|  |  |
| --- | --- |
| Use case name: | Sign in |
| Participating  Actors: | initiated by: students |
| The flow of events: | 1. The student enters the website in the sign page. 2. The system shows a form. 3. The student completes the form and chose “create account”. 4. The system check if all fields completed. 5. The system will then compare each field the student enters with the student data it has from the university. 6. If all data are matches the system will add new account and show accept message. |
| Alternative flows: | First alternative flow A1: start at step 4 in the main flow, there is a missing field:  5. the system will show an error message “there is a missing field”.  6. the user will complete the fields and the flow will return to step 3. |
| Exception flows: | First exemption flow E1: start at step 5 in the main flow, there is an unmatched data.  6. the system will show an error message “data is not correct”, and the use case will fail. |
| Entry condition | The system has the student university data. |
| Exit conditions | The student has an account. |

|  |  |
| --- | --- |
| Use case name: | Register a project |
| Participating  Actors: | initiated by: students  supervisor, employee |
| The flow of events: | 1. The student chose a suggestion from the suggestions list and chose “apply”. 2. The system will show a registration form. 3. The student chose the number of his team member, and then enter their university ID, and then chose “apply”. 4. The system will check the registration conditions for all students in this request by the use of the students university data.  * The system will check if the students complete more or equal to 100 hours. * The system will check if students complete the necessary courses (application for junior, junior for senior1, senior1 for senior2). * Finally, the system will check if the all the team members hours are close to each other (the difference less than 7 hours).  1. If all these conditions are true the system will send an accept message and sent this request to all other team members to take their acceptance of the project registration request. 2. the other students will receive the request. 3. If all student accepted this request the system will send the request to the supervisor of this project. 4. The system will enable any student to request for other project or any other students to request using their names. 5. The supervisor will receive the request. 6. If the supervisor accepts this request:  * The system will inform the employee of the new project that ready to register. * The employee will receive the request and register the project on the university system and chose “complete”. * The system will send notification to all team member about the acceptance. * The system will add the project in the page “my project” for all team member with their supervisor. |
| Exception flows: | First exception flow E1: start at step 7 if one of the team members reject the request.  8. the system will delete the request and will not send it the supervisor.  9. the system will send notification to other students to inform them of the reject, and the use case will fail.  Second exception flow E2: start at step 10 from the main flow, if the supervisor rejects the request:  11.The system will delete the request from all students.  12.The system will send a notification of the response.  13.the student can request again for another project, and the use case will fail. |
| Entry condition | The student had logging in |
| Exit conditions | The students had registered a project. |

|  |  |
| --- | --- |
| Use case name: | Delete request |
| Participating  Actors: | initiated by: students |
| The flow of events: | 1. The student chose to delete a request he made for a project. 2. Fist the system will check if all other students accept to send this request the system enable the student from delete this request. 3. If other students did not accept yet the system will ask the student to confirm his decision. 4. The student will confirm his decision. 5. The system will delete the request from the database and from other students pages. 6. The system will send a notification for other students about the updates. |
| Entry condition | The student had logging in  The student had a request. |
| Exit conditions | The request is deleted. |

|  |  |
| --- | --- |
| Use case name | Display registered projects list |
| Participating actors | initiated by all users. |
| Flow of events | 1. The actor selects the “Display registered project List” option from the user interface. 2. The system will display the registered project list. 3. If the user chooses to filter the list. 4. The system will show two options. 5. If the user chooses “by department”. 6. The system will filter the list and display it. 7. If the user chooses “by supervisor”. 8. The system will filter the list and display it. |
| Entry conditions: | user log in |
| Exit conditions: | registered project list displayed |