Use cases V3

Table of Contents

[1 General 8](#_Toc409419325)

[1.1 Register 8](#_Toc409419326)

[1.1.1 Register normally 8](#_Toc409419327)

[1.1.2 Register via other social networks (Facebook, Google +, Twitter, Linkedin) 9](#_Toc409419328)

[1.2 Sign in 9](#_Toc409419329)

[1.3 Change password 10](#_Toc409419330)

[1.4 Remember me 10](#_Toc409419331)

[1.5 Forgot the password 11](#_Toc409419332)

[1.6 Search information in general 11](#_Toc409419333)

[1.7 See more 12](#_Toc409419334)

[1.8 Loads contents for wall 13](#_Toc409419335)

[1.9 Care a status 14](#_Toc409419336)

[1.10 Uncare a status 15](#_Toc409419337)

[1.11 Care a comment 15](#_Toc409419338)

[1.12 Uncare a comment 16](#_Toc409419339)

[1.13 Comment to status 16](#_Toc409419340)

[1.14 Comment to a comment 17](#_Toc409419341)

[1.15 Share status 17](#_Toc409419342)

[1.16 Post status 18](#_Toc409419343)

[1.17 Delete status 18](#_Toc409419344)

[1.18 Delete a comment 19](#_Toc409419345)

[1.19 Load recent activities 19](#_Toc409419346)

[1.20 Add connection 20](#_Toc409419347)

[1.21 Notify for adding connections 20](#_Toc409419348)

[1.22 Answer connection notifications 21](#_Toc409419349)

[1.23 Notify new messages (from chatting) 21](#_Toc409419350)

[1.24 Read notification of new messages (chatting) 22](#_Toc409419351)

[1.25 Notify when there is a user who comments on your status 22](#_Toc409419352)

[1.26 Read notification when a user comment on your status 23](#_Toc409419353)

[1.27 Notify when there is a new comment for a comment of your status 23](#_Toc409419354)

[1.28 Read notification when there is a new comment for a comment of your status 24](#_Toc409419355)

[1.29 Notify when there is a user who shares a status to you 24](#_Toc409419356)

[1.30 Read notification when a user shares status to you 25](#_Toc409419357)

[1.31 Notify when there is a new comment for your comment 25](#_Toc409419358)

[1.32 Read notification when there is a new comment for your comment 25](#_Toc409419359)

[1.33 Notify when a group of career is shared to you 26](#_Toc409419360)

[1.34 Read notification when a group of career is shared to you 26](#_Toc409419361)

[1.35 Accept for joining to a shared group of career 27](#_Toc409419362)

[1.36 Send messages (chatting) 27](#_Toc409419363)

[1.37 Receive messages (chatting) 28](#_Toc409419364)

[1.38 Follow an individual, company 28](#_Toc409419365)

[1.39 Unfollow an individual, company 29](#_Toc409419366)

[1.40 Payment 29](#_Toc409419367)

[1.41 Upgrade 30](#_Toc409419368)

[2 Individuals 31](#_Toc409419369)

[2.1 Invite friends from other social networks (Facebook, Google +, Twitter, Linkedin) 31](#_Toc409419370)

[2.2 View basic information of other Indiviaduals 31](#_Toc409419371)

[2.3 View full profile of other individuals 32](#_Toc409419372)

[2.4 Edit basic information of individuals (owner) 33](#_Toc409419373)

[2.5 Edit background of individuals (owner) 34](#_Toc409419374)

[2.6 Add new photos to professional photo section 35](#_Toc409419375)

[2.7 Remove a photo from professional photo section 35](#_Toc409419376)

[2.8 Add new experience items to experiences of background of individuals (owner) 35](#_Toc409419377)

[2.9 Remove skills of individuals (owner) 36](#_Toc409419378)

[2.10 Add new skills of individuals (owner) 37](#_Toc409419379)

[2.11 Add new education\_item to education of individuals (owner) 37](#_Toc409419380)

[2.12 edit education\_item to education of individuals (owner) 38](#_Toc409419381)

[2.13 Edit contact info of individuals 38](#_Toc409419382)

[2.14 Add reference item 39](#_Toc409419383)

[2.15 Remove a reference 39](#_Toc409419384)

[2.16 Insert suggested post about companies to list of changes in the wall of individuals 40](#_Toc409419385)

[2.17 Load suggested employments to individuals (worker) 40](#_Toc409419386)

[2.18 Load suggested groups of careers to individuals (worker) 41](#_Toc409419387)

[2.19 Edit privacy for individuals (students, worker) 41](#_Toc409419388)

[2.20 Search (advanced) 42](#_Toc409419389)

[2.21 Edit basic information of individuals (after register) 42](#_Toc409419390)

[2.22 Add an avatar of individuals 43](#_Toc409419391)

[2.23 Suggest 100 employments based on the last searching values saved 43](#_Toc409419392)

[2.24 Statistic list of actors viewing your profile of individual 44](#_Toc409419393)

[2.25 Advertise the profile of vip individuals to1000 companies 44](#_Toc409419394)

[2.26 Introduce 100 companies to VIP individuals for following 45](#_Toc409419395)

[3 Companies 45](#_Toc409419396)

[3.1 View full profile of other companies 45](#_Toc409419397)

[3.2 Edit basic information of profile of companies (owner) 46](#_Toc409419398)

[3.3 Edit the introductory paragraph of background of companies (owner) 47](#_Toc409419399)

[3.4 Edit an employment item of employment section of profile of companies (owners) 47](#_Toc409419400)

[3.5 View an employment in details 48](#_Toc409419401)

[3.6 Apply for an employment of a company 48](#_Toc409419402)

[3.7 Post an employment in profile of company 49](#_Toc409419403)

[3.8 Edit privacy for companies 49](#_Toc409419404)

[3.9 Search (advanced) 50](#_Toc409419405)

[3.10 Edit basic information of company(after register) 50](#_Toc409419406)

[3.11 Add an logo of comapny (after-resigter) 51](#_Toc409419407)

[3.12 Suggest 100 profiles of individuals based on the last searching values saved 51](#_Toc409419408)

[3.13 Statistic list of actors viewing your profile of company 52](#_Toc409419409)

[3.14 Advertise VIP companies to 1000 individuals (worker) 52](#_Toc409419410)

[3.15 Introduce employments of VIP company to individuals(worker) 53](#_Toc409419411)

[4 Groups 53](#_Toc409419412)

[4.1 Comment to status in group (of career, of schools) 53](#_Toc409419413)

[4.2 Groups of careers 54](#_Toc409419414)

[4.2.1 Edit profile of group (of career) 54](#_Toc409419415)

[4.2.2 Post an employment in group (of career) 54](#_Toc409419416)

[4.2.3 Load popular status in group (of career) 55](#_Toc409419417)

[4.2.4 Load list of employments in group (of career) 55](#_Toc409419418)

[4.2.5 Load list of members of group 56](#_Toc409419419)

[4.2.6 Load status in group of careers 56](#_Toc409419420)

[4.2.7 Save employments in group (of career) 57](#_Toc409419421)

[4.2.8 Share an employment of group (of career) 57](#_Toc409419422)

[4.2.9 Join to a group (group of career) 58](#_Toc409419423)

[4.2.10 Unjoin a group of career 58](#_Toc409419424)

[4.2.11 Display advertisements in groups of careers 59](#_Toc409419425)

[4.2.12 Load introduction part of groups of careers when actors visit 59](#_Toc409419426)

[4.2.13 Share a group of career to other users 60](#_Toc409419427)

[4.2.14 Answer to notification inviting to join a group of career 60](#_Toc409419428)

[4.3 Groups of schools 61](#_Toc409419429)

[4.3.1 View profile of group of school when actors visit 61](#_Toc409419430)

[4.3.2 Edit profile of group of school 61](#_Toc409419431)

[4.3.3 View list of current students of group of school 62](#_Toc409419432)

[4.3.4 Add students to groups of schools automatically (AFTER completing profile-after register) 62](#_Toc409419433)

[4.3.5 Follow a group (group of schools) 63](#_Toc409419434)

[4.3.6 Unfollow a group of school 63](#_Toc409419435)

[4.3.7 View list of ex- students of group of school 64](#_Toc409419436)

[4.3.8 Specify yourself as a member of the group of school (individuals) 64](#_Toc409419437)

[4.3.9 Shows suggested groups of schools related to the group of school that users are viewing 65](#_Toc409419438)

[5 Administration 66](#_Toc409419439)

[5.1 Manage school set 66](#_Toc409419440)

[5.1.1 Add new school 66](#_Toc409419441)

[5.1.2 Remove a school 66](#_Toc409419442)

[5.1.3 Update a school 67](#_Toc409419443)

[5.2 Manage career set 67](#_Toc409419444)

[5.2.1 Add new career 67](#_Toc409419445)

[5.2.2 Remove a career 68](#_Toc409419446)

[5.2.3 Update a career 68](#_Toc409419447)

[5.3 Manage groups 69](#_Toc409419448)

[5.3.1 Groups of careers 69](#_Toc409419449)

[5.3.2 Groups of schools 69](#_Toc409419450)

[5.4 Manage advertisement 71](#_Toc409419451)

[5.4.1 Add an advertisement 71](#_Toc409419452)

[5.4.2 Remove an advertisement 71](#_Toc409419453)

[5.5 Manage types of companies 72](#_Toc409419454)

[5.5.1 Add a type of company 72](#_Toc409419455)

[5.5.2 Edit a sort of company 73](#_Toc409419456)

[5.5.3 Delete a sort of company 73](#_Toc409419457)

[5.6 Manage level of position 74](#_Toc409419458)

[5.6.1 Add a level of position 74](#_Toc409419459)

[5.6.2 Edit a level of position 74](#_Toc409419460)

[5.6.3 Delete a level of position 75](#_Toc409419461)

[5.7 Manage degree (of individuals) (Bachelor, Master, PhD,…) 76](#_Toc409419462)

[5.7.1 Add a degree 76](#_Toc409419463)

[5.7.2 Edit a degree 76](#_Toc409419464)

[5.7.3 Delete a degree 77](#_Toc409419465)

[5.8 Manage working states of individuals 77](#_Toc409419466)

[5.8.1 Add a working state 77](#_Toc409419467)

[5.8.2 Edit a working state 78](#_Toc409419468)

[5.8.3 Delete a working state 78](#_Toc409419469)

[5.9 Manage users 79](#_Toc409419470)

[5.9.1 View list of users and do statistics for numbers of users 79](#_Toc409419471)

[5.9.2 View list of active users and do statistics for numbers of active users 80](#_Toc409419472)

[5.10 Statistics 80](#_Toc409419473)

[5.10.1 Do statistics about total viewing times in a duration 80](#_Toc409419474)

[5.10.2 Do statistics about number of active and inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) 81](#_Toc409419475)

[5.10.3 Do statistics about number of active users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) 82](#_Toc409419476)

[5.10.4 Do statistics about number of inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) 82](#_Toc409419477)

[5.10.5 Do statistics about proportion of numbers of students, individuals finding jobs, workers, companies in a specific period (pie chart) 83](#_Toc409419478)

[5.10.6 Do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart) 84](#_Toc409419479)

[5.10.7 Do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart) 85](#_Toc409419480)

[5.10.8 Do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart) 85](#_Toc409419481)

[5.10.9 Do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart) 86](#_Toc409419482)

# General

## Register

### Register normally

|  |
| --- |
| **Use case**: Register (normal) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Individuals provide **first\_name**, **last\_name**, **email**, **password** (2 times), **gender** (male-female), **is student or worker**. Companies provide **company\_name**, **first\_name** and **last\_name** of registers, **email**, **password** (2 times)  N2. The system check the email whether the email existed, password and confirming password are the same  N3. Create an account for actor but this account is not active.  N4. The system saves creating a new account into log table (id of new account, type of action, creating time)  N5. The system sends an email that includes an active link to the email address of actors  N6. Actors click the active link and will be redirect to active page of GJOBS. Then the account will be active officially  N7. The system saves an account active into log table (id of new account, type of action, active time) |
| **Alternative course:**  A2-a. Email of register existed  A2.a.1 Require enter other email address  A2-b. Password and confirming password are not correct  A2.-b.1. Re-enter passwords |
| **Exceptions:** |
| **Post-conditions:**  New account |

Notes:

An email is only used for an account

An actor can create an account delegating for a company. So there will be many accounts delegating for a company. But these accounts will act independently.

### Register via other social networks (Facebook, Google +, Twitter, Linkedin)

|  |
| --- |
| **Use case:** Register via other social networks (Facebook, Google +, Twitter, Linkedin) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Individuals provide **email**, **password** of other social networks, verify they are **students or workers**  N2. The system log in other social networks and get information of actors  N3. Create an account for actor but this account is not active  N5. The system saves creating a new account into log table (id of new account, type of action, creating time)  N6. The system sends an email that includes an active link to the email address of actors  N7. Actors click the active link and will be redirect to active page of GJOBS. Then the account will be active official  N8. The system saves an account active into log table (id of new account, type of action, active time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New account |

## Sign in

|  |
| --- |
| **Use case:** Sign in |
| **Actor:** administrators- subAdministrators, individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors inputs **email** and **password**  N2. The system gets email and password based on **email** from database  N3. The system compares email and password of inputs to email a password in database  N4. The system saves action sign-in into log table (id of account, type of action, sign-in time) |
| **Alternative course:**  A1-a. Signing-in has failed 3 times  A1.1. The account is locked temporarily and redirect to the page resetting password  ~~A1.2. The system saves the action “fail 3 times” into log table (id of account, type of action, time)~~  A1.3. Actor inputs email which was used to register the account to receive the links for resetting password  A1.4. The system send the link for resetting password to the email of user  A1.5. In the reset password page, actor determine new password and confirm new password  A1.6. The system saves new password  A1.7. The system saves the action “new password” into log table (id of account, type of action, time) |
| **Exceptions:** |
| **Post-conditions:**  Log in is successful or not successful |

## Change password

|  |
| --- |
| **Use case:** Change password |
| **Actor:** administrators- subAdministrators, individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors want to change password |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors inputs **newpassword** and **confirming password**  N2. The system compares **newpassword** and **confirming password** that are equal  N3. The system saves **newpassword** into database  N4. The system saves the action “change password” into log table (id of account, type of action, changing-password time) |
| **Alternative course:**  A2-a. newpassword and confirmingpassword are not equal  A2-a.1 Re-enter newpassword and confirmingpassword |
| **Exceptions:** |
| **Post-conditions:**  New password |

## Remember me

|  |
| --- |
| **Use case:** Remember me |
| **Actor:** administrators- subAdministrators, individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1.The system save current **email** and **password** to a cookie for next signing-in |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New password |

## Forgot the password

|  |
| --- |
| **Use case:** forgot the password |
| **Actor:** administrators- subAdministrators, individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system redirect the actors to screen “forgot the password”  N2. Actors enter **email** which was used to register to receive the **link** for reset the password  N3. The system send the email for the reset the password to the email of actor  N4. Actors visit the new screen where they enter **new password** and **confirm new password**  N5. The system saves the action “change password” into log table (id of account, type of action, changing-password time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New password |

## Search information in general

|  |
| --- |
| **Use case:** Search information |
| **Actor:** administrators- subAdministrators, individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide the string for searching  N2. The system compares the string with ***name\_of\_individuals****,* ***current\_job\_of\_individuals****,* ***name\_of\_companies****,* ***name\_of\_ groups\_of\_career*** *or* ***name\_of\_groups\_of\_school, name of employment***  N3. The system shows results to actors as follow:   * ***Individuals***(students, worker)*: Last\_name+first\_name, avatar, number of followers, current job, current company, type of element* * ***Employments****: Name of job, name of company, salary ( fixed or deal), type of element* * ***Company****: name of company, avatar, numbers of followers, field of company or career, type of element* * ***Group of career****: name of group, avatar, numbers of members, type of element* * ***Group of schools****: name of group, avatar, number of members, type of element* |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Results are showed |

*??How many results will be showed in a page*

*??How to extend more results for viewing*

*??Sort: alphabel*

## See more

|  |
| --- |
| **Use case:** see more |
| **Actor:** all |
| **Description:** when loads list of something. We don’t load all of element in same time. We just load some of them and load more if we need |
| **Trigger:** |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provides the id of last item, type of element, numbers of elements which they want to see, value for sorting  N2. The system loads desired elements base type  N3. The system displays to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  More item are showed |

## Loads contents for wall

|  |
| --- |
| **Use case:** loads contents for wall |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors refresh the wall |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Base on email of actors. The system loads all following items based on the changing time of the last action:   * Status that the actors post. * Status that the actors care. * Status from their groups of career, groups of school * Status from followed individuals, followed companies * Status that followed individual or companies has commented * Status that friends care * Status that friends comment * Status that your connections, followed individuals or connections are shared * Status that has comment cared by your connection, your followed individuals or companies. * Employments from followed companies * Status that is shared by other   An element of the list:   * Status that the actors post or followed individual or companies post: avatar or logo of poster,name of poster, state of status (There is someone who has just comment to the status,... Just now-1 minute), content, posting time, numbers of caring, numbers of comments, numbers of sharing * Comments: avatar or logo of poster, name of poster, content, number of caring, number of comments * Status from group: name of group, avatar or logo of poster,name of poster, state of status (There is someone who has just comment to the status,... Just now-1 minute), content, posting time, numbers of caring, numbers of comments, numbers of sharing * Status is issued from your connections, followed individuals or companies, Status that is shared by other: avatar and name of followed individuals or companies, name of action that followed individuals and companies do ,avatar or logo of poster,name of poster, state of status (There is someone who has just comment to the status,... Just now-1 minute), content, posting time, numbers of caring, numbers of comments, numbers of sharing * Status of employment: avatar of poster, name of company, content (name of job, location for working, career, salary), state of status, content, posting time, numbers of caring, numbers of comments, numbers of sharing.   N2. The system displays the list of changes of status based on timeline |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Status, suggested companies related to actors are showed |

“Care” an comment: status will be not showed in the our wall

Number of “care” of comment are different from number of “care” status

## Care a status

|  |
| --- |
| **Use case:** care a status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors click care |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id and id of the status.  N2. The system saves information of “care” including id of actors and id of the status into database.  N3. The system saves the action “care a status” into log table (id of account, type of action, id of status, caring time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Care a status |

## Uncare a status

|  |
| --- |
| **Use case:** uncare a status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors click uncare |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id and id of status.  N2. The system removes relationship “care” status of actor from database  N3. The system saves the action “uncare a status” into log table (id of account, type of action, id of status, uncaring time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  uncare a status |

## Care a comment

|  |
| --- |
| **Use case:** care a comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors click care |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of actor and id of the status, id of cared comment  N2. The system saves their id and id of the status, id of cared comment, time into database.  N3. The system saves the action “care a comment” into log table (id of account, type of action, id of status, id of comment, caring time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Care a comment |

## Uncare a comment

|  |
| --- |
| **Use case:** uncare a comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors click uncare |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id and id of status, id of comment  N2. The system removes relationship “care” comment of the status of actor from database  N3. The system saves the action “uncare a comment” into log table (id of account, type of action, id of status, id of comment, uncaring time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  uncare a comment |

## Comment to status

|  |
| --- |
| **Use case:** comment to status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors comment to a status |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id, id of status, content of comment  N2. The system save id of actor, id of status, content of comment, time to database  N3. The system adds a notification to owner of the status (owner of the comment, id of owner of status, status id, comment id, time)  N4. The system saves the action “comment to status” into log table (id of account, type of action, id of status, id of comment, commenting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New comment to status |

## Comment to a comment

|  |
| --- |
| **Use case:** comment to a comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** Actors comment to a comment |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id, id of status, content of comment, id of comment that commented  N2. The system save id of actor, id of status, content of comment, id of comment that commented, time to database  N3. The system gets statusId of comment that actors comment and add comment-comment notification (actorId, statusId, **id of owner of status**, host comment, newcommentId, time) to *CommentToStatusNotification,* to *Comment\_CommentNotification*  N4. The system saves the action “comment to comment” into log table (id of account, type of action, id of status, id of comment (that is commented), commenting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New comment to comment |

## Share status

|  |
| --- |
| **Use case:** share status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** share status |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id, id of status, id or email of receiver (individuals or companies)  N2. The system save id of actor, id of status, id of receiver, time to database  N3. The system adds sharing notification (id of actor, id of receiver, status id, time)  N4. The system saves the action “share comment” into log table (id of account, type of action, id of status, id of receiver, sharing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Status is shared |

## Post status

|  |
| --- |
| **Use case:** post status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** post status |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id, content of status. Content of status includes:  A link to a website  Text  File:(firstly, provide the directory of file in local computer. Secondly, up the file to server. Then, add new directory to the content of status)  N2. The system save id or email of actors, content of status, time to database  N3. The system saves the action “post status” into log table (id of account, type of action, id of status, posting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New status |

## Delete status

|  |
| --- |
| **Use case:** delete status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors delete status |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of status  ~~N2. The system removes the status, all comments, all relationship (care status, care comment, share,) from database~~  N2. The system sets the status to removing state  N3. The system saves the action “delete status” into log table (id of account, type of action, id of status, deleting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Status is deleted |

## Delete a comment

|  |
| --- |
| **Use case:** delete a comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors delete a comment |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of comment  ~~N2. The system removes comments, all relationship (care comment,) from database~~  N2. The system sets the comment to removing state  N3. The system saves the action “delete comment” into log table (id of account, type of action, id of comment, deleting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  comment is deleted |

## Load recent activities

|  |
| --- |
| **Use case:** load recent activities |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actor logged in to the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id or email  N2. The system load logItems from log table, that are acted by actor based on id or email and logItems are sorted by time. N latest actions will be showed first.  N3. Show logitems to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Logitems |

## Add connection

|  |
| --- |
| **Use case:** add connection |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors want to have a connection with others |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of receiver (who the actors want to connect), inviting message  N2. The system creates a adding connection notification (id of actor, id of receiver, inviting message) to database and wait for answering from receiver |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New connection notifications |

## Notify for adding connections

|  |
| --- |
| **Use case:** Notify for adding connections |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in to the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id  N2. The system load N connection notification item (id of notification, id of connection, id of sender- email of sender, id of target- email of target, id of answer (accept, not accept), state (answered or not answered)) that is not answered based on id of account.  N3. The system shows connection notifications to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Connection notifications are showed to actors |

## Answer connection notifications

|  |
| --- |
| **Use case:** Answer connection notifications |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors answer the connection notifications |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of connection notification, id of connection, id of type of answers (accept or not accept)  N2. The system creates, actives connection (id of actor1, id of actor2, time) to database  N3. The system saves action “accept connection” to log table (id of actor, id of connection, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Connection notifications are showed to actors |

Notes: notification about new status of follower, connection, group, new comment to the status, new shared status

## Notify new messages (from chatting)

|  |
| --- |
| **Use case:** notify about new message |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id,  N2. The system loads message notification (id of notification, id of message, id of sender, id of receiver)  N3. The system shows the message notification to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Message notifications are showed |

## Read notification of new messages (chatting)

|  |
| --- |
| **Use case:** Read notification of new messages (chatting) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Show the conversation to actor (when actor click on notification) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Notify when there is a user who comments on your status

|  |
| --- |
| **Use case:** Notify when there is a user who comments on your status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id  N2. The system bases on ids of status of the actors to get N notifications of new comments of others for the status that the actor haven’t read them yet (id of notification, id of status, id of the user who comment, id of comment, state-read or not read)  N3. The system shows the notifications of new comments to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notifications of new comments are showed |

## Read notification when a user comment on your status

|  |
| --- |
| **Use case:** Read notification when a user comment on your status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Show status containing new comment based on id of status |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Notify when there is a new comment for a comment of your status

|  |
| --- |
| **Use case:** Notify when there is a new comment for a comment of your status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id  N2. The system bases on ids of status of the actors to get N notifications of new comments of others for the comments of the status that the actor haven’t read them yet (id of notification, id of status, id of parental comment, id of the user who comment, id of new comment, state-read or not read)  N3. The system shows the notifications of new comments of the comments to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notifications of new comments of the comments are showed |

## Read notification when there is a new comment for a comment of your status

|  |
| --- |
| **Use case:** Read notification when there is a new comment for a comment of your status |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Show status to actors based on id of status get from notification |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Notify when there is a user who shares a status to you

|  |
| --- |
| **Use case:** Notify when there is a user who shares a status to you |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id  N2. The system bases on the id of actor to get a notification about a status shared to the actor by the other users. (id of notification, id of the sharing user, id of the shared user, id of status, state-read or not read)  N3. The system shows the notifications of sharing status to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notifications of sharing status are showed |

## Read notification when a user shares status to you

|  |
| --- |
| **Use case:** Read notification when a user shares status to you |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Show status that is shared to actors (based on id of shared status) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Notify when there is a new comment for your comment

|  |
| --- |
| **Use case:** Notify when there is a new comment for your comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id  N2. The system bases on their id to get status that contains their comment and get status that have new comments for their status (haven’t read yet) (id of notification, id of actor, id of status, id of comment of actor, id of new comment, time)  N3. The system shows the notifications to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification for new comments of the actor’s comments |

## Read notification when there is a new comment for your comment

|  |
| --- |
| **Use case:** Read notification when there is a new comment for your comment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Show status to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Notify when a group of career is shared to you

|  |
| --- |
| **Use case:** Notify when a group of career is shared to you |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id (id of actors)  N2. The system bases on the id of actor to get a notification about sharing a group of career to the actor (id of notification, id of sender, id of group, id of actor (receiver), answer- accept or not accept)  N3. The system shows the notifications about sharing a group of career to the actor |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notifications about sharing a group of career to the actor are showed |

## Read notification when a group of career is shared to you

|  |
| --- |
| **Use case:** Read notification when a group of career is shared to you |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click button read or click on notification |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification  N2. The system saves the answer (id of notification, type of answer (read already)) to database  N3. Redirect actors to page of group |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification is read |

## Accept for joining to a shared group of career

|  |
| --- |
| **Use case:** Accept for joining to a shared group of career |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors click accept |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id of notification, id of group  N2. The system saves the answer (id of notification, type of answer (read already)) to database (notification has been read)  N3. The system adds the actor to the group  N4. The system saves log for action”join to a group of career” to database (id of actor, type of action, id of group, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-condition**  New member of group of career |

## Send messages (chatting)

|  |
| --- |
| **Use case:** Send messages (chatting) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id (id of actors), id of receiver, content of messages  N2. The system saves message (id of actors, id of receiver, content, time) to database.  N3. The system creates and show a notification to the receiver. |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New messages |

## Receive messages (chatting)

|  |
| --- |
| **Use case:** Receive messages (chatting) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors log in the system |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id (id of actors), id of users who the actor communicate with  N2. The system reads N latest messages in communication  N3. The system displays the messages to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Latest messages are showed |

## Follow an individual, company

|  |
| --- |
| **Use case:** Follow an individual, company |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors want to follow an individual, company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of other users they want to follow  N2. The system creates relationship “follow” (actor (id of actor) follows others (id of other users)) to database  N3. The system saves log of action “follow” to database (id of actor, type of action, id of followed users, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New following |

## Unfollow an individual, company

|  |
| --- |
| **Use case:** unfollow an individual, company |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors want to unfollow an individual, company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of other users they want to unfollow  N2. The system removes relationship “follow” (actor (id of actor) follows others (id of other users)) from database  N3. The system saves log of action “unfollow” to database (id of actor, type of action, id of unfollowed users, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Following is removed |

## Payment

|  |
| --- |
| **Use case:** payment |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors want to proceed a payment |
| **Preconditions:**  Account is authenticated  System is running  Payment services are running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, list of items (id of item, amount)  N2. The system loads details of items (id of item, name of item, unit price) and calculate the total price (including discount) and shows them to actors  N3. The actors choose form of payment and provide information:  -visa card, master card: provides first\_name + last\_name, credit or debit card number, expiration date, security code , country and postal code  -paypal  N4. The system shows information of payment for reviewing:  -visa card, master card: list of item, unit price, amount, price, payment date, first name, last name, credit or debit number, expiration date, security code, country, postal code  -paypal: list of item, unit price, amount, price, payment date  N5. Payment:  -visa card, master card: do payment  -paypal: actors provide email and password, do payment  N6. The system receive confirmation form payment service providers and show notification to actors  N7. The system saves log of action “payment” to database (id of account, type of action, type of payment, id of payment, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Proceeded payment |

## Upgrade

|  |
| --- |
| **Use case:** Upgrade |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** actors decide to upgrade to become a VIP account |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads list of upgrading packages including id of package, name of package, unit price of package, amount  N2. Actors choose just one package that they want by providing id of that package and amount  N3. The system redirect actors to use case payment  N4. Payment is successful. The system set the state of account to VIP and save this state to database  N5. The system saves the log of action “upgrade” to database (id of account, type of action, id of package, amount, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  VIP Account |

# Individuals

## Invite friends from other social networks (Facebook, Google +, Twitter, Linkedin)

|  |
| --- |
| **Use case:** Invite friends from other social networks (Facebook, Google +, Twitter, Linkedin) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors choose a specific social network and provide email and password to sign in the social network  N2. The system get list of email addresses of friends of the actors in the other social networks  N3. Actors choose list of email addresses of their friends who they want to invite  N4. The system sends emails including a link which links to main page of GJOBS to email addresses of friends  N5. The system saves the action “invite friends” into log table (id of account, type of action, email of friend, socialnetwork, inviting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Inviting emails are sent |

## View basic information of other Indiviaduals

|  |
| --- |
| **Use case:** View basic information of other individuals |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide an email or id of other individuals that they want to view  N2. The system loads basic information of the target user including avatar, first name + last name, state of job (working, finding job, studying), current job item ( name of current job , country where they work for, career, name of current company, level of position) name of previous company, name of school (pick the latest school in education), numbers of followers, numbers of connections.  N3. The system shows basic information to the actors  N4. The system saves the action “view basic information of other individuals” into log table (id of account, type of action, id of target account, viewing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Basic information of the target account |

## View full profile of other individuals

|  |
| --- |
| **Use case:** View full profile of other individuals |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide an email or id of other individuals that they want to view  N2. The system checks connection between actors and target individuals. If the target individuals has applied to the employment of the company (current actor), the actor can view full profile of the target as connection even there is no connection between them. Or if the current actor is VIP company, the current actor can view full profile of the individual without any requirement including ignore privacy  N3. The system load privacy of the target individuals  N4. The system loads information based on privacy (if the current actor is VIP company, the filter step based on privacy will be removed):   * **Basic information** of the target user including avatar, first name + last name, state of job (working, finding job, studying), current job item ( name of current job, country where they work for, career, name of current company, level of position) name of previous company, name of school (pick the latest school in education), numbers of followers, numbers of connections; *(always public)* * **Backgrounds** including summary, file cv, experiences (each experience item contains name of job, name of company-include a link to the main page of company, logo of company, starting date- ending date or up to now, level of position, location, description of experiences); (*always public*) * **Skills** (each skill item has its name); (*always public* ) * **Education** (each item will contain name of school, degree, major, starting year-ending year, logo of school); (*always public* ) * **Contact info (**email, birthday, IM-online account, interests, phone, address, marital status**);** (based on privacy public, connections or private ) * **Applied jobs** (name of job, avatar of company, name of company, field of company or career, applied date); (based on privacy public, connections or private ) * **References** (name of users, avatar, current job, name of company), (based on privacy public, connections or private ) * **Connections** (name of users, avatar, current job, name of company), (based on privacy public, connections or private ) * **Groups** (Logo, name of group), (based on privacy public, connections or private ) * **Companies followed** (Logo, name of company, field of company or career), (based on privacy public, connections or private ) * **Professional photos** (file picture) (based on privacy public, connections or private )   N5. The system saves the action “view full profile of other individuals” into log table (id of account, type of action, id of target account, viewing time)  N6. The system show full profile to actors |
| **Alternative course:**  E2-a.There is no connection between actors and target individuals  E2-a.1 Actors need to send a request to view full profile to target individuals and wait for answer. The target need to accept the request if it’s ok for the actors view the full profile |
| **Exceptions:** |
| **Post-conditions:**  Full profile of the target |

## Edit basic information of individuals (owner)

|  |
| --- |
| **Use case:** edit basic information (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine which information (avatar ,last name, first name,state of job (working, finding job, studying), current job ( name of current job , country where they work for, career, name of current company, level of position), name of previous company, name of school) (**exactly is id of edited information**) is edit and new information.  N2. The system validates new values based on id of information  N3. The system saves new values  N4. The system saves the action “edit basic information” into log table (id of account, type of action, id of edited item, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New basic information |

## Edit background of individuals (owner)

|  |
| --- |
| **Use case:** edit background (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine which information (summary, file cv, experience item *(logo ,Name of company, name of job, location, starting date-ending date or up to now, description of the job)*) (**exactly is id of edited information**) is edit and new information.  N2. The system validates new values based on id of information  N3. The system saves new values  N4. The system saves the action “edit background information” into log table (id of account, type of action, id of edited item, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New background information |

## Add new photos to professional photo section

|  |
| --- |
| **Use case:** Add new photo to professional photo section |
| **Actor:** individuals (students, worker) (owner) |
| **Description:** |
| **Trigger:** actors want to add new photo to professional photo section |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors specific directory of the file in local computer and description of the file and location will be used to save in photo store  N2. The system uploads the file to server and saves directory of the file, id of actor, description and uploading time to database  N3. The system creates a status whose content is the uploaded photo  N4.The system saves logItem of action “upload new photo” to database (id of actor, action, name of file, uploading time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New photos in professional photo section |

## Remove a photo from professional photo section

|  |
| --- |
| **Use case:** remove a photo from professional photo section |
| **Actor:** individuals (students, worker) (owner) |
| **Description:** |
| **Trigger:** actors want to remove a photo from professional photo section |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors specific photo id (directory of file in photo store)  N2. The system deletes the specific file from store  N3.The system saves logItem of action “delete a professional photo” to database (id of actor, action, name of file, deleting time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  A removed photo |

## Add new experience items to experiences of background of individuals (owner)

|  |
| --- |
| **Use case:** add new experience items to experiences of background of individuals |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide new experience item (logo, *Name of company, name of job, location, starting date-ending date or up to now, description of the job, level of position*)  N2. The system validates new values  N3. The system saves new values  N4. The system saves the action “add new experience items” into log table (id of account, type of action, id of new experience item, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New experience item is added |

## Remove skills of individuals (owner)

|  |
| --- |
| **Use case:** remove skills (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of skill which will be removed  N2. The system remove the skills based on id from skills of the individuals  N3. The system saves the action “remove skill” into log table (id of account, type of action, id of removed skill, removing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  skills are removed from kills of individuals |

Note: in database, kills of individuals will be stored as a long string. A comma will be used to separate two skills

## Add new skills of individuals (owner)

|  |
| --- |
| **Use case:** add new skills (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of skills  N2. The system create a new id and save the new skill into skills of background of individuals  N3. The system saves the action “add new skill” into log table (id of account, type of action, id of new skill, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  new skills |

## Add new education\_item to education of individuals (owner)

|  |
| --- |
| **Use case:** add new education\_item to education of individuals (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide education\_item (id of school, avatar, name of school, starting year-ending year, degree, major, grade, activities\_societies, description)  N2. The system validates values  N3. The system adds new education\_item to education of individuals  N4. The system saves the action “add new education item” into log table (id of account, type of action, id of new education, adding time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  new education\_item |

## edit education\_item to education of individuals (owner)

|  |
| --- |
| **Use case:** edit education\_item to education of individuals (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of education\_item (id of school, avatar, name of school, starting year-ending year, degree, major, grade, activities\_societies, description) and new values  N2. The system validates values  N3. The system updates education\_item to education of individuals  N4. The system saves the action “edit education item” into log table (id of account, type of action, id of updated education, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Updating education\_item |

## Edit contact info of individuals

|  |
| --- |
| **Use case:** edit contact info of individuals (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide fields (email, birthday, IM, interests, phone, address, marital status) and new values that will be updated  N2. The system validates values  N3. The system updates new contact info  N4. The system saves the action “edit contact info” into log table (id of account, type of action, id of edited item, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Updating contact info |

## Add reference item

|  |
| --- |
| **Use case:** add reference item |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of user who is a reference  N2. The system validates id of reference (exist or not exist)  N3. The system adds the reference to list of references  N4. The system saves the action “add reference” into log table (id of account, type of action, id of reference, time) |
| **Alternative course:**  A2-a.Id of reference is not existed  A2-a.1 notify to actors to modify |
| **Exceptions:** |
| **Post-conditions:**  New reference |

## Remove a reference

|  |
| --- |
| **Use case:** remove reference |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of reference who will be removed from the list of references  N2. The system validates id of reference (exist or not exist)  N3. The system removes the reference from the list of references  N4. The system saves the action “remove reference” into log table (id of account, type of action, id of reference, time) |
| **Alternative course:**  A2-a.Id of reference is not existed  A2-a.1 notify to actors to modify |
| **Exceptions:** |
| **Post-conditions:**  The reference is removed |

## Insert suggested post about companies to list of changes in the wall of individuals

|  |
| --- |
| **Use case:** insert suggested companies to list of changes in the wall of individuals and companies |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** Actors refresh the wall |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system generates the suggested posts  A suggest post for follow:  Company for individual: logo, name of company, content contains introductory paragraph of the company.  N2. Insert the suggested posts into the list of changes of the wall of individuals and companies |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:** |

## Load suggested employments to individuals (worker)

|  |
| --- |
| **Use case:** load suggested employments to individuals |
| **Actor:** individuals ( worker) |
| **Description:** |
| **Trigger:** actors visit any page of their pages |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of actors  N2. The system chooses **N** suggested employments (name of job, logo of company, name of company, location) based on actors (current job, skills, education) to show to actors for applying |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested employments are showed to actors |

## Load suggested groups of careers to individuals (worker)

|  |
| --- |
| **Use case:** load suggested groups of careers to individuals |
| **Actor:** individuals ( worker) |
| **Description:** |
| **Trigger:** actors visit any page of their pages |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of actors  N2. The system chooses **N** suggested groups of career (name of group of career, logo of group, numbers of members) based on actors (current job, education) to show to actors for applying |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested groups of careers are showed to actors |

## Edit privacy for individuals (students, worker)

|  |
| --- |
| **Use case:** Edit privacy |
| **Actor:** individuals (students, worker) (owner) |
| **Description:** |
| **Trigger:** actors want to edit privacy |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors specific item of privacy (provide id of privacy item, id of value of privacy item): (type of privacy: public, connection, private)   * Who can see your contact info * Who can see your applied jobs * Who can see your references * Who can see your connections * Who can see your groups * Who can see companies that you follow * Who can see your professional picture   N2. The system saves new privacy based on id of privacy item and value of privacy item.  N3.The system saves new action “edit privacy” into log table (id of account, action, id of privacy item, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  updated privacy |

## Search (advanced)

|  |
| --- |
| **Use case:** search (advanced) |
| **Actor:** individual (vip) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide searching values of employments that they need (career, type of company, address, name of job of employments, level of position, numbers of experience years in need, name of school, name of major, degree)  N2. The system bases on searching values to search employments of companies |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suitable employments |

## Edit basic information of individuals (after register)

|  |
| --- |
| **Use case:** Edit basic information (after register) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** after register |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads **schools**, **majors**, **degrees**, **nationalities** to display to actors  N2. Actors choose the **name of university**, **major**, **starting year** and **ending year** for studying at that school, **degree**, **nationality**. This is just a sample of education and will be add to education of the account.  N3. The system saves their information to database  N4. The system saves the action “edit basic information-after register” into log table (id of account, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Name of university, major, starting year, ending year, degree and nationality are saved |

## Add an avatar of individuals

|  |
| --- |
| **Use case:** Add an avatar |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide avatar by choosing a file from their computer or get a photo by using their camera  N2. The system saves their avatar to database  N3. The system saves the action “change avatar” into log table (id of account, type of action, changing-avatar time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New avatar |

## Suggest 100 employments based on the last searching values saved

|  |
| --- |
| **Use case:** Suggest 100 employments based on the last searching values saved |
| **Actor:** VIP individual (student, worker) |
| **Description:** |
| **Trigger:** VIP individuals log in successfully |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads searching values of actors based on id of actors  N2. The system search 100 max-match employments (career, type of company, address, name of job of employments, level of position, numbers of experience years in need, name of school, name of major, degree)  N3. The system shows results to suggestion place in profiles of the actors (just show N item- rest items will be showed when actors use use-case see more) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested employments |

## Statistic list of actors viewing your profile of individual

|  |
| --- |
| **Use case:** Statistic list of actors viewing your profile of individual |
| **Actor:** individual (vip)(owner)(student, worker) |
| **Description:** |
| **Trigger:** Actors want to statistic list of actors viewing your profile of individual |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1.The system bases on id of the actors loads list of actors who view profile of the current individual from log table (just gets log of action “View basic information of other individuals”, “View full profile of other individuals”)  N2. The system shows N latest viewer (id of viewer, name of viewer (as link to profile) to actors (rest viewer will be loads in use-case “see more”) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  list of actors viewing a profile of the current individual |

## Advertise the profile of vip individuals to1000 companies

|  |
| --- |
| **Use case:** View full profile of other companies |
| **Actor:** the system |
| **Description:** |
| **Trigger:** after individuals upgrade |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:** |
| **Alternative course:**  N1. The system loads all VIP individuals  N2. For each VIP individuals, the system gets 1000 random companies to introduce the VIP individuals profile (name, degree, experiences) to wall of company |
| **Exceptions:** |
| **Post-conditions:**  Profile of VIP individuals are introduced to companies |

## Introduce 100 companies to VIP individuals for following

|  |
| --- |
| **Use case:** Introduce 100 companies to VIP individuals for following |
| **Actor:** the system (automatically) |
| **Description:** |
| **Trigger:** After individuals upgrade |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The systems loads all current VIP individuals  N2. For each VIP individuals, the system loads 100 companies that has the same career with current job of the current VIP individuals  N3. The system suggests companies to the wall of the current VIP individuals for following |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Companies are suggested to individuals |

# Companies

## View full profile of other companies

|  |
| --- |
| **Use case:** View full profile of other companies |
| **Actor:** individuals (students, workers), companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide an email or id of other companies that they want to view  N2. The system loads information:   * **Basic information** of the target company including logo, name of company, country (location - city) where they work for, type of company,field of company or career, link to website, address, number of applier, numbers of followers, numbers of connections; *(always public)* * **Backgrounds** including summary (*always public*) * **Employments:** each employment item includes name of job, description of job *(always public* ) * **Contact info (**email, birthday, IM-online account-instant message, interests, phone, address, marital status**);** (*based on privacy public, connections or private* ) * **Connections** (name of users, avatar, current job, name of company), (*based on privacy public, connections or private* ) * **Applied candidates (** name of individuals, avatar, current job, name of company**)** *(based on privacy public, connections or private )* * **Groups** (Logo, name of group), *(based on privacy public, connections or private )* * **Followers** (Avartar, name of follower, current job, name of company *)*   N3. The system show full profile to actors,  N4. The system saves the action “view full profile of other” into log table (id of account, type of action, id of target, time) |
| **Alternative course:**  E2-a.There is no connection between actors and target companies  E2-a.1 Actors need to send a request to view full profile to target companies and wait for answer. The target need to accept the request if it’s ok for the actors view the full profile |
| **Exceptions:** |
| **Post-conditions:** |

## Edit basic information of profile of companies (owner)

|  |
| --- |
| **Use case:** edit basic information of profile of companies (owner) |
| **Actor:** companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine which information (logo, name of company, location, field of company, type of company, address, link of website) (**exactly is id of edited information**) is edit and new information.  N2. The system validates new values based on id of information  N3. The system saves new values  N4. The system saves the action “edit basic information of profile of company” into log table (id of account, type of action, id of edited item, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Updating basic information of company |

## Edit the introductory paragraph of background of companies (owner)

|  |
| --- |
| **Use case:** edit the introductory paragraph of background of companies (owner) |
| **Actor:** companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of introductory paragraph and new content  N2. The system saves new introduction  N3. The system saves the action “edit introductory paragraph of background of companies” into log table (id of account, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Updating new introductory paragraph |

## Edit an employment item of employment section of profile of companies (owners)

|  |
| --- |
| **Use case:** edit an employment item of employment section of companies (owner) |
| **Actor:** companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of employment\_item, new content (name of job, level of position, min-salary-> max-salary or deal, address, career, degree, age, appearance, description of job, requirement of job, deadline for application)  N2. The system saves the employment\_item  N3. The system saves the action “edit an employment item of companies” into log table (id of account, type of action,id of employment item, id of edited fields, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Updating employment\_item |

## View an employment in details

|  |
| --- |
| **Use case:** view an employment in details (owner) |
| **Actor:** companies, individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of employment\_item,  N2. The system load content (name of job, level of position, min-salary-> max-salary or deal, address, career, degree, age, appearance, description of job, requirement of job, deadline for application) based on id of employment  N3. The system displayed the employment\_item to actors  N4. The system saves the action “view an employment in details” into log table (id of account, type of action, id of company, id of employment, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Details of an employment |

## Apply for an employment of a company

|  |
| --- |
| **Use case:** apply for an employemnt of a company (owner) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of employment\_item,  N2. The system creates applying item (relationship item) include id of employement\_item, id of actors, applying time  N3. The system saves this item into list of applying item  N4. The system saves the action “apply for an employment of a company” into log table (id of account, type of action, id of company, id of employment, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New applying item |

## Post an employment in profile of company

|  |
| --- |
| **Use case:** post an employment in profile of group |
| **Actor:** companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide (name of job, level of position, min-salary-> max-salary or deal, address, career, degree, age, appearance, description of job, requirement of job, deadline for application)  N2. The system creates an employment\_item in database. The new information of the new record is saved in database  N3. The system saves id of actor, action, time into log table  N4. The system saves the action “post an employment in profile of company” into log table (id of account, type of action, id of company, id of new employment, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New employment |

## Edit privacy for companies

|  |
| --- |
| **Use case:** Edit privacy |
| **Actor:** companies (owner) |
| **Description:** |
| **Trigger:** actors want to edit privacy |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors specific item of privacy (provide id of privacy item, id of value of privacy item): (type of privacy: public, connection, private)   * Who can see your contact info * Who can see your connections * Who can see your applied candidates * Who can see your groups * Who can see your follower   N2. The system saves new privacy based on id of privacy item and value of privacy item.  N3.The system saves new action “edit privacy” into log table (id of account, action, id of privacy item, editing time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  updated privacy |

## Search (advanced)

|  |
| --- |
| **Use case:** search (advanced) |
| **Actor:** company (vip) |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide searching values of profiles (career, level of position, numbers of experience years, name of school, major at school, degree, age, gender, address)  N2. The system bases on searching values to search profiles of individuals  N3. The system saves searching values to database for suggestion in the future |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suitable profiles of individuals |

## Edit basic information of company(after register)

|  |
| --- |
| **Use case:** Edit basic information of company (after register) |
| **Actor:** companies |
| **Description:** |
| **Trigger:** after register |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads name of company (based on registration step), **field of working of companies, type of company**  to display  N2. The actors provides field of working of the company, type of company ,starting year, address, link of website  N3. The system saves their information to database  N4. The system saves the action “edit basic information-after register” into log table (id of account, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Name of university, major, starting year, ending year, degree and nationality are saved |

## Add an logo of comapny (after-resigter)

|  |
| --- |
| **Use case:** Add an avatar of comapny |
| **Actor:** companies |
| **Description:** |
| **Trigger:** |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide logo by choosing a file from their compute  N2. The system saves their logo to database  N3. The system saves the action “change logo” into log table (id of account, type of action, changing-logo time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New logo |

## Suggest 100 profiles of individuals based on the last searching values saved

|  |
| --- |
| **Use case:** Suggest 100 profiles of individuals based on the last searching values saved |
| **Actor:** VIP companies |
| **Description:** |
| **Trigger:** VIP companies log in successfully |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads searching values of actors based on id of actors  N2. The system search 100 max-match profiles of individuals (career, level of position, numbers of experience years, name of school, major at school, degree, age, gender, address)  N3. The system shows results to suggestion place in profiles of the actors (just show N item- rest items will be showed when actors use use-case see more) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested profiles of individuals |

## Statistic list of actors viewing your profile of company

|  |
| --- |
| **Use case:** Statistic list of actors viewing your profile of company |
| **Actor:** company (vip)(owner) |
| **Description:** |
| **Trigger:** Actors want to statistic list of actors viewing your profile of company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1.The system bases on id of the actors loads list of actors who view profile of the current company from log table (just gets log of action “view full profile of other company”  N2. The system shows N latest viewer (id of viewer, name of viewer (as link to profile) to actors (rest viewer will be loads in use-case “see more”) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  list of actors viewing a profile of the current company |

## Advertise VIP companies to 1000 individuals (worker)

|  |
| --- |
| **Use case:** Advertise VIP companies to 1000 individuals (worker) |
| **Actor:** the system (automatically) |
| **Description:** |
| **Trigger:** After companies upgrade |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The systems loads all current VIP companies  N2. For each VIP company, the system loads 1000 random individuals and insert the post (name of company, introductory paragraph of company) to the wall of individuals for following |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Introductory post of company to individuals |

## Introduce employments of VIP company to individuals(worker)

|  |
| --- |
| **Use case:**Introduce employments of VIP company to individuals(worker) |
| **Actor:** the system (automatically) |
| **Description:** |
| **Trigger:** After companies upgrade |
| **Preconditions:**  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The systems loads all current VIP companies  N2. For each VIP company, the system loads current employment (deadline doesn’t meet).  N3. For each of employment of each company, the system chooses 100 profile of individuals (worker) who are suitable to details of employment and suggest the employment to suggesting place of individuals (worker) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Employments are introduced to suitable individuals (worker) |

# Groups

## Comment to status in group (of career, of schools)

|  |
| --- |
| **Use case:** comment to status |
| **Actor:** individuals (member of group or follower) (students, worker), companies (member of group or follower), |
| **Description:** |
| **Trigger:** Actors comment to a status in a group of career |
| **Preconditions:**  Actors are **members or follower** of a group of career  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their email or id, id of status, content of comment  N2. The system saves id of actor, id of status, content of comment, time to database  N3. The system saves id of actor, action, id of status, id of comment, id of group, time to log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New comment to the status in group (of career, of schools) |

## Groups of careers

### Edit profile of group (of career)

|  |
| --- |
| **Use case:** edit profile of group |
| **Actor:** admin, sub-admin, group admin |
| **Description:** |
| **Trigger:** admin, sub-admin edits profiles of group |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, new description of profile of group  N2. The system bases on id of group to update new description of group  N3. The system saves id of actors, action, id of group into log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New description of group |

### Post an employment in group (of career)

|  |
| --- |
| **Use case:** post an employment in group |
| **Actor:** companies |
| **Description:** |
| **Trigger:** companies post an employment in group |
| **Preconditions:**  Actors are member of group  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, id of actors, the employment information (name of job, level of position, min-salary-> max-salary or deal, address, career, degree, age, appearance, description of job, requirement of job, deadline for application)  N2. The system saves employment\_item(id of group, id of actors, name of job, level of position, min-salary-> max-salary or deal, address, career, degree, age, appearance, description of job, requirement of job, deadline for application) into database  N3. The system creates new status based on new employment  N4. The system saves id of actors, action, id of group, id of employment into log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New employment in group |

### Load popular status in group (of career)

|  |
| --- |
| **~~Use case:~~** ~~load popular status in group~~ |
| **~~Actor:~~** ~~Individuals (students, worker), companies~~ |
| **~~Description:~~** |
| **~~Trigger:~~** ~~individuals, companies want to load popular status~~ |
| **~~Preconditions:~~**  ~~Account is authenticated~~  ~~System is running~~  ~~Database is up-to-date, on-line~~ |
| **~~Normal course:~~**  ~~N1. Actors provide id of group~~  ~~N2. The system loads N status (avatar of poster, name of poster, Content, time, numbers of care, numbers of share, numbers of comment, posting time, list of people that commented on the status, comments (avatar of poster, name of poster, content, numbers of care, numbers of share, numbers of comments, posting time, numbers of people that comment to the specific comment)) with the most number of comments.~~  ~~N3. The system saves id of actors, action, id of group, into log table~~ |
| **~~Alternative course:~~** |
| **~~Exceptions:~~** |
| **~~Post-conditions:~~**  ~~Popular status are showed~~ |

### Load list of employments in group (of career)

|  |
| --- |
| **Use case:** load list of employments in in group |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to view list of employments in group |
| **Preconditions:**  Actors are member of group  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads N employments (name of job, name of company, location, posting date, list of members that shared the job)  N3. The system saves id of actors, action, id of group into log table  N4. The system shows the result to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Employments of group are showed |

### Load list of members of group

|  |
| --- |
| **Use case:** load list of member of group |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies enter to the page that contains members of the group |
| **Preconditions:**  Actors are member of group  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system bases on id of group to load list of members (avatar, name, current job, location of company)  N3. The system saves id of actors, action, id of group into log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:** |

### Load status in group of careers

|  |
| --- |
| **Use case:** load status in groups of careers |
| **Actor:** Individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to load status |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads N status (avatar of poster, name of poster, Content, time, numbers of care, numbers of share, numbers of comment, posting time, list of people that commented on the status, comments (avatar of poster, name of poster, content, numbers of care, numbers of share, numbers of comments, posting time, numbers of people that comment to the specific comment)) based on the timeline.  N3. The system saves id of actors, action, id of group, into log table  N4. The system showed result to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  status are showed |

### Save employments in group (of career)

|  |
| --- |
| **Use case:** save employments in in group |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to save employments in group |
| **Preconditions:**  Actors are member of group  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, id of employment  N2. The system add saved employments (jobs) into the list of saved employments of the actors (add saved employment item (id of actor, id of employment, saved time))  N3. The system saves id of actors, action, id of group, id of employment, time into log table  N4. The system shows the result to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New saved employment |

Note: the saved employments will be only showed in the group (not in the wall of actors)

### Share an employment of group (of career)

|  |
| --- |
| **Use case:** share an employment of group |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to share employments in group |
| **Preconditions:**  Actors are member of group  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, id of employment, id of receiver  N2. The system add saved employments (jobs) into the list of shared employments of the actors (add shared employment item (id of actor, id of employment, shared time)) and add shared employment item to wall of receiver  N3. The system saves id of actors, action, id of group, id of employment, time into log table  N4. The system shows the result to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New shared employment |

### Join to a group (group of career)

|  |
| --- |
| **Use case:** join to a group (group of career) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to join to a group |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of actor, id of group  N2. The system add the members to group (save id of actor and id of group to database)  N3. The system saves id of actors, action, id of group , time into log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New member to a group |

### Unjoin a group of career

|  |
| --- |
| **Use case:** unjoin a group of career |
| **Actor:** individuals (students, worker) ( members), companies ( members) |
| **Description:** |
| **Trigger:** actors unjoin a group of career |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of group of career  N2. The system checks whether actors are members of the group  N3. Remove id of actor from list of member of group of career  N4. The system saves log of action “unjoin group of career” to database (id of actor, type of action, id of group, time) |
| **Alternative course:**  A2-a. actors are not members  A2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  Actors leave the group of career |

### Display advertisements in groups of careers

|  |
| --- |
| **Use case:** Display advertisements in groups of careers |
| **Actor:** individuals (member of groups of career) (students, worker), companies (members) |
| **Description:** |
| **Trigger:** Actors enter the groups of careers |
| **Preconditions:**  Actors are members  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id, id of group  N2. The system loads suggested advertisements based on information of actors and id of group  N3. The system loads N suggested advertisements in groups of career (Images of advertisements) and displays to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested advertisements are showed in groups of career |

### Load introduction part of groups of careers when actors visit

|  |
| --- |
| **Use case:** Load introduction part of groups of career |
| **Actor:** individuals (students, worker) (visitors, members), companies (visitor, members), admin, sub-admin |
| **Description:** |
| **Trigger:** actors visit the main page of groups of careers |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads logo of groups of careers, name of groups, numbers of members of group  N3. The system saves id of actors, action, id of group , time into log table  N4. The system shows introduction section to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Introduction section of group is showed |

### Share a group of career to other users

|  |
| --- |
| **Use case:** Share a group of career to other users |
| **Actor:** individuals (students, worker) (visitors, members), companies (visitor, members) |
| **Description:** |
| **Trigger:** actors want to share a group to other users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of group, id of receiver  N2. The system creates a notification item about sharing a group to other users (id of notification, id of actor, id of group, id of receiver, time, type of answer- accept or not accept, is answered or not answered ) to database  N3. The system saves a logItem of the action “sharing a group of career to the other users” to database (id of sender, action, id of receiver, id of group, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New notification about sharing a group to other users |

### Answer to notification inviting to join a group of career

|  |
| --- |
| **Use case:** Answer to notification inviting to join a group of career |
| **Actor:** individuals (students, worker) (visitors, members), companies (visitor, members) |
| **Description:** |
| **Trigger:** actors answer to notifications inviting to join a group of career |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of notifications (inviting to join a group of career), type of answer (join or not join – accept or not accept)  N2. Mark the notification as this is answered  N3. Processing the actor joins to the group or not (use case: join to a group of career) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Notification about inviting to join a group of career is proceed |

## Groups of schools

### View profile of group of school when actors visit

|  |
| --- |
| **Use case:** view profile of group of school |
| **Actor:** individuals (students, worker), companies, admin, sub-admin |
| **Description:** |
| **Trigger:** individuals, companies, admin, sub-admin want to view profile of group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads introduction, homepage (website), phone, address, Leaders of schools, map that show location of school  N3. The system saves id of actors, action, id of group , time into log table  N4. The system shows the profile to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Profile of group of school is showed |

### Edit profile of group of school

|  |
| --- |
| **Use case:** edit profile of group |
| **Actor:** admin, sub-admin, group admin |
| **Description:** |
| **Trigger:** actors want to edit profile of group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, new profile (introduction, homepage (website), phone, address, Leaders of schools, map that show location of school)  N2. The system updates new profile based on field and new value of the field  N3. The system saves log of action “Edit profile of group of school” to database (id of actor, type of action, id of group, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New description of group of school |

### View list of current students of group of school

|  |
| --- |
| **Use case:** view list of current students of group of school |
| **Actor:** individuals(member) (students), companies, admin, sub-admin |
| **Description:** |
| **Trigger:** individuals, companies, admin, sub-admin want to view list of current students of group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads list of current students (name of student, major, type of study, starting year-ending year)(ending year <= now)  N3. The system saves id of actors, action, id of group , time into log table  N4. The system shows list of current students to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Current students of group of school is showed |

### Add students to groups of schools automatically (AFTER completing profile-after register)

|  |
| --- |
| **Use case:** add students to groups of schools automatically (after completing profile-after register) |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** individuals finish the profile after registering |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group, schools  N2. The system add actors to group of schools based on schools provided  N3. The system notifies to the actors about becoming a member of groups of schools  N4. The system saves the action “add new member to groups of schools automatically” into log table (id of member, type of action, id of group, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Actors become a member of groups of schools |

### Follow a group (group of schools)

|  |
| --- |
| **Use case:** follow to a group(group of schools) |
| **Actor:** individuals (students, worker), companies |
| **Description:** |
| **Trigger:** individuals, companies want to following a group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of actor, id of group (group of school)  N2. The system adds the members to group (save id of actor, relationship and id of group to database)  N3. The system saves id of actors, action, id of group , time into log table |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New follower for the groups of school |

### Unfollow a group of school

|  |
| --- |
| **Use case:** Unfollow a group of school |
| **Actor:** individuals (students, worker) ( members), companies ( members) |
| **Description:** |
| **Trigger:** actors unfollow a group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide their id, id of group of school  N2. The system checks whether actors are members of the group  N3. Remove id of actor from list of member of group of school  N4. The system saves log of action “unfollow group of school” to database (id of actor, type of action, id of group, time) |
| **Alternative course:**  A2-a. actors are not members  A2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  Actors don’t follow the group of school |

### View list of ex- students of group of school

|  |
| --- |
| **Use case:** view list of ex-students of group of school |
| **Actor:** individuals (VIP) (students, worker), companies (VIP), admin, sub-admin, group admin |
| **Description:** |
| **Trigger:** actors want to view list of ex-students of group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group  N2. The system loads list of ex-students (name of student, current job, type of degree, starting year-ending year)(ending year > now)  N3. The system saves id of actors, action, id of group , time into log table  N4. The system shows list of ex-students to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Ex-students of group of school is showed |

### Specify yourself as a member of the group of school (individuals)

|  |
| --- |
| **Use case:** comment to status |
| **Actor:** individuals (students, worker) |
| **Description:** |
| **Trigger:** Actors click to specify themselves as a student of this group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors are redirected to a form for education to provide more details  N2. Actors provide id, id of group, value of education item ( id of school, avatar, name of school, starting year-ending year, degree, major, grade, activities\_societies, description)  N3. The system adds new education item to education of actors based on id of the actors  N4. The system saves new information of new member into group (id of actors, id of group, joining time) in database  N5. The system saves id of actor, action, id of group, time to log table. The system saves id of actor, action, time of add new education item to log table. |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New member for a group of school  New education item in education section of actors |

### Shows suggested groups of schools related to the group of school that users are viewing

|  |
| --- |
| **Use case:** show suggested groups of schools in the group of school for actors |
| **Actor:** individuals (member of groups of school or follower) (students, worker), companies (followers) |
| **Description:** |
| **Trigger:** Actors enter the groups of schools |
| **Preconditions:**  Actors are members or follower  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id, id of group  N2. The system loads suggested groups of schools based on information of actors and id of group  N3. The system loads N suggested groups of schools (name of groups of school, location) and displays to actors |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Suggested groups of schools are showed |

# Administration

## Manage school set

### Add new school

|  |
| --- |
| **Use case:** Add new school |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to add new school to the school set |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of school, logo  N2. The system saves school (name, logo) to database  N6. The system saves log of action “add new school” to database (id of actor, type of action, id of school, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New school |

### Remove a school

|  |
| --- |
| **Use case:** Remove a school |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to remove a school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of school from the list of school that is showed  N2. The system check whether there is any object that refer to the career  N3. The system remove the school from the school set  N4. The system saves log of action “remove a school” to database (id of actor, type of action, id of school, time) |
| **Alternative course:**  A2-a. There is an object that refer to the school  A2-a.1 Notify to actor and require deleting the object or not delete the school |
| **Exceptions:** |
| **Post-conditions:**  A school is removed |

### Update a school

|  |
| --- |
| **Use case:** Update a school |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to modify a school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of school, new details (new name)  N2. The system save new details of the school  N6. The system saves log of action “update a school” to database (id of actor, type of action, id of school, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  A school is updated |

## Manage career set

### Add new career

|  |
| --- |
| **Use case:** Add new career |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to add new career to the career set |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of career  N2. The system saves career (name) to database  N6. The system saves log of action “add new career” to database (id of actor, type of action, id of career, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New career |

### Remove a career

|  |
| --- |
| **Use case:** Remove a career |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to remove a career |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of career from the list of career that is showed  N2. The system check whether there is any object that refer to the career  N3. The system remove the career from the career set  N4. The system saves log of action “remove a career” to database (id of actor, type of action, id of career, time) |
| **Alternative course:**  A2-a. There is an object that refer to the career  A2-a.1 Notify to actor and require deleting the object or not delete the career |
| **Exceptions:** |
| **Post-conditions:**  A career is removed |

### Update a career

|  |
| --- |
| **Use case:** Update a career |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** Actors want to modify a career |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of career, new details (new name)  N2. The system save new details of the career  N6. The system saves log of action “update a career” to database (id of actor, type of action, id of career, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  A career is updated |

## Manage groups

### Groups of careers

#### Create a new group of career

|  |
| --- |
| **Use case:** Create a new group of career |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors require creating a new group of career |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of career  N2. The system checks whether a group of the career existed  N3. The system creates a new group of career whose name is name of career  N4. The system saves log of action “create new group of career” to database (id of actor, type of action, id of career, id of group of career, time) |
| **Alternative course:**  A2-a. The group of career existed  A2-a.1 Notify to the actor |
| **Exceptions:** |
| **Post-conditions:**  New group of career |

### Groups of schools

#### Create a new group of school

|  |
| --- |
| **Use case:** Create a new group of school |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors require creating a new group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of school, id of account that will be a group admin (default is id of this admin)  N2. The system checks whether a group of the school existed  N3. The system creates a new group of school whose name is name of school  N4. The system adds the specific account to the member list (if the system admin is also the group admin, the admin will receive no any notification or status from the group of schools)  N5. The system adds the specific account to the group admin list  N6. The system saves log of action “create new group of schools” to database (id of actor, type of action, id of school, id of group of school, time) |
| **Alternative course:**  A2-a. The group of school existed  A2-a.1 Notify to the actor |
| **Exceptions:** |
| **Post-conditions:**  New group of school |

#### Set the group admin for groups of schools

|  |
| --- |
| **Use case:** Set group admin for groups of schools |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors set a member of group of school (delegates of schools) become a group admin to manage the group |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group of school, id of member who will become the group admin  N2. The system add the member to the list of admin of the group of school  N3. The system saves log of action “add new group admins” to database (id of actor, type of action, id of member, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  New group admin |

#### Remove a group admin from the admin list of a group of school

|  |
| --- |
| **Use case:** Remove a group admin from the admin list of a group of school |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors remove a group admin from the admin list of a group of school |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide id of group of school, id of the group admin who will be removed  N2. The system removes the specific group admin from the list of admins of the specific group (the group admin is still a member of the group)  N3. The system saves log of action “remove group admins” to database (id of actor, type of action, id of member, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  A group admin is removed |

## Manage advertisement

### Add an advertisement

|  |
| --- |
| **Use case:** Add an advertisement |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors add new advertisement |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide advertisement (image), location  N2. Check whether there is any advertisement using this location  N3. The system saves advertisement, location to database  N4. The system saves log of action “add new advertisement” to database (id of actor, type of action, id of advertisement, location, time) |
| **Alternative course:**  A2-a.Location is using for other advertisement  A2-a.1 Notify to the actor  A2-a.2 Require removing the advertisement from this location before add new advertisement |
| **Exceptions:** |
| **Post-conditions:**  New advertisement |

### Remove an advertisement

|  |
| --- |
| **Use case:** Remove an advertisement |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors remove new advertisement |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide location  N2. The system removes advertisement at this location  N3. The system saves log of action “remove new advertisement” to database (id of actor, type of action, id of advertisement, location, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  an advertisement is removed |

## Manage types of companies

### Add a type of company

|  |
| --- |
| **Use case:** Add a type of company |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors add a new type of company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of new kind of company  N2. The system validates the name of new kind of company  N3. The system creates a id and saves new kind of company (id, name of new kind of company) to database  N4. The system saves log of action “add new kind of company” to database (id of actor, type of action, id of new kind of company, time) |
| **Alternative course:**  E2-a. name of kind of company has existed  E2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  New sort of company |

### Edit a sort of company

|  |
| --- |
| **Use case:** Edit a sort of company |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors edit a sort of company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of sort of company that will be edited and provide new name of the type of company  N2. The system validates the name of new kind of company  N3. The system saves new name of this type of company (id, new name of kind of company) to database  N4. The system saves log of action “edit a kind of company” to database (id of actor, type of action, id of kind of company, time) |
| **Alternative course:**  E2-a. new name of kind of company has existed  E2-a.1 notify to actors  E2-a.2 require new name |
| **Exceptions:** |
| **Post-conditions:**  Name of sort of company is edited |

### Delete a sort of company

|  |
| --- |
| **Use case:** Delete a sort of company |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors delete a sort of company |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of sort of company that will be deleted  N2. The system validates whether this type of company exist or not exist  N3. The system check to be sure that there is no item contain a reference to this type of company  N4. The system deletes the type of company from database based on id of sort of company  N5. The system saves log of action “delete a kind of company” to database (id of actor, type of action, id of kind of company, time) |
| **Alternative course:**  E2-a. id of the type of company has not existed  E2-a.1 notify to actors  E3-a.There is an item that contain a reference to this type of company  E3-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  A sort of company is deleted |

## Manage level of position

### Add a level of position

|  |
| --- |
| **Use case:** Add a level of position |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors add a new level of position |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of level of position  N2. The system validates the name of level of position  N3. The system creates a id and saves new level of position (id, name of new level of position) to database  N4. The system saves log of action “add new level of position” to database (id of actor, type of action, id of new level of position, time) |
| **Alternative course:**  E2-a. name of level of position has existed  E2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  New level of position |

### Edit a level of position

|  |
| --- |
| **Use case:** Edit a level of position |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors edit a level of position |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of level of position that will be edited and provide new name of the level of position  N2. The system validates the name of new level of position  N3. The system saves new name of this level of position (id, new name of level of position) to database  N4. The system saves log of action “edit a level of position” to database (id of actor, type of action, id of level of position, time) |
| **Alternative course:**  E2-a. new name of level of position has existed  E2-a.1 notify to actors  E2-a.2 require new name |
| **Exceptions:** |
| **Post-conditions:**  Name of level of position is edited |

### Delete a level of position

|  |
| --- |
| **Use case:** Delete a level of position |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors delete a level of position |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of level of position that will be deleted  N2. The system validates whether this level of position exist or not exist  N3. The system check to be sure that there is no item contain a reference to this level of position  N4. The system deletes the level of position from database based on id of level of position  N5. The system saves log of action “delete a level of position” to database (id of actor, type of action, id of level of position, time) |
| **Alternative course:**  E2-a. id of the level of position has not existed  E2-a.1 notify to actors  E3-a.There is an item that contain a reference to this level of position  E3-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  A level of position is deleted |

## Manage degree (of individuals) (Bachelor, Master, PhD,…)

### Add a degree

|  |
| --- |
| **Use case:** Add a degree |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors add a degree |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of degree  N2. The system validates the name of degree  N3. The system creates a id and saves new degree (id, name of new degree) to database  N4. The system saves log of action “add new degree” to database (id of actor, type of action, id of new degree, time) |
| **Alternative course:**  E2-a. name of degree has existed  E2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  New degree |

### Edit a degree

|  |
| --- |
| **Use case:** Edit a degree |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors edit a degree |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of degree that will be edited and provide new name of the degree  N2. The system validates the name of new degree  N3. The system saves new name of this degree (id, new name of degree) to database  N4. The system saves log of action “edit a degree” to database (id of actor, type of action, id of degree, time) |
| **Alternative course:**  E2-a. new name of degree has existed  E2-a.1 notify to actors  E2-a.2 require new name |
| **Exceptions:** |
| **Post-conditions:**  Name of degree is edited |

### Delete a degree

|  |
| --- |
| **Use case:** Delete a degree |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors delete a degree |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of degree that will be deleted  N2. The system validates whether this degree exist or not exist  N3. The system checks to be sure that there is no item contain a reference to this degree  N4. The system deletes the degree from database based on id of degree  N5. The system saves log of action “delete a degree” to database (id of actor, type of action, id of degree, time) |
| **Alternative course:**  E2-a. id of the degree has not existed  E2-a.1 notify to actors  E3-a.There is an item that contain a reference to this degree  E3-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  A degree is deleted |

## Manage working states of individuals

### Add a working state

|  |
| --- |
| **Use case:** Add a working state |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors add a working state |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide name of working state  N2. The system validates the name of working state  N3. The system creates a id and saves new working state(id, name of new working state) to database  N4. The system saves log of action “add new working state” to database (id of actor, type of action, id of new working state, time) |
| **Alternative course:**  E2-a. name of working state has existed  E2-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  New working state |

### Edit a working state

|  |
| --- |
| **Use case:** Edit a working state |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors edit a working state |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of working state that will be edited and provide new name of the working state  N2. The system validates the name of new working state  N3. The system saves new name of this working state(id, new name of working state) to database  N4. The system saves log of action “edit a working state” to database (id of actor, type of action, id of working state, time) |
| **Alternative course:**  E2-a. new name of working state has existed  E2-a.1 notify to actors  E2-a.2 require new name |
| **Exceptions:** |
| **Post-conditions:**  Name of working state is edited |

### Delete a working state

|  |
| --- |
| **Use case:** Delete a working state |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** |
| **Trigger:** actors delete a working state |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors determine id of working state that will be deleted  N2. The system validates whether this working state exist or not exist  N3. The system checks to be sure that there is no item contain a reference to this working state  N4. The system deletes the working state from database based on id of working state  N5. The system saves log of action “delete a working state ” to database (id of actor, type of action, id of working state , time) |
| **Alternative course:**  E2-a. id of the working state has not existed  E2-a.1 notify to actors  E3-a.There is an item that contain a reference to this working state  E3-a.1 notify to actors |
| **Exceptions:** |
| **Post-conditions:**  A working state is deleted |

## Manage users

### View list of users and do statistics for numbers of users

|  |
| --- |
| **Use case:** View list of users |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to view the list of users including active users and inactive users and numbers of users |
| **Trigger:** actors require list of users including active users and inactive users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads lists of users including active users (id of users, email, password, registering time, active state) and inactive users (sorting based on id) and count the numbers of users -(just load N first users-rest of users will be load in use case see more)  N2. The system shows the list of users and numbers of users to actors  N2. The system saves log of action “view list of users ” to database (id of actor, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  List of users and numbers of users |

### View list of active users and do statistics for numbers of active users

|  |
| --- |
| **Use case:** View list of active users |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to view the list of active users and numbers of users |
| **Trigger:** actors require list of active users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The system loads lists of users with active state (id of users, email, password, registering time, active state)(sorting based on id) and count the numbers of active users- (just load N first users-rest of users will be load in use case see more)  N2. The system shows list of active users and numbers of active users to actors  N3. The system saves log of action “view list of active users ” to database (id of actor, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  List of active users and numbers of active users |

## Statistics

### Do statistics about total viewing times in a duration

|  |
| --- |
| **Use case:** do statistics about total viewing times in a duration |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to know how many viewing times there are in the specific duration |
| **Trigger:** actors want to do statistics about total viewing times in the duration |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. The actors provides the duration that they want to do statistics  N2. The system get Log-Item (id of actor, type of action - log in, time) about use case Sign-in of all users in the particular duration (except signing-in of system admin) and count the total viewing times  N3. The system shows the total viewing times to the actors  N4. The system saves log of action “Do statistics about total viewing times in a duration ” to database (id of actor, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  Total viewing times |

### Do statistics about number of active and inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart)

|  |
| --- |
| **Use case:** do statistics about number of active and inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to do statistics related to numbers of users (weekly, monthly, quarterly, half of year, yearly) |
| **Trigger:** actors want to do statistics about numbers of users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide type of statistics (weekly, monthly, quarterly, half of year, yearly) and starting period and ending period, type of chart  N2. The system validates logic between type of statistics and starting period and ending period  N3. The system calculate numbers of time item (weekly, monthly, quarterly, half of year, yearly) between starting and ending period  N4. The system creates a list of users whose creating time is belong to time item for each time item and count the numbers of users for each time item  N5. The system creates the table and the chart for this figures  N6. The system shows the table and the chart to actors  N7. The system saves log of action “Do statistics about number of active and inactive users” to database (id of actor, type of action, time) |
| **Alternative course:**  E2-a. Relation of type of statistics (weekly,…) and starting period and ending period is wrong  E2-a.1 Notify to users |
| **Exceptions:** |
| **Post-conditions:**  Table and the chart for statistics about number of active and inactive users |

### Do statistics about number of active users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart)

|  |
| --- |
| **Use case:** do statistics about number of active users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to do statistics related to numbers of active users (weekly, monthly, quarterly, half of year, yearly) |
| **Trigger:** actors want to do statistics about numbers of active users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide type of statistics (weekly, monthly, quarterly, half of year, yearly) and starting period and ending period, type of chart  N2. The system validates logic between type of statistics (weekly,…) and starting period and ending period  N3. The system calculate numbers of time item (weekly, monthly, quarterly, half of year, yearly) between starting and ending period  N4. The system creates a list of active users whose creating time is belong to time item for each time item and count the numbers of active users for each time item  N5. The system creates the table and the chart for this figures  N6. The system shows the table and the chart to actors  N7. The system saves log of action “Do statistics about number of active users” to database (id of actor, type of action, time) |
| **Alternative course:**  E2-a. Relation of type of statistics (weekly,…) and starting period and ending period is wrong  E2-a.1 Notify to users |
| **Exceptions:** |
| **Post-conditions:**  Table and the chart for statistics about number of active users |

### Do statistics about number of inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart)

|  |
| --- |
| **Use case:** do statistics about number of inactive users (weekly, monthly, quarterly, half of year, yearly) (including the tables and chart – bar chart or line chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** the system admins wants to do statistics related to numbers of inactive users (weekly, monthly, quarterly, half of year, yearly) |
| **Trigger:** actors want to do statistics about numbers of inactive users |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide type of statistics (weekly, monthly, quarterly, half of year, yearly) and starting period and ending period, type of chart  N2. The system validates logic between type of statistics (weekly,…) and starting period and ending period  N3. The system calculate numbers of time item (weekly, monthly, quarterly, half of year, yearly) between starting and ending period  N4. The system creates a list of inactive users whose creating time is belong to time item for each time item and count the numbers of inactive users for each time item  N5. The system creates the table and the chart for this figures  N6. The system shows the table and the chart to actors  N7. The system saves log of action “Do statistics about number of inactive users” to database (id of actor, type of action, time) |
| **Alternative course:**  E2-a. Relation of type of statistics (weekly,…) and starting period and ending period is wrong  E2-a.1 Notify to users |
| **Exceptions:** |
| **Post-conditions:**  Table and the chart for statistics about number of inactive users |

### Do statistics about proportion of numbers of students, individuals finding jobs, workers, companies in a specific period (pie chart)

|  |
| --- |
| **Use case:** Do statistics about proportion of numbers of students, workers, companies in a specific period (pie chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** Do statistics about proportion of numbers of students, workers, companies in a specific period (pie chart) |
| **Trigger:** actors want to make a pie chart for proportion of numbers of students, workers, companies in a specific period |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide a specific period for doing a pie chart  N2. The system gets all list of active users whose creating time is no later than the specific period and restore the state of working (studying, working, finding jobs) of the users based on *state change table* (studying<-> working<->finding a job), clusters users to student cluster, individual finding jobs cluster, worker cluster, company cluster.  N3. The system computes numbers of users in each cluster  N4. The system calculates the ratio of each cluster to the total active users of all clusters  N5. The system makes a pie chart and legend for each cluster  N6. The system shows the pie chart to actors  N7. The system saves log of action “Do statistics about proportion of numbers of students, workers, companies in a specific period” to database (id of actor, type of action, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  The pie chart for statistics about proportion of numbers of students, workers, companies in a specific period |

### Do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart)

|  |
| --- |
| **Use case:** Do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** Actors want to do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart) |
| **Trigger:** actors want to do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart) |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide a specific starting period, ending period, type of chart  N2. The system gets all list of active users whose creating time is no later than the ending period and gets list of follower created between starting period and ending period.  N3. The system computes numbers of followers for each user during starting period and ending period.  N4. The system creates a table, a chart and show to actors  N5. The system saves log of action “Do statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart)” to database (id of actor, type of action, starting period, ending period, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  The chart for statistics about numbers of follower of users (with starting period, ending period) (table, line chart, bar chart) |

### Do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart)

|  |
| --- |
| **Use case:** Do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** Actors want to do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart) |
| **Trigger:** actors want to do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart) |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide a specific starting period, ending period, type of chart  N2. The system gets all list of active users whose creating time is no later than the ending period and gets list of care created between starting period and ending period.  N3. The system computes numbers of care for each user during starting period and ending period.  N4. The system creates a table, a chart and show to actors  N5. The system saves log of action “Do statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart)” to database (id of actor, type of action, starting period, ending period, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  The chart for statistics about numbers of care of users (with starting period, ending period) (table, line chart, bar chart) |

### Do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart)

|  |
| --- |
| **Use case:** Do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** Actors want to do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart) |
| **Trigger:** actors want to do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart) |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide a specific starting period, ending period, type of chart  N2. The system gets all list of active users whose creating time is no later than the ending period and gets list of status created between starting period and ending period.  N3. The system computes numbers of status for each user during starting period and ending period.  N4. The system creates a table, a chart and show to actors  N5. The system saves log of action “Do statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart)” to database (id of actor, type of action, starting period, ending period, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  The chart for statistics about numbers of status of users (with starting period, ending period) (table, line chart, bar chart) |

### Do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart)

|  |
| --- |
| **Use case:** Do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart) |
| **Actor:** admin, sub-admins (system admin) |
| **Description:** Actors want to do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart) |
| **Trigger:** actors want to do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart) |
| **Preconditions:**  Account is authenticated  System is running  Database is up-to-date, on-line |
| **Normal course:**  N1. Actors provide a specific starting period, ending period, type of chart  N2. The system gets all list of active users whose creating time is no later than the ending period and gets list of status created between starting period and ending period N3. For each user, The system gets all log-item “View basic information of other Indiviaduals”, “View full profile of other individuals” (the current actor is individuals (student, worker)), “View full profile of other companies” that target is the current actor and the actions happen between starting period and ending period  N4. The system counts numbers of retrieved logItems  N5. The system creates a table, a chart and show to actors  N6. The system saves log of action “Do statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart)” to database (id of actor, type of action, starting period, ending period, time) |
| **Alternative course:** |
| **Exceptions:** |
| **Post-conditions:**  The chart for statistics about numbers of viewed times of users (with starting period, ending period) (table, line chart, bar chart) |