# Functional requirements

|  |  |
| --- | --- |
| **Name:** | R. #1. Find a path that takes to a new person who can be met |
| **Summary:** | Find a path that takes to a new person who can be met and this path is the best one |
| Input: |  |
|  |  |
| Results: | Path found |
|  | Path not found |

|  |  |
| --- | --- |
| **Name:** | R. #2. Model the force that unites a person with another |
| **Summary:** | This refers to the fact that they can be Friends, best friends or acquaintances |
| Input: | Force that user chose |
|  |  |
| Results: |  |
|  |  |

|  |  |
| --- | --- |
| **Name:** | R. #3. Show all the people that user follows |
| **Summary:** | Show the username of all the people that user follows |
| Input: |  |
|  |  |
| Results: | User follows with username |
|  | User doesn’t follow anyone |

|  |  |
| --- | --- |
| **Name:** | R. #4. Show all the user followers |
| **Summary:** | Show the username of all the user followers |
| Input: |  |
|  |  |
| Results: | User followers with username |
|  | User doesn’t have followers |

|  |  |
| --- | --- |
| **Name:** | R. #5. Cancellation of a customer’s account |
| **Summary:** | Delate client data from database and add it to an exclusive datbase  for those who desert |
| Input: |  |
|  |  |
| Results: | Changed the amount of the customer |
|  | User does not have a saving account |

|  |  |
| --- | --- |
| **Name:** | R. #6. Remove an specific follow |
| **Summary:** | User can unfollow users |
| Input: | String username |
|  |  |
| Results: |  |
|  |  |

|  |  |
| --- | --- |
| **Name:** | R. #7. Modify user information |
| **Summary:** | Modify user information (username or password) to change the password user need to enter actual password before change it |
| Input: | String newInformation |
|  |  |
| Results: | Information change correctly |
|  | Information could not be change |

|  |  |
| --- | --- |
| **Name:** | R. #8. Register a new user |
| **Summary:** | Register a new user that doesn’t have a account, user have to enter name and last name, gender, faculty, username and password |
| Input: | String name, String lastname, char gender, String faculty, String username, String password |
|  |  |
| Results: | User account was created correctly |
|  | User account could not be created |

|  |  |
| --- | --- |
| **Name:** | R. #9. Allow sign-in |
| **Summary:** | Allow sign-in in accounts already create if password and username match |
| Input: | String username, String password |
|  |  |
| Results: | Sign-in correctly |
|  | Username or password doesn’t match |

|  |  |
| --- | --- |
| **Name:** | R. #10. Delete account |
| **Summary:** | User can delete his account, this action Will remove all followers and won’t be able to sign-in again. To delete account user have to enter password |
| Input: | String password |
|  |  |
| Results: | Account deleted correctly |
|  | Password doesn’t matc, account couldn’t be deleted |

# Non-functional requirements

* Implement a graph (adjacent matrix and list)
* Implement graph methods like Dijkstra, BFS, DFS and others
* Must have a graphical interface