**EXPERIMENT 4**

**AIM:** To write Use case description for the case study “University Registration System”.

**DESCRIPTION:**

Use case description is the detailed explanation of all the use cases defined in the use case diagram. It defines the flow and function of the use case: when will the use case be active, when will it end, what are the necessary conditions to be fulfilled for its functioning.

The following is the template for use case description:

1. **Introduction**: Briefly describe the use case.
2. **Actors**: List actors that participate and interact with this use case.
3. **Pre-Condition:** List pre-conditions here. List the system state/conditions which must be true before this Use Case can be executed.
4. **Post-Conditions:** Post-conditions on a use case lists possible states that the system can be in at the end of the use case execution. The system must be in one of those states. A post-condition also states actions that the system performs at the end of the use case, regardless of what occurred in the use case.
5. **Flow of events:**

Basic flow: primary events on use case execution

Alternate flow: Any other possible flow.

1. **Special Requirements:** Enter any special requirements such as Performance requirements, Security requirements, and user interface requirements.
2. **Related use cases:** List the related use cases if any.

The use case description should be in a box.

**STEPS TO BE FOLLOWED:**

1. Determine all the use cases.
2. Take each use case one by one and determine their functionalities and constraints.
3. Determine all the requirements for each use case.
4. Write the description for each use case in the given standard format.
5. Build a box around all the use case descriptions.

**RESULT:**

**1. REGISTER**

|  |
| --- |
| **Introduction :** The use case documents the steps that must be followed in order to register into the system. |
| **Actors** Administrator, Student, Faculty |
| **Pre-Conditions :** None |
| **Post Conditions**: If the use case is successful, the administrator/ student/ faculty are registered into the system else the system state remains unchanged. |
| **Flow of Events**  **Basic Flow**   1. The system requests that the Student/Faculty enter his name, email address, address, password and contact number and the role of the user (Student/Faculty).   2. The Student/Faculty enters the required information  3. The system validates the entered credentials.  4. A request for an account creation is sent to the Administrator  5. The Administrator reviews the request and after verification of credentials, creates an account with the provided information and sends a confirmation to the customer. |
| **Alternative Flow 1: Invalid Details**  If in the basic flow, the actor enters an invalid email id, the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the account creation process at which point the Use Case ends.  **Alternative Flow 2: An Account with the same credentials already exists**  If the Student/Faculty enters an email such that an account with the same id preexists, the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the account creation process at which point the Use Case ends.  **Alternative Flow 3: User Exits**  This allows the user to exit at any time during the use case. The use case ends |
| **Special Requirements:** None |
| **Associated Use Case:** None |

**2. LOGIN**

|  |
| --- |
| **Introduction:** The use case documents the steps that must be followed in order to login into the system. |
| **Actors:** Administrator, Student, Faculty |
| **Pre-Condition:** The administrator/ student/ faculty must be registered onto the system before the use case begins. |
| **Post Condition:** If the use case is successful, the administrator/ student/ faculty is logged into the system else the system state remains unchanged. |
| **Event Flow**  **Basic Flow**  1. The system requests that the actor enter his login id, password. The role of the user is determined based on the login id.  2. The Administrator/Student/Faculty enters his/her login id, password.  3. The system validates the entered credentials and logs the actor into the system. |
| **Alternative Flow**  **Alternative Flow 1: Invalid Login Details**  If in the basic flow, the actor enters an invalid login id, password the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the login at which point the Use Case ends.  **Alternative Flow 2: User Exits**  This allows the user to exit at any time during the use case. The use case ends |
| **Special Requirement:** None |
| **Associated Use Case:** Register |

**3. EDIT DETAILS**

|  |
| --- |
| **Introduction:** This use case documents the steps that must be followed to edit account details. |
| **Actors:** Administrator, Student, Faculty. |
| **Pre-Conditions:** The Administrator/Student/Faculty must be logged into the system before this use case begins. |
| **Post Conditions:** If the use case is successful, the Administrator/Student/Faculty details are edited in the system else the system state remains unchanged. |
| **Event Flow**  **Basic Flow:**  This use case starts when the Administrator/Student/Faculty wishes to edit their details.   1. Administrator/Student/Faculty updates the required details. 2. The System validates and details are successfully updated in the system. |
| **Alternative Flow 1: Invalid Entry**  If any mandatory fields are left blank or of improper format, a popup is displayed showing an error message.  **Alternative Flow 2: Actor Exits** If the Administrator/ Faculty/ Student exits the system without successfully completing the use case, the system state remains unchanged. |
| **Specific Requirement:** None |
| **Associated Use Case:** Login |

**4. VIEW DEPARTMENT DETAILS**

|  |
| --- |
| **Introduction:** This use case documents the steps that must be followed to view department details. |
| **Actors:** Administrator, Student, Faculty |
| **Pre-Conditions:** The Administrator/Student/Faculty must be logged into the system before this use case begins. |
| **Post Conditions**  If the use case is successful, the department details are displayed else an error message is displayed. |
| **Event Flow**  **Basic Flow:**  This use case starts when the Administrator/Student/Faculty wishes to view details of a department.   1. The Administrator/Student/Faculty selects a department from the list. 2. The Department details are displayed. |
| **Alternative Flow 1: Department Details are not available** If the Department details are not available, an error message is shown. The user can either choose to return to the basic flow of the use case or the use case ends.  **Alternative Flow 2: User Exits**  If at any point of time, the Administrator/Student/Faculty decides not to view the department details, the use case ends. |
| **Special Requirements:** None |
| **Associated Use Case:** Login |

**5. VIEW TIME TABLE**

|  |
| --- |
| **Introduction:** This use case documents the steps that must be followed to view time table. |
| **Actors:** Administrator, Student, Faculty |
| **Pre-Conditions:** The Administrator/Student/Faculty must be logged into the system before this use case begins. |
| **Post Conditions**  If the use case is successful, the department details are displayed else an error message is displayed. |
| **Event Flow**  **Basic Flow:**  This use case starts when the Administrator/Student/Faculty wishes to view the time table.   1. The Administrator/Student/Faculty selects a department and section from the list and the time table is displayed. 2. The Faculty can directly view his/her time table. |
| **Alternative Flow 1: Time Table is not available** If the Time Table of the selected department is not available, an error message is shown. The user can either choose to return to the basic flow of the use case or the use case ends.  **Alternative Flow 2: User Exits**  If at any point of time, the Administrator/Student/Faculty decides not to view the time table, the use case ends. |
| **Special Requirements:** None |
| **Associated Use Case:** Login |

**6. COURSE SELECTION**

|  |
| --- |
| **Introduction:** This use case documents the steps that must be followed for a student to select course. |
| **Actors:** Student |
| **Pre-condition**: The student must be logged into the system. |
| **Post-condition:** If the use case is successful, the student course details are updated in the system else the system state remains unchanged. |
| **Basic flow:**  1. The student selects the courses alloted to the department of the student.  2. Submit button is clicked and the courses are updated in the system. |
| **Alternate flow 1: No Seats Available**   1. If the seats corresponding to any course selected is full, a popup message is shown containing the name of the course having no seats. The user returns to the basic flow of the use case.   **Alternate flow 2: User Exits**   1. The user decides to cancel the course selection procedure and exits from the system. The use case ends. |
| **Special Requirements:** None |
| **Associated use cases:** Login |

**7. PAY REGISTRATION FEES**

|  |
| --- |
| **Brief Description:** This use case documents the steps that must be followed to pay the registration fees. |
| **Actors:** Administrator, Student |
| **Pre-Conditions :** The Administrator/Student must be logged into the system |
| **Post Conditions**: If the use case is successful, the Administrator/Student is able to pay the registration fees, if any. |
| **Flow of Events**  **Basic Flow**  1. The Student/Administrator selects the option to pay the registration fees.  2. The system displays the registration fees of the student inclusive of all taxes as the case may be.  3. The Student pays the bill using online modes of payment- Credit Card/Debit Card/Net Banking |
| **Alternative Flow 1: Invalid Details**  If in the basic flow, the Student/Administrator enters an invalid information the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the payment process at which point the Use Case ends.    **Alternative Flow 2: User Exits**  This allows the user to exit at any time during the use case. The use case ends |
| **Special Requirements:** None |
| **Associated Use Case:** Login, Receipt Generation |

**8. RECEIPT GENERATION**

|  |
| --- |
| **Introduction:** This use case generates a receipt after successful payment by the user. |
| **Actors:** Administrator, Student |
| **Pre-condition**: The customer must be logged into the system.   A successful payment should be made |
| **Post-condition:** A receipt is generated and sent to the student. |
| **Basic flow:**  1. After successful payment by the user an invoice consisting of details such as payment reference id, date, time, bank details is generated.  2. The receipt is emailed to the student. |
| **Alternate flow 1: User Exits**  The user decides to cancel the post and exists from the system. The use case ends. |
| **Special Requirements:** None |
| **Associated use cases:** Login |

**9. PRINT REGISTRATION FORM**

|  |
| --- |
| **Introduction:** This use case generates a registration after successful payment of registration fees and course selection. |
| **Actors:** Administrator, Student |
| **Pre-condition**: The customer must be logged into the system.   A successful payment should be made.  Courses should have been selected. |
| **Post-condition:** A registration form is downloaded onto the user’s system. |
| **Basic flow:**  1. A registration form is generated containing student details and courses selected. |
| **Alternate Flow 1: Courses not have been selected**  If the user has not selected the courses, an error message is shown to the user and the use case ends.  **Alternate Flow 2: Registration Fees not paid**  If the user has not yet paid the registration fess, an error message is shown to the user and the use case ends.  **Alternate flow 3: User Exits**  The user decides to cancel the post and exists from the system. The use case ends. |
| **Special Requirements:** None |
| **Associated use cases:** Login , Pay Registration Fees |

**7. MAINTAIN STUDENT DETAILS**

|  |
| --- |
| **Brief Description:** This use case documents the steps that must be followed in order to maintain Student details. This includes adding, deleting, updating ,viewing student details. |
| **Actors:** Administrator |
| **Pre-Conditions :** The Administrator must be logged into the system. |
| **Post Conditions**: If the use case is successful, the required information is added/updated/deleted/viewed from the system. Otherwise, the system state is unchanged. |
| **Flow of Events**  **Basic Flow:**  The use case begins when the Administrator wishes to add, delete, update, view student information from the system.   1. The system requests that the Administrator specify the function he/she would like to perform. 2. If the Administrator/Manager selects Add Room , Add a Room subflow is executed. 3. If the Administrator/Manager selects Update Room Details , Update a Room subflow is executed. 4. If the Administrator/Manager selects Delete Room , Delete a Room subflow is executed.   Basic Flow 1: Add a room  1. The system requests that the Administrator/Manager enter the room information.  2. Once the Administrator/Manager provides the requested information, the operator checks if the room number is unique. The room is added to the system.    Basic Flow 2: Update Room Details    1. The system requests that the Administrator/Manager enter the room number  2. The Administrator/Manager enter the room number  3. The system retrieves and displays the room details  4. The Administrator/Manager make the desired changes to the hotel room information  5. Once the Administrator/Manager updates the necessary information the system updates the student record with the updated information  Basic Flow 3: View Room Details    1. The system requests that the Administrator/Manager specify the room number  2. The system retrieves and displays the room information |
| **Alternative Flow 1: Invalid Details**  If in the basic flow, the Student/Administrator enters an invalid information the system displays an error message. The user can choose to either return to the beginning of the basic flow or cancel the payment process at which point the Use Case ends.  **Alternative Flow 2: User Exits**  This allows the user to exit at any time during the use case. The use case ends |
| **Special Requirements:** None |
| **Associated Use Case:** Login, Receipt Generation |

10 Maintain Booking Details

Introduction: This use case documents the steps that must be followed in order to maintain Hotel Booking details. This includes adding , updating ,viewing room information

Actors: Administrator, Manager

Pre-Conditions

The Administrator/Manager must be logged in to the system before the use case begins

Post Conditions

If the use case is successful, the required information is added/updated/deleted/viewed from the system. Otherwise, the system state is unchanged.

Evet Flow

Basic Flow

The use case begins when the Administrator/Manager wishes to add, update, view room information from the system.

1. The system requests that the Administrator/Manager specify the function he/she would like to perform.

2. If the Administrator/Manager selects Add Room , Add a Room subflow is executed

3. If the Administrator/Manager selects Update Room Details , Update a Room subflow is executed

4. If the Administrator/Manager selects Delete Room , Delete a Room subflow is executed

Basic Flow 1: Add a room

1. The system requests that the Administrator/Manager enter the room information.

2. Once the Administrator/Manager provides the requested information, the operator checks if the room number is unique. The room is added to the system.

Basic Flow 2: Update Room Details

1. The system requests that the Administrator/Manager enter the room number

2. The Administrator/Manager enter the room number

3. The system retrieves and displays the room details

4. The Administrator/Manager make the desired changes to the hotel room information

5. Once the Administrator/Manager updates the necessary information the system updates the student record with the updated information

Basic Flow 3: View Room Details

1. The system requests that the Administrator/Manager specify the room number

2. The system retrieves and displays the room information

Alternate Flow

Alternate Flow 1: User Exits

This allows the user to exit at any time during the use case. The use case ends

Alternate Flow 2: Room does not exist

If the entered information does not return any reference to a room in the hotel , a message reporting failure is displayed

Alternate Flow 3: Room already exists

If in the Add Room Sub flow, a room with the number already exists , the system displays an error message. The administrator returns to the basic flow and may reenter information

Special Requirement

None

Associated Use Case :

None