*Use Case: Manage Universities (Add)*

*Inputs:*

*• University Name*

*• University ID*

*• Location*

*• Contact Info (Phone number or email)*

*• Announcement Info (deadline, program)*

*• Degree Name*

*• Degree ID*

*Output:*

*1-Confirmation Messages.*

*2-System adds new university in the list.*

*Primary Actor: Administrator*

*Stakeholders and Interest:*

***1- Administrator:***

*Wants to add universities in the system accurately and efficiently.*

***2-Universities:***

*Wants to display their universities correct information in the system.*

***3-Teacher (Our project stakeholder):*** *Wants all the features*

*implemented correctly for this manage universities and*

*every feature is user-friendly.*

***4-Students:***

*Wants correct information display about universities for their decision-making process.*

*Pre-conditions:*

*• Administrator is logged in.*

*• The “Add university” feature is available on admin system interface.*

*Post-Conditions:*

*• The new university is added to the system.*

*• Added university is visible to the students.*

*Main Success Scenario:*

• The Administrator navigates to the “Manage Universities” section.

• The Administrator selects the “Add university” option.

• System asks the Administrator to fill in the details of university.

• Administrator also adds the degree-levels of that university one-by-one.

• Administrator also adds announcements related to specific degree level.

• System validates the details when Administrator submits the details.

• System adds a new university in pre-existing list after validation.

• System sends a confirmation message to the Administrator of successfully adding new university.

*Alternative Flows:*

*\*a. Any time system fails:*

*To support recovery and maintenance of the system,*

*1- System signals an error message to the administrator in case*

*of any errors.*

*2- System ensures all the draft changes are saved properly.*

*3- System reconstructs prior state after the failure diminished.*

*4a- Invalid Details:*

*If the details are invalid or incomplete, system*

*displays an error message to the Administrator.*

*4b- Adding pre-existing university:*

*If the university entered is already present on the system, system displays an error message to Administrator.*

*5a-New University adding errors:*

*If system fails to add the new university into the system, it must display*

*an error message of something went wrong.*

*Special Requirements:*

*1- Text must be visible from 1 meter.*

*2- Robust Recovery must be assured when data fetching,*

*updating data or rendering data errors occur.*

*3- System must provide back-up of system’s data if system slows*

*down or fails suddenly.*

*Frequency Of Occurrence:*

*Could occur only if needed not frequently.*

Use Case: Manage Administrator Access

Inputs:

1-Full New Admin Name

2-New Admin Email Address

3-New Admin Unique Username ID

4-New Admin Password (Auto generated or manually entered)

Output:

1-Confirmation Messages.

2-Email sent to new Administrator with his/her login credentials.

Primary Actor: System Administrator

Stakeholders and Interest:

**1-System Administrator:**

Wants to manage administrator access fully privately, only those new administrators which system administrator gives access can manipulate and confirm changes in the system.

**2-Universities:**

Wants a secure website and authorized administrators, who make changes in the universities data accurately and efficiently.

***3-Teacher (Our project stakeholder):*** *Wants all the features*

*implemented correctly for this manage admin access and*

*every feature is user-friendly.*

Pre-conditions:

1-System Administrator is logged in.

Post-Conditions:

1-Email is sent to new Administrator with login credentials

2-New administrator is added into the system.

Main Success Scenario:

1-System Administrator navigates to “Manage Admin Access” section.

2- System Administrator selects the option of “Add new Administrator”

3-System asks the system Administrator to fill in details of new Administrator.

4-System Administrator submits the form.

5-System sends an email to the new Administrator with login credentials.

6-System sends a confirmation message to the system administrator that email is sent.

7-System stores unique Id and password in the system.

8-System updates the list of administrators in the system, adding new Administrator.

9-System sends a confirmation message to the system administrator that new admin is successfully added.

Alternative Flows:

*\*a. Any time system fails:*

*To support recovery and maintenance of the system,*

*1- System signals an error message to the administrator in case*

*of any errors.*

*2- System ensures all the draft changes are saved properly.*

*3- System reconstructs prior state after the failure diminished.*

*3a- Invalid Email Address:*

*If the email format is incorrect and admin already existed, system displays an error message to the Administrator.*

*5a- Email sending failure:*

*If the system fails to send email to new admin, it should display an error message of something went wrong to system admin.*

*8a-New Admin adding errors:*

*If system fails to add the new admin into the system, it must display an error message of something went wrong.*

Special Requirements:

*1- Text must be visible from 1 meter.*

*2- Robust Recovery must be assured when data fetching,*

*updating data or rendering data errors occur.*

*3- System must provide back-up of system’s data if system slows*

*down or fails suddenly.*

Technology and Data Variations list:

*1- System allows manually keyboard passwords or also gives option of automatic password generator.*

Frequency Of Occurrence:

*Could be Nearly Continuous.*

Use Case: Manage Programs (Delete)

Inputs:

Program name

Program ID

Specified University (To delete program)

Output:

1-Confirmation Messages.

2-System deletes program from the selected university.

Primary Actor: System Administrator

Stakeholders and Interest:

**1- Administrator:**

Wants to delete programs from the specified universities in the system accurately and efficiently.

**2-Universities:**

Wants to display programs of their universities correctly in the system and if the program is removed from that university, they will want quick and accurate removal of that program from system(website) too.

**3-Teacher (Our project stakeholder):**

Wants all the features implemented correctly for this manage programs and

every feature is user-friendly.

**4-Students:**

Wants correct information display about programs for their decision-making process.

Pre-conditions:

• Administrator is logged in.

• The “Delete Program” feature is already available on admin system interface.

Post-Conditions:

• The specified program is deleted from the system.

Main Success Scenario:

• The Administrator navigates to the “Manage Programs” section.

• The Administrator selects the specific university and view the list of programs in it.

• The Administrator selects the desired program.

• System gives the option to delete the program.

• The Administrator selects the option to delete.

• System validates or confirms from Administrator to delete that program.

• Administrator confirms the deletion.

• System deletes that program from that university.

• System displays a confirmation message that program is deleted successfully.

Alternative Flows:

\*a. Any time system fails:

To support recovery and maintenance of the system,

1- System signals an error message to the administrator in case

of any errors.

2- System ensures all the draft changes are saved properly.

3- System reconstructs prior state after the failure diminished.

8a- Deletion fails:

If the deletion fails, system must display message of something went wrong.

Special Requirements:

1- Text must be visible from 1 meter.

2- Robust Recovery must be assured when data fetching,

updating data or rendering data errors occur.

3- System must provide back-up of system’s data if system slows

down or fails suddenly.

Frequency Of Occurrence:

Could occur only if needed not frequently.