Suneeth Gopinathan Sujatha(s3675153)

Vishal Unnikrishnan Thottupurakal(s3672904)

Design

We have 5 classes in this assignment and 2 interfaces, of which class Person is an abstract class which has subclasses Adult and Child. Adult class implements methods from the interface called Children and Child class implements methods from interface called Parent. We have getters and setters given in Person class which has profile member variables in it which gets inherited by the both subclasses.

Main function is invoked in the class Mininet , which instantiates object from the driver class. Driver class is where all the process happens in the project. We use arraylist's to store data in time of execution. When the program starts we have 4 profiles of [Suneeth James Rose Joe] in a way that James and Rose are parents of Joe. We give those data via constructors and are instantly stored into the array list. We have Single dimensional array list to store usernames, age, status, image and key.

Key is the most important aspect in this project as it uniquely identifies each user, so while accessing profile along with name , key also acts as a credential. Then we have a 2 dimensional ArrayList which stores name of friends for each user-[friendList] and its corresponding key values for the users stored inside the array list are stored in [trackList]. trackID is a single dimensional arrayList which is used to store which user has these friends list, that is each index will contain corresponding key value or id of the user. So through finding the index from trackID we are able to extract the position of each user in friendList arrayList by means of the key or id value, which enables us to store data in it.

Class hierarchy is pretty simple as mentioned earlier an Abstract class called Person which subclasses Adult and Child which has interfaces Children and parent respectively. Driver class has all the functionalities of adding users , adding friends ,deleting accounts , displaying profile, checking whether direct friend or not etc.

So when using the arrayList to work with data ,it provides consistency when data is added or deleted. It provides various functions which are helpful while adding or removing data. The array list friendList and trackList provides a proper backtracking of friends of an user. Data are added into the list using a single dimensional arraylist and by adding it to the latter. In case of data deletion, we remove the corresponding row and insert it into the back of the list , since trackID keeps track of the position of the user , when a callback is initialized it makes it easy to fetch data without any problem.

Connections finding as said above is enabled via the those 3 arrayList which in way act as a 3 dimensional arrayList , so friends details are stored on. In case of children and parents, we have separate arrayList for that purpose as it is stored and updated as our program runs. Any changes are directly updated to these arrayLists. In design could have created friendlist single dimensional arrayList for each object of Child or Adult created but when deleting the account this seemed more logical and more challenging in many ways that's why opted for this method.

Since we have only 4 user's hardcoded into the system, so when creating users the key starts from 5 and so on. While creating users, design is in way that u cannot create a child as 5^{th} entry as there exists no other users who can be parent of that child. So we have to created much more users – Adult and since there are no other children left in the system we have assign children count of those users as 0 and later can connect that certain user to another adult and then they both can coexist as an parent to a child.