td><td>none</td><td>export the relationships and persons exists in graph to a data file.</td></tr>
<tr><td>Graph</td><td>constructGraph</td><td>String FileName</td><td>Construct a graph based on data in the given file; Throw fileNotFoundExceptio</td></tr>
</table>

|  |  |  |  |
|---|---|---|---|
|  |  |  | n if the file name is not found |
| void | searchPeople | String name | Find the people with the given name and shows in the graph as being the central with friends around it by calling getFriendsOfPerson( name) |
| List<String> | showFriendsBtw | String Person1, String Person 2 | Return a list of mutual friends between the two persons with the given name |
| List<String> | showShortestPathBtw | String Person1, String Person2 | Return a list of people to display the sequence of friends that is the shortest path between any two users within a connected component. |
| int | showGroupsNum | Graph graph | Return an integer that indicates the number of connected components (groups) in the social network |

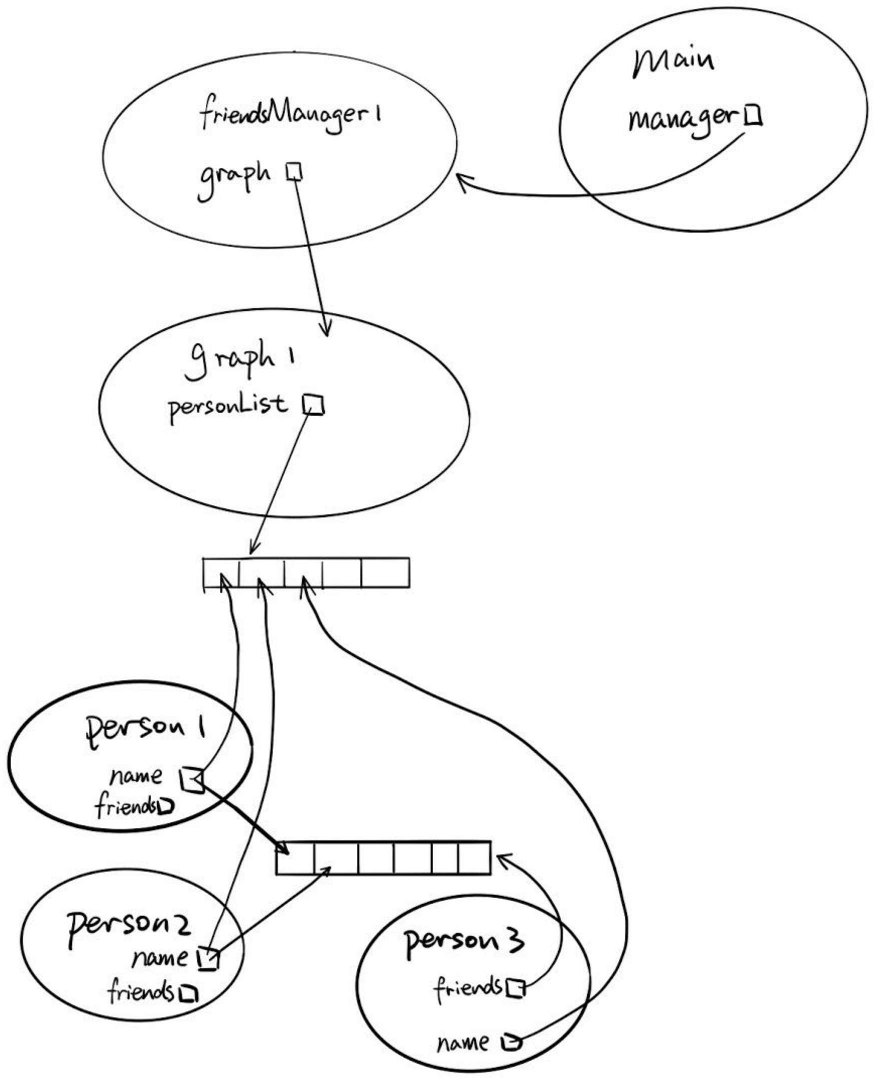
| Main |
|---|
| WINDOW_WIDTH: int<br>WINDOW_HEIGHT: int<br>APP_TITLE: String<br>log: File<br>args: List<String> |
|  |

Main (default constructor): void
start(Stage primaryStage): void
setButton(): void
setTextField(): void
fileReader(): String
main(): void

<table>
<tr><th colspan="3">Main</th></tr>
<tr><td colspan="3"><b>Field</b></td></tr>
<tr><td>Field Type</td><td>Field Name</td><td>Valid Range</td></tr>
<tr><td>List<String></td><td>args</td><td></td></tr>
<tr><td>int</td><td>WINDOW_WIDTH</td><td>final variable designed by coder with appropriate positive number</td></tr>
<tr><td>int</td><td>WINDOW_HEIGHT</td><td>final variable designed by coder with appropriate positive number</td></tr>
<tr><td>FriendsManager</td><td>manager</td><td>initialize a manager for methods to be called</td></tr>
<tr><td>String</td><td>title</td><td>Appropriate app title</td></tr>
<tr><td>File</td><td>log</td><td>.txt</td></tr>
<tr><td colspan="3"><b>Constructor</b></td></tr>
<tr><td colspan="2">Constructor Names</td><td>Parameter Types</td></tr>
<tr><td colspan="2">Main (default constructor)</td><td>none</td></tr>
<tr><td colspan="2">Main</td><td>String title</td></tr>
<tr><td colspan="3"><b>Methods</b></td></tr>
</table>

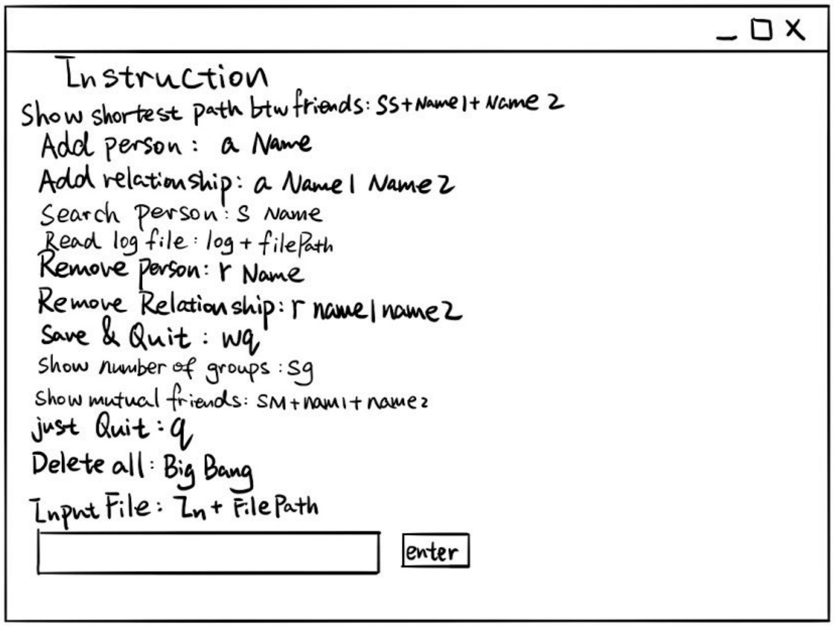
| Return Type | Method Name | Parameter List | Brief Description |
|---|---|---|---|
| void | start | Stage stage | these are GUI methods for get a window |
| void | launch | String[] args | these are GUI methods for get a window |
| void | main | String[] args | these are GUI methods for get a window |
| void | saveToLog | File log.txt | A simultaneous method to save instructions of user |
| void | showLog | none | read a status log for instructions |
| void | centerPerson | String name | center the person that has been chosen |
| void | handler | Button button | Set up a clickable button. If the button is clicked, then an even occur |

# Object Diagrams



# GUI Sketch

Initial interface:



Instruction

Show shortest path btw friends: SS+Name1+Name 2

Add person : a Name

Add relationship: a Name1 Name2

Search person: S Name

Read log file: log + filePath

Remove person: r Name

Remove Relationship: r name1 name2

Save & Quit : wq

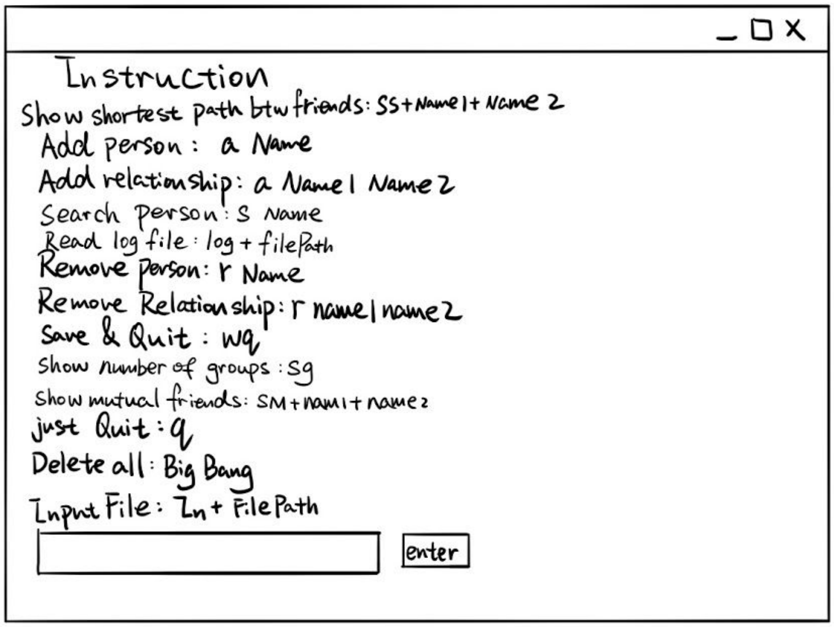
Show number of groups : sg

Show mutual friends: SM+nam1+name2

just Quit : q

Delete all: Big Bang

Input File: In+ FilePath



Instruction

Show shortest path btw friends: SS+Name1+Name 2

Add person : a Name

Add relationship: a Name1 Name2

Search person: S Name

Read log file: log + filePath

Remove person: r Name

Remove Relationship: r name1 name2

Save & Quit : wq

Show number of groups : sg

Show mutual friends: SM+nam1+name2

just Quit : q

Delete all: Big Bang

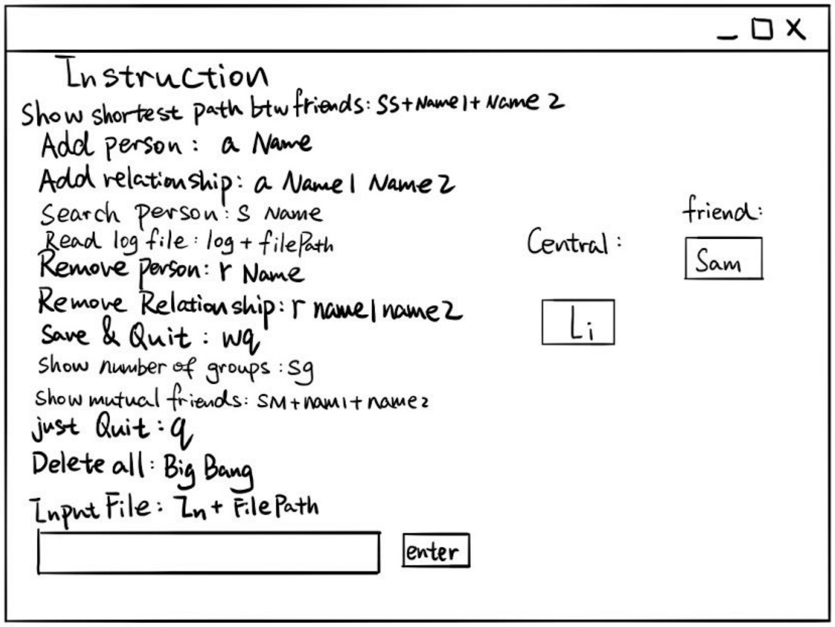
Input File: In+ FilePath

After type

"a Sam Li"

After type

"S Li"



Instruction

Show shortest path btw friends: SS+Name1+Name 2

Add person: a Name

Add relationship: a Name1 Name2

Search person: S Name

Read log file: log + filePath

Remove person: r Name

Remove Relationship: r name1 name2

Save & Quit: wq

Show number of groups: Sg

Show mutual friends: SM+nam1+name2

just Quit: q

Delete all: Big Bang

Input File: In+ FilePath