King Saud University



Collage of Computer Sciences

1436H/1437H - Term 2

Online Examination System

CSC 342 Project

(Phase 1)

|  |  |
| --- | --- |
| Student Name | ID |
| Abdulsalam Alfaisal | 433102180 |
| Abdullah Alshthri | 430105330 |
| Abdulaziz Saleh Al-Akouz | 433103063 |
| Ahmad Alsodari | 431100358 |

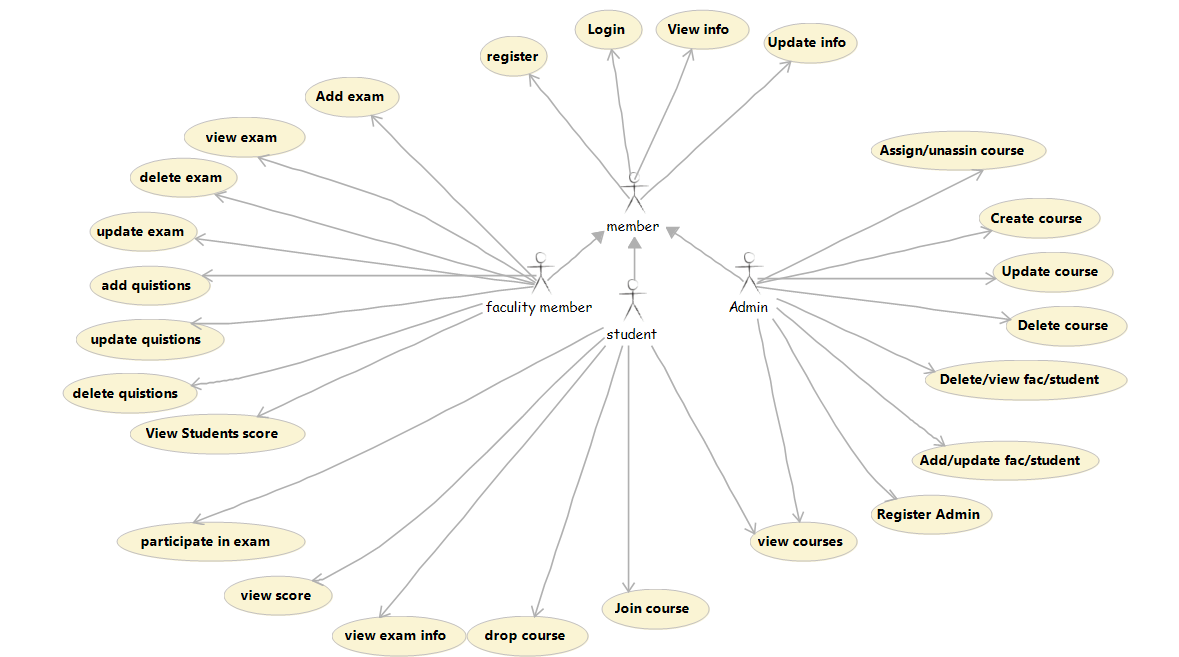
Functional Requirements:

* Members should be able to : Login , Register to the system.
* Members should be able to : View their info , update their info.
* Faculty member should be able to : manage Exams (add,update,view,delete) , manage questions of an exam.
* Faculty Members should also be able to : view students scores on a particular exam.
* Students should be able to : join course ,drop course,view his own courses, view exam’s info(date,time), participate in exam , and view their own score on a particular exam.
* Admins should be able to : Create a course, delete a course, assign faculty to course, unassign faculty to a course.
* Admins should also be able to : add/update faculty members and students.
* Admins should also be able to : Delete/View Faculty members and Students

Non-Functional Requirements:

* System must run on all famous Operation systems (Windows,Linux,Mac).
* System must be running 24 hours a day , 7 days a week.
* System must handle many students and faculty members.

Use Case Diagram:



Use Case Descriptions:

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name:** register | | **UC ID:** 000 |
| **actor:**  faculty member ,student | | **Priority (H,M,L):** H |
| **Goal:** register in system's website | |  |
| * **Input:** user name , id , password and other info. | | |
| * **Steps:** | | |
| **Actor** | **System** | |
| **1. Actor enter** user name .Alt1  **3.** a Alt1  **5.** enter academic ID . Alt1  **7. enter email address. Alt1**  **9. enter captcha litters . Alt1**  **11. click submit button . Alt1** | 2. Sys check if the name chosen is available , display "available" . \*EXP1  4. if password more than 8 litters display "ok" \*EXP2 .  6. check if the ID belong to the institute. Display "verified ". \*EXP3  8. if email format is correct , display "Correct".\*EXP4  10. if captcha is correct , display "Correct" . \*EXP5  12.system perform the required process to complete the registration (update data base ,send varication email , etc) | |
| **Exceptional flows:**  EXP1: user name already used*. Display "user name already used" Go to step 1 .*  EXP*2 : display" use more than 8 liters"*  *Go to step 3.*  EXP*3 :id not verified by institute display "id does not belong to the institute "go to step 5*  EXP4: email format not correct . display "email format not correct"  **Go to step 7**  EXP**5: captcha wrong . display "try again " update captcha , go to 9**  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. |  | |
| **Post-conditions:** *data base will be updated with user info ,* | | |
| * **Output:** confirmation and user information to client by email. | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Login | | **UC ID:** 001 |
| **actor:** admin, faculty member, student | | **Priority (H,M,L):** H |
| **Goal:**  login to system**.** | | |
| Input: username and password | | |
| **Pre-conditions:** user already registered | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.user enter username and password. Alt1  2. user conform information. Alt1 | 3.system check if the username is registered, and check if the password is valid for the username, usecase end. EXP1 | |
| **Exceptional flows:**  EXP1: if the username not found in the system database or entered password not match with username password "username or password not correct, please enter username and password". Goto 1.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** user logged in successfully | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: View info | | **UC ID:** 002 |
| **actor:** faculty member, student | | **Priority (H,M,L):** L |
| **Goal:**  display all user info | | |
| **Pre-conditions:** user logged in | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.user chose to view info .  3. user press back. | 2.system display user info (fullname, ID , Email, other info).  4. system back to system interface. | |
| **Exceptional flows:**  None | | |
| **Post-conditions:** user info displayed | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name** Update info | | **UC ID:**  003 |
| **actor:** faculty member, student | | **Priority (H,M,L):** L |
| **Goal:**  update user info | | |
| **Pre-conditions:** user viewed info | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.user press update info .Alt1  3. user enter info . Alt1  5. use press back. Alt1 | 2.system ask to enter (fullname, other info).EXP1  4. system save the info and display “info updated”  6. system back to system interface | |
| **Exceptional flows:**  EXP1: user pressed back . go to 5  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** user info updated | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name** Add exam | | **UC ID:** 004 |
| **actor:** faculty member | | **Priority (H,M,L):** M |
| **Goal:**  to add exam | | |
| **Pre-conditions:** user assigned to course | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.user press add exam. Alt1  3.user chose course. Alt1  5. user enter date and time. Alt1 | 2.system display list of courses assigned to user. EXP1  4. system ask for date and time.  6. system check for date and time, generate new exam with exam ID, and upload to database.EXP2 | |
| **Exceptional flows:**  EXP1: user not assigned to course . display “no course assigned to”. go to system interface  EXP2: if exam date is during a weekend or the time entered is from 12AM to 7AM , display “ invalid date or time” go to 5  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** exam added | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name** view exam | | **UC ID:** 005 |
| **actor:** faculty member | | **Priority (H,M,L):** L |
| **Goal:**  to view exam info | | |
| **Pre-conditions:** user added exam | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.user press view exam. Alt1  3.user chose exam. Alt1  5. user press exit. Alt1 | 2.system display list of exams added by the user. EXP1  4. system display exam info(Date,Time,Questions,…etc).  6. system back to system interface | |
| **Exceptional flows:**  EXP1: no exams added by the user , display “ no exams” go to system interface  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** exam info viewed | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name:** Delete Exam | | **UC ID:** 06 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L): M** |
| **Stakeholders:***Admin* | |  |
| **Goal:**  **To make the faculty or admin to be able to delete any exam on the system.** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam ID** | | |
| **Pre-conditions:**   * *Exam ID avaliable* | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor inputs the ID of the exam that need to be deleted. \*ALT1**  **3. actor press confirm button. \*ALT2** | **2. System checks database and if exam found , and check that exam created by this actor ,display exam info , and confirmation message. \*EXP1**  **4. system run the appropriate processes to delete the exam, display finish message.**  **\*Exp2**  **5. Exit Delete Exam form.** | |
| **Exceptional flows:**  **Exp1: exam not found , go to step 1.**  **Exp2: an error accrued while attempting to delete exam , display error message then go to step 1.**  **Alterative Flows:**  **Alt1: actor can abort the operation, go to step 5.**  **Alt2: actor can press cancel button, go to step 1.** | | |
| **Post-conditions:** *delete the selected exam from DB.* | | |
| **Output: successful or failed message.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name: Update Exam** | | **UC ID:** 07 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L):** |
| **Stakeholders:***Admin* | |  |
| **Goal:** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam ID, Exam Info (name,time,date,..etc)** | | |
| **Pre-conditions:**   * *Exam ID is available* * *New info available* | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor enter exam ID. \*Alt1**  **3. actor enter the new info of the exam.**  **5. actor press Confirm button. Alt2** | **2. system check if exam exist, and created by this actor. If found display exam info, then ask actor to enter new info. \*Exp1**  **4. system display confirmation message.**  **6. system run the necessary processes to update exam info. If succeeded display ‘Done’ message. \*Exp2**  **7. Exit Update Exam Form.** | |
| **Exceptional flows:**  **Exp1: if exam is Not found, display “exam not found” ,then go to step 1.**  **Exp2: if an error accrued while updating the exam, display error message , then go to step 1.**  **Alternative Flows:**  **Alt1: actor can abort the operation , go to step 7.**  **Alt2: actor can press cancel button, go to step 1.** | | |
| **Post-conditions:** *update Exam Info in the DB.* | | |
| **Output: return ‘Successful” or ‘Failed’ message.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name:** Add Questions | | **UC ID:** 08 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L): M** |
| **Stakeholders:** | |  |
| **Goal:**  **Add qurstion(s) to a specific exam** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam ID , Question and answer** | | |
| **Pre-conditions:**   * **Logged in as a Faculty Member.** * **Exam is available.** * **Question(s) Prepared.** | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor enters Exam ID \*Alt1**  **3. actor enters the question and the multiple choices specifying the correct one. \*Alt1** | **2. system checks the Exam ID and created by this actor. If available display “available”, then ask actor for question and multiple choice answer. \*Exp1**  **4. system run the appropriate processes to add the question. If succeeded , display “Succeeded”. Go to step 3. \*Exp2**  **5. exit the use case.** | |
| **Exceptional flows:**  **Exp1: if exam not found , display “Exam not found”, then go to step 1.**  **Exp2: if an error accured , display the error, then go to step 3.**  **Alternative Flows:**  **Alt1: actor can cancel the operation. Go to step 5.** | | |
| **Post-conditions:** *Add question(s) to the specified exam.* | | |
| **Output: “succeeded” or “failed” message.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name:** Update Question | | **UC ID:** 09 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L): M** |
| **Stakeholders:** | |  |
| **Goal: to update the question(s) of a specific exam.** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam Id, Updated question(s)** | | |
| **Pre-conditions:**   * **Logged in as a Faculty Member.** * **Exam available.** * **Question(s) ready.** | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor enters Exam ID. \*Alt1**  **4. actor select the question need to be updated. \*Alt1**  **6. actor enter the updated question and choices. \*Alt1** | **2. system checks the Exam ID and created by this actor, if found go to next step. \*Exp1**  **3. display the questions of the exam, and ask actor to select the question need to be updated.**  **5. system ask actor to enter the updated question and choices.**  **7. system run the appropriate processes to update the question. If succeeded system displays “Succeeded”, then go to step 3. \*Exp2**  **8. exit the use case.** | |
| **Exceptional flows:**  **Exp1: if exam is not found, display “exam not found”, then go to step 1.**  **Exp2: if an error accrued, display the error, then go to step 6.**  **Alternative Flows:**  **Alt1: actor can cancel the operation, go to step 8.** | | |
| **Post-conditions:** *update the question(s) of the specified exam.* | | |
| **Output: “succeeded” or “failed” message.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name:** Delete Question | | **UC ID:** 10 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L): M** |
| **Stakeholders:** | |  |
| **Goal: to delete the question(s) of a specific exam.** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam Id.** | | |
| **Pre-conditions:**   * **Logged in as a Faculty Member.** * **Exam available.** * **Question(s) Known.** | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor enters Exam ID. \*Alt1**  **4. actor select the question need to be deleted. \*Alt1** | **2. system checks the Exam ID and created by this actor, if found go to next step. \*Exp1**  **3. display the questions of the exam, and ask actor to select the question need to be deleted.**  **5. system run the appropriate processes to update the question. If succeeded system displays “Succeeded”, then go to step 3. \*Exp2**  **6. exit the use case.** | |
| **Exceptional flows:**  **Exp1: if exam is not found, display “exam not found”, then go to step 1.**  **Exp2: if an error accrued, display the error, then go to step 6.**  **Alternative Flows:**  **Alt1: actor can cancel the operation, go to step 6.** | | |
| **Post-conditions:** *delete the question(s) of the specified exam.* | | |
| **Output: “succeeded” or “failed” message.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System: Online Examination System** | | **Student Group ID:** |
| **Use Case name:** View Students Scores | | **UC ID:** 11 |
| **Primary actor:** *Faculty Member* | | **Priority (H,M,L):** |
| **Stakeholders:** | |  |
| **Goal: to view the scores of the students of an exam.** | | |
| **Relationships**   * **Includes:** * **Extends:** | | |
| **Input: Exam ID.** | | |
| **Pre-conditions:**   * **Logged in as a Faculty Member.** * **Exam is available** * **Exam has been conducted (students took the exam).** | | |
| **Steps:** | | |
| **Actor** | **System** | |
| **1. actor enters the Exam ID. \*Alt1** | **2. system checks the Exam ID and exam created by this actor, if found and exam been already conducted go to next step. \*Exp1**  **3. display list of the students took the exam and there scores.**  **4. exit the use case.** | |
| **Exceptional flows:**  **Exp1: if exam not found or has not been conducted yet , display “Exam not found” or “exam not conducted”, then go to step 1.**  **Alternative Flows:**  **Alt1: actor can cancel the operation, go to step 4.** | | |
| **Post-conditions:** | | |
| **Output:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Students Names:** |
| **Use Case name:** join course | | **UC ID:** 012 |
| **Primary actor:** student | | **Priority (H,M,L):** M |
| **Goal:** join a course | | |
| **Input:** course name | | |
| **Pre-conditions:**course is avaliable | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter electronic gate . Alt1 2. student select a course name. Alt1 3. student click on submit. Alt1 | 2.1 system display information course  Exp 2.1.1  3.1 system add the course | |
| **Exceptional flows**:  2.1.1 the course is not avaliable , go to step [2]  2.1.2 the course Inconsistent with another course  a. System offers Change the courses  a1. student accept  a2. Resume  b1.student refuses  b2. Fail  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** the course is added | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Students Names:** |
| **Use Case name:** drop course | | **UC ID:** 013 |
| **Primary actor:** student | | **Priority (H,M,L):** M |
| **Goal:** drop a course | | |
| **Input:** course name | | |
| **Pre-conditions:** corse is added | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter system interface. Alt1 2. student select a course name. Alt1 3. student click on submit. Alt1 | 1.1 system display list of course  Exp 1.1.1  2.1 system display course information  Exp 2.1.1  3.1 system drop the course | |
| **Exceptional flows:**   * + 1. system can not drop the course , go to step [1]   **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** course is dropped | | |
| **Output:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Students Names:** |
| **Use Case name:** View Courses | | **UC ID:**  014 |
| **Primary actor:** student | | **Priority (H,M,L):**M |
| **Goal:** view courses | | |
| **Pre-conditions:** Student view the courses | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter system interface. Alt1 2. student select a course name. Alt1 3. student click on submit. Alt1 | 1.1 system display list of course  Exp 1.1.1  2.1 system display course information  3.1system back to system interface | |
| **Exceptional flows:**   * + 1. course is not avaliable , return to step [1]   **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** courses is viewed | | |
| **Output:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name** view exam information | | **UC ID:**  015 |
| **actor:** Student | | **Priority (H,M,L):** H |
| **Goal:**  to view exam info | | |
| **Pre-conditions:** Exam is avaliable | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter system interface. Alt1 2. student select a course. Alt1 3. student select a Exam. Alt1 4. student click on Back. Alt1 | 1.1 display courses that the student joined.  2.1 system display list of exams  . EXP1  3.1 system display exam info(Date,Time).  4.1 system back to system interface | |
| **Exceptional flows:**  EXP1: Exam is not avaliable , go to step [2]  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** exam info viewed | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name:** Participate in Exam | | **UC ID:** 016 |
| **actor:** Student | | **Priority (H,M,L):** H |
| **Goal:**  Involve in the exam | | |
| **Pre-conditions:** Exam is avaliable | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter system interface. Alt1 2. student select a course. Alt1 3. select participate in exam Alt1 4. student click confirm Alt1 5. student start to solve the exam 6. student click on submit , or time is finished. Alt1 | 1.1 system display list of courses that the student joined.  3.1 system display exam info ( date,time ) EXP1  4.1 system Add you in the exam  4.2 student enter exam.  6.1 calculate and upload score | |
| **Exceptional flows:**  EXP1: Exam is not avaliable , go to step [2]  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** the Exam is involved | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name:** View Score | | **UC ID:** 017 |
| **actor:** Student | | **Priority (H,M,L):** M |
| **Goal:**  See the exam scoure | | |
| **Pre-conditions:** Done from The Exam | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1. student enter system interface. Alt1 2. student select a Exam. Alt1 3. student click on submit. Alt1 | 1.1 system display list of exams  . EXP1  2.1 system display exam info ( Scoure )  3.1 system Back to System interface | |
| **Exceptional flows:**  EXP1: Exam is not avaliable , go to step [1]  EXP2: student was not tested , go to step [1]  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** Scoure is viewed | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name:** register admin | | **UC ID:**  018 |
| **actor:**  admin | | **Priority (H,M,L):** H |
| **Goal:** register in system's website | |  |
| * **Input:** user name , id , password and other info. | | |
| * **Steps:** | | |
| **Actor** | **System** | |
| 1. Enter Ticket Number , given by an admin. \*ALT1  **3. Actor enter** user name \*alt1  **5.** enter password . \*alt1  **7. enter email address. \*alt1**  **9. enter captcha litters . \*alt1**  **11. click submit button \*alt1** | 2. check if Ticket Number is valid, if valid display “valid” , and proceed to next step. \*Exp0  4. Sys check if the name chosen is available , display "available" . \*EXP1  6. if password more than 8 litters display "ok" \*EXP2 .  8. if email format is correct , display "Correct".\*EXP3  10. if captcha is correct , display "Correct" . \*EXP4  12.system perform the required process to complete the registration (update data base ,send varication email , etc) | |
| **Exceptional flows:**  **Exp0: Ticket No. is not valid , go to step 1**  EXP1: user name already used. Display "user name already used" Go to step 1 .  EXP2 : display" use more than 8 liters"  Go to step 3.  EXP3: email format not correct . display "email format not correct"  Go to step 5  EXP4: captcha wrong . display "try again " update captcha , go to 7  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. |  | |
| **Post-conditions:** *data base will be updated with user info ,* | | |
| * **Output:** confirmation and user information to client by email. | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Assign/Unassign Course to faculty. | | **UC ID:** 019 |
| **actor:** admin | | **Priority (H,M,L):** L |
| **Stakeholders:** administration | |  |
| **Goal:** assign or unassign course to faculty | | |
| Input: name of the course , course number , faculity name | | |
| **Pre-conditions:** the course that need toassign or unassign created previously | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin enter the course name . Alt1  2.admin choose assign course or unassign course . Alt1  3.admin enter the faculty member ID. Alt1 | 1.1. System check if the course available display "available" . \*EXP1  3.1. system doesn't found the faculty. \*EXP2 | |
| **Exceptional flows:**  EXP1: not found course or wrong spell. Display "course is not found, create course first or try again" goto 1.  EXP2: enter the faculty ID again display "enter faculty ID correctly". Goto 3.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** data base will be updated with the changes | | |
| **Output:** send the changes info to faculty email**.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Create Course | | **UC ID:** 020 |
| **actor:** admin | | **Priority (H,M,L):** M |
| **Goal:**  create course by name and number**.** | | |
| Input: name of the course , course number ,prerequisite course, number of credit hours . | | |
| **Pre-conditions:** the course that need tocreate is not created previously | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin enters course name. Alt1  2.admin enters the course number. Alt1  3.admin assign prerequisite course by writing its name and number. Alt1  4.admin enter the credit hour to the creating course. Alt1 | 1.1system check if the chosen name available. EXP1.  3.1. system search about the assigned course then assign it to the creating course as prerequisite course. EXP2.  4.system assign the credit to the creating course. | |
| **Exceptional flows:**  EXP1: if the chosen name used previously display "chosen name used try another course try again". Goto 1.  EXP2: if the written prerequisite course name not found .display "prerequisite course not found, try again". Goto 3.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** data base will be updated with the changes | | |
| **Output:** send the changes info to admin email**.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Update Course | | **UC ID:** 021 |
| **actor:** admin | | **Priority (H,M,L):** L |
| **Goal:**  update course info. | | |
| Input: name of the course , course number ,prerequisite course, number of credit hours. | | |
| **Pre-conditions:** the course that need toupdate is created previously | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin enters course name that needs to update. Alt1  2.admin update the course number.  3.admin update the assigned . prerequisite course by writing the new course name and number. Alt1  4.admin update the credit hour to the creating course. Alt1 | 1.1system check if the chosen name available. EXP1.  3.1. system search about the assigned course then assign it to the creating course as prerequisite course. EXP2.  4.system assign the credit to the creating course. | |
| **Exceptional flows:**  EXP1: if the chosen name used previously display "chosen name used try another course try again". Goto 1.  EXP2: if the written prerequisite course name not found .display "prerequisite course not found, try again". Goto 3.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** data base will be updated with the new changes | | |
| **Output:** send the changes info to admin email**.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Add/Update Fac,Student | | **UC ID:** 022 |
| **actor:** admin | | **Priority (H,M,L):** L |
| **Goal:**  to add or update faculty or student to the system. | | |
| Input: ID of the new faculty or student and | | |
| **Pre-conditions:** the student or faculty that need toupdate is created previously. | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin choose between student or faculty. Alt1  2.admin choose between add or update. Alt1  3.if update: admin enter the Fac,STD ID need to update. if add: admin enter the new faculty or student info.  Alt1 | 1.1. system deal with the information as what chosen (student or faculty).  2.2. system deal with the information as what chosen (add or update).  3.3. system search about the ID if the ID found and update chosen, then you can update EXP1. If ID not found and "add" chosen, then you can add. EXP2.  3.4.system generate default user and password for the new member (student or faculty). | |
| **Exceptional flows:**  EXP1.search ID not found . display "enter correct ID".goto 3.  EXP2. Entered a used ID display "ID used, try another ID".goto3.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** data base will be updated with the new changes | | |
| **Output:** send the changes info to admin email**.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Delete/View Fac,Student | | **UC ID:**  023 |
| **actor:** admin | | **Priority (H,M,L):** L |
| **Goal:**  to add or update faculty or student to the system. | | |
| Input: ID of the faculty or student, delete or view selection. | | |
| **Pre-conditions:** the student or faculty that need toview or delete is created previously. | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin choose between student or faculty. Alt1  2.admin choose between view or delete. Alt1  3.if view: admin enter the Fac,Std ID want to view. if delete: admin enter the ID want to delete. Alt1 | 1.1. system deal with the information as what chosen (student or faculty).  2.2. system deal with the information as what chosen (view or delete).  3.3. system search about the ID if the ID found and view chosen, then system display the information. If ID found and "delete" chosen, then system delete the information from database. EXP1. | |
| **Exceptional flows:**  EXP1.search ID not found . display "enter correct ID".goto 3.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** data base will be updated with the new changes. | | |
| **Output:** send the changes info to admin email**.** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Description** | | |
| **System:** online examination system | | **Student Group ID:** |
| **Use Case name**: Delete Course | | **UC ID:** 024 |
| **actor:** admin | | **Priority (H,M,L):** L |
| **Goal:**  delete course. | | |
| Input: name of the course . | | |
| **Pre-conditions:** the course is created previously | | |
| **Steps:** | | |
| **Actor** | **System** | |
| 1.admin enters course name that needs to be deleted. Alt1  2.admin conform. Alt1 | 1.1system check if the chosen name available. EXP1.  2.1. system delete the course. EXP2. | |
| **Exceptional flows:**  EXP1: if the course not exist display "course not available ". Goto 1.  EXP2: if the system cant delete the course .display "error while deleting". Goto 1.  **Alternative flows:**  Alt1: actor can cancel Operation , exit use case. | | |
| **Post-conditions:** course will be deleted | | |
| **Output:** send the changes info to admin email**.** | | |

Class Diagram:

