**OBJECT ORIENTED ANALYSIS AND DESIGN**

**PROJECT -I**

SOLUTION Q-1

**Step 1: Define a Method to Find Use-Cases**

**Methodology:**

1. **Identify Actors:**
   * List all the primary and secondary actors who interact with the system.
2. **Determine Actor Goals:**
   * For each actor, identify their goals when using the system.
3. **Identify Use-Cases:**
   * For each goal, define a use-case that describes how the system helps the actor achieve that goal.
4. **Outline Use-Cases:**
   * Use a consistent format to describe each use-case:
     + **Use-Case Name**
     + **Actor(s)**
     + **Goal**
     + **Main Success Scenario**
5. **Validate Use-Cases:**
   * Review the use-cases with stakeholders to ensure completeness and accuracy.

**Step 2: Execute the Method to Identify the Use-Cases**

**1. Identify Actors**

From the scenario, the actors are:

* **Students**
* **Faculty**
* **University Administrative Staff**

**2. Determine Actor Goals**

**Students' Goals:**

* Obtain information about courses.
* Find out when and where classes meet.
* Register for classes.
* Drop classes.
* Print transcripts.

**Faculty's Goals:**

* Find out what classes they are assigned to teach.
* Know when and where these classes meet.
* Get a list of students registered for each class.
* Assign grades to students.

**Administrative Staff's Goals:**

* Add new faculty and students.
* Remove faculty and students who have left.
* Enter and update information about each course.
* Enter schedules for classes offered each term.
* Perform other housekeeping tasks.

**3. Identify Use-Cases**

For each goal, we'll define a corresponding use-case.

**Step 3: List All Found Use-Cases Using the Specified Format**

**Use-Case 1: View Course Information**

* **Actor**: Student
* **Goal**: Obtain detailed information about courses, including when and where classes meet.
* **Main Success Scenario**:
  + The student logs into the system.
  + Searches for courses by name, code, or department.
  + The system displays course details, schedules, and locations.

**Use-Case 2: Register for Classes**

* **Actor**: Student
* **Goal**: Register for desired classes.
* **Main Success Scenario**:
  + The student logs into the system.
  + Selects classes to enroll in.
  + The system checks for prerequisites and schedule conflicts.
  + Upon validation, the student is enrolled, and the system updates their schedule.

**Use-Case 3: Drop Classes**

* **Actor**: Student
* **Goal**: Drop previously registered classes.
* **Main Success Scenario**:
  + The student logs into the system.
  + Views their current class schedule.
  + Selects the class to drop.
  + The system processes the request and updates the student's schedule.

**Use-Case 4: Print Transcript**

* **Actor**: Student
* **Goal**: Print an academic transcript.
* **Main Success Scenario**:
  + The student logs into the system.
  + Navigates to the transcript section.
  + Requests a transcript.
  + The system generates and displays the transcript, which the student can print.

**Use-Case 5: View Assigned Classes**

* **Actor**: Faculty
* **Goal**: Find out what classes they are assigned to teach.
* **Main Success Scenario**:
  + The faculty member logs into the system.
  + Accesses their teaching schedule.
  + The system displays a list of assigned classes with times and locations.

**Use-Case 6: View Class Schedule**

* **Actor**: Faculty
* **Goal**: Know when and where their classes meet.
* **Main Success Scenario**:
  + The faculty member accesses class details.
  + The system displays schedules and classroom locations for each class.

**Use-Case 7: View Student Roster**

* **Actor**: Faculty
* **Goal**: Get a list of students registered for each class.
* **Main Success Scenario**:
  + The faculty member selects a class from their schedule.
  + The system displays the roster of enrolled students.

**Use-Case 8: Assign Grades**

* **Actor**: Faculty
* **Goal**: Assign grades to students in their classes.
* **Main Success Scenario**:
  + The faculty member selects a class.
  + Enters grades for each student.
  + Submits the grades.
  + The system records the grades and updates student transcripts.

**Use-Case 9: Add New Students**

* **Actor**: Administrative Staff
* **Goal**: Add new students to the system.
* **Main Success Scenario**:
  + The staff member logs into the system with administrative access.
  + Navigates to the "Add New Student" section.
  + Enters student details (personal information, program of study).
  + The system creates a new student account.

**Use-Case 10: Remove Students**

* **Actor**: Administrative Staff
* **Goal**: Remove students who have left the university.
* **Main Success Scenario**:
  + The staff member searches for the student.
  + Confirms the removal.
  + The system deactivates the student's account and archives records.

**Use-Case 11: Add New Faculty**

* **Actor**: Administrative Staff
* **Goal**: Add new faculty members to the system.
* **Main Success Scenario**:
  + The staff member navigates to "Add New Faculty."
  + Enters faculty details (contact information, department).
  + The system creates a new faculty account.

**Use-Case 12: Remove Faculty**

* **Actor**: Administrative Staff
* **Goal**: Remove faculty members who have left.
* **Main Success Scenario**:
  + The staff member searches for the faculty member.
  + Confirms the removal.
  + The system deactivates the faculty member's account.

**Use-Case 13: Update Course Information**

* **Actor**: Administrative Staff
* **Goal**: Enter and update information about each course the university offers.
* **Main Success Scenario**:
  + The staff member selects a course to update or creates a new course.
  + Enters or modifies course details (title, description, credits).
  + The system saves the changes.

**Use-Case 14: Schedule Classes**

* **Actor**: Administrative Staff
* **Goal**: Enter the schedules for classes offered each term.
* **Main Success Scenario**:
  + The staff member selects a course offering.
  + Assigns it to a term, sets times, locations, and assigns faculty.
  + The system updates the class schedule for students and faculty.

**Use-Case 15: Perform Housekeeping Tasks**

* **Actor**: Administrative Staff
* **Goal**: Perform any other necessary maintenance tasks.
* **Main Success Scenario**:
  + The staff member accesses the maintenance section.
  + Performs tasks such as updating system settings or generating reports.
  + The system executes the tasks and confirms completion.

SOLUTION – Q2)

Based on the problem description, the key rules for the university registration system are:

1. **Course vs Course Offering**: A course (e.g. "CS 430: Principles of Object-Oriented Software Construction") may be offered multiple times, such as once in the fall and once in the spring. Each offering has some common information (course name, description, credits) and some unique information (meeting time, location, enrolled students).
2. **Student Registration**: Students can register for course offerings, drop courses, and view their transcript.
3. **Faculty Course Management**: Faculty can view the courses they are assigned to teach, view the list of students enrolled in each of their courses, and assign grades to students.
4. **Administrative Tasks**: Administrative staff can add/remove faculty and students, manage the information about each course offered, and update class schedules.
5. **Common Information vs Unique Information**: Course offerings will have some information in common (course name, description, credits) and some information that is unique to that specific offering (meeting time, location, enrolled students).

**6. Notifications**

**Important Dates**: System alerts users about deadlines and events.

**Class Changes**: Notify affected parties of schedule changes or cancellations.

**7. Data Privacy and Security**

**Compliance**: Adhere to data protection regulations (e.g., FERPA).

**Password Policies**: Enforce strong passwords and regular updates.

**8. System Performance**

**Availability**: Ensure high uptime, especially during peak periods.

**Scalability**: Handle increased load during registration times.

**9. Error Handling and Support**

**Clear Messages**: Provide understandable error explanations.

**Help Resources**: Offer guides and support contact information.

**10. Reporting and Analytics**

**Administrative Reports**: Generate reports on enrollment and scheduling.

SOLUTION Q-3

**Use Case: Register for Classes (1)**

**1. Use Case Name: Register for Classes**

**2. Primary Actor: Student**

**3. Stakeholders and Interests:**

* **Student:** Wants to enroll in desired classes without schedule conflicts, ensuring prerequisites are met.
* **University Administrative Staff:** Needs accurate enrollment data for resource allocation and reporting.
* **Faculty:** Wants to know the number of students enrolled in their classes.

**4. Preconditions:**

* The student is authenticated and logged into the system.
* The registration period is currently open.
* The student has no holds on their account preventing registration (e.g., financial, disciplinary).
* Course offerings for the term are available in the system.
* The student meets all prerequisites for the desired course.

**5. Postconditions:**

* **Success Postcondition:** The student is enrolled in the selected class(es), and their schedule is updated.
* **Failure Postcondition:** The student's enrollment status remains unchanged, and appropriate error messages are provided.

**6. Main Success Scenario (Basic Flow):**

1. **Student accesses the registration system** from their dashboard after logging in.
2. **System displays the list of available courses** for the term, including details such as course codes, titles, schedules, locations, and remaining seat capacity.
3. **Student searches for desired courses** using filters (e.g., department, course level, time slots).
4. **Student selects a course** to register for by adding it to their registration cart.
5. **System checks for prerequisites** and confirms the student has met them.
6. **System checks for schedule conflicts** with the student's existing enrolled classes.
7. **System checks for holds** on the student's account that may prevent registration.
8. **System verifies seat availability** in the selected class.
9. **System updates the class's enrollment count.**
10. **System sends a confirmation notification** to the student via email and/or in-system messaging.
11. **Student reviews their updated schedule** to ensure accuracy.

**7. Alternative Flows:**

* **7a. Prerequisite Not Met:**
  + **At Step 5**, if the student has not met the prerequisites:
    - **5a.1** System displays an error message indicating unmet prerequisites.
    - **5a.2** Student may choose to seek an override from the faculty or select a different course.
    - **Use Case Ends.**
* **7b. Schedule Conflict Detected:**
  + **At Step 6**, if a time conflict is detected:
    - **6b.1** System displays a warning about the schedule conflict.
    - **6b.2** Student may choose to drop the conflicting class or select an alternative course.
    - **Use Case Returns to Step 3** if the student selects a different course.
* **7c. Account Hold Present:**
  + **At Step 7**, if there is a hold on the student's account:
    - **7c.1** System informs the student of the hold and provides instructions on how to resolve it.
    - **7c.2** Student cannot proceed with registration until the hold is cleared.
    - **Use Case Ends.**
* **7d. Class Full (No Seat Availability):**
  + **At Step 8**, if no seats are available:
    - **8d.1** System notifies the student that the class is full.
    - **8d.2** System offers the option to join a waitlist if available.
      * **8d.2.1** If the student joins the waitlist:
        + **System adds the student to the waitlist** and confirms their position.
        + **Use Case Ends.**

**Use Case: Drop a Class (2)**

**1. Use Case Name: Drop a Class**

**2. Primary Actor: Student**

**3. Stakeholders and Interests:**

* **Student:** Wants to adjust their schedule by removing a class without penalties.
* **University Administrative Staff:** Needs accurate enrollment data.
* **Faculty:** Needs to know the current roster of students.

**4. Preconditions:**

* The student is authenticated and logged into the system.
* The add/drop period is currently open.
* The class is eligible to be dropped (e.g., not past the drop deadline, no restrictions).

**5. Postconditions:**

* **Success Postcondition:** The class is removed from the student's schedule, and enrollment data is updated.
* **Failure Postcondition:** The student's schedule remains unchanged, and the student is informed of the reason.

**6. Main Success Scenario (Basic Flow):**

1. **Student accesses their current schedule** through the system dashboard.
2. **Student selects the class** they wish to drop.
3. **System verifies that the class is eligible to be dropped** (e.g., within the allowed period).
4. **Student confirms the drop action.**
5. **System removes the class** from the student's schedule.
6. **System updates the class's enrollment count**, increasing available seats by one.
7. **System sends a confirmation notification** to the student.
8. **Student reviews their updated schedule** to ensure the class has been removed.

**7. Alternative Flows:**

* **7a. Drop Deadline Passed:**
  + **At Step 3**, if the drop deadline has passed:
    - **3a.1** System informs the student that the class cannot be dropped due to the deadline.
    - **Use Case Ends.**
* **7b. Class Not Eligible for Drop:**
  + **At Step 3**, if the class has restrictions (e.g., mandatory course, special program requirements):
    - **3b.1** System displays a message explaining why the class cannot be dropped.
    - **Use Case Ends.**
* **7c. System Error Occurs:**
  + **At any step**, if a system error occurs:
    - **System displays an error message** and logs the issue.
    - **Student may retry the action** or contact support.
    - **Use Case Ends or Retries from the Appropriate Step.**

**Use Case: View Course Information (3)**

**1. Use Case Name: View Course Information**

**2. Primary Actor: Student**

**3. Stakeholders and Interests:**

* **Student:** Wants detailed information to make informed decisions about course selections.
* **Faculty:** Provides accurate course descriptions and requirements.
* **Administrative Staff:** Ensures course information is current and accurate.

**4. Preconditions:**

* The student is authenticated and logged into the system.
* Course data is available and up-to-date.

**5. Postconditions:**

* **Success Postcondition:** Student has viewed the desired course information.
* **Failure Postcondition:** Student is informed if the course information is unavailable.

**6. Main Success Scenario (Basic Flow):**

1. **Student navigates to the course catalog** within the system.
2. **Student searches for courses** by filters such as course code, title, department, or instructor.
3. **System displays a list of courses** matching the search criteria.
4. **Student selects a course** to view detailed information.
5. **System displays detailed course information**, including:
   * Course description
   * Prerequisites
   * Credit hours
   * Schedule and location of offerings
   * Instructor information
   * Enrollment capacity and availability
6. **Student reviews the information** and may choose to add the course to their registration cart.

**7. Alternative Flows:**

* **7a. No Courses Found:**
  + **At Step 3**, if no courses match the search criteria:
    - **3a.1** System displays a message indicating no courses were found.
    - **3a.2** Student may modify search criteria or exit the search.
    - **Use Case Returns to Step 2** if the student continues searching.
* **7b. Course Details Unavailable:**
  + **At Step 5**, if detailed information is missing:
    - **5b.1** System informs the student that some details are currently unavailable.
    - **Use Case Continues**, displaying available information.

**Use Case : Assign Grades to Students (4)**

**1. Use Case Name: Assign Grades to Students**

**2. Primary Actor: Faculty Member**

**3. Stakeholders and Interests:**

* **Faculty Member:** Wants to accurately record and submit grades for students in their classes.
* **Students:** Expect timely and accurate updates to their academic records.
* **University Administration:** Needs grade data for reporting and accreditation purposes.

**4. Preconditions:**

* The faculty member is authenticated and logged into the system.
* The faculty member is assigned to teach the class.
* The grading period is currently open.
* The class roster is up-to-date.

**5. Postconditions:**

* **Success Postcondition:** Grades are successfully recorded in the system, and students' academic records are updated.
* **Failure Postcondition:** No changes are made to students' grades, and the faculty member is informed of any issues.

**6. Main Success Scenario (Basic Flow):**

1. **Faculty member logs into the system** and navigates to their dashboard.
2. **Faculty selects the class** for which they want to assign grades from their list of assigned classes.
3. **System displays the class roster**, including student names and IDs.
4. **Faculty enters grades** for each student on the roster.
5. **Faculty reviews the entered grades** for accuracy.
6. **Faculty submits the grades** for finalization.
7. **System validates the grades**, ensuring they meet grading standards (e.g., valid grade formats).
8. **System records the grades**, updating each student's academic record.
9. **System sends confirmation notifications** to the faculty member and optionally to students.
10. **Students can view their updated grades** in their respective accounts.

**7. Alternative Flows:**

* **7a. Invalid Grade Entry:**
  + **At Step 7**, if an invalid grade is detected:
    - **7a.1** System highlights the invalid entries and provides an error message specifying the issue.
    - **7a.2** Faculty corrects the invalid grade(s).
    - **Use Case Returns to Step 5** for review before resubmission.
* **7b. Grading Deadline Passed:**
  + **At Step 1**, if the grading period has closed:
    - **7b.1** System displays a message indicating that the grading deadline has passed.
    - **7b.2** Faculty cannot proceed without administrative override.
    - **Use Case Ends.**
* **7c. System Error Occurs:**
  + **At any step**, if a system error occurs:
    - **System displays an error message** and logs the issue.
    - **Faculty may retry the action** or contact technical support.
    - **Use Case Ends or Retries from the Appropriate Step.**

**Use Case : Add a New Student (5)**

**1. Use Case Name: Add a New Student**

**2. Primary Actor: University Administrative Staff**

**3. Stakeholders and Interests:**

* **Administrative Staff:** Needs to efficiently add new students to the system with accurate information.
* **Student:** Expects to gain access to the system and its services promptly.
* **University Administration:** Requires accurate student records for planning and reporting.

**4. Preconditions:**

* The administrative staff member is authenticated and has the necessary permissions.
* The student has been accepted for admission and has provided all required information.

**5. Postconditions:**

* **Success Postcondition:** The new student is added to the system with a unique student ID and login credentials.
* **Failure Postcondition:** The student is not added to the system, and the staff member is informed of any issues.

**6. Main Success Scenario (Basic Flow):**

1. **Administrative staff logs into the system** and navigates to the "Add New Student" section.
2. **Staff enters the student's personal information**, including name, date of birth, contact details, and admission ID.
3. **Staff inputs academic program information**, such as major, degree sought, and anticipated start term.
4. **System validates the entered data** for completeness and correctness.
5. **System checks for duplicate records** to ensure the student is not already in the system.
6. **Staff confirms the addition** of the new student.
7. **System creates a new student record**, assigning a unique student ID.
8. **System generates login credentials** (username and temporary password) for the student.
9. **System sends a welcome email** to the student's provided email address with login instructions.
10. **Student is now able to log into the system** and access student services.

**7. Alternative Flows:**

* **7a. Incomplete or Invalid Information:**
  + **At Step 4**, if required information is missing or invalid:
    - **4a.1** System highlights missing or incorrect fields and provides error messages.
    - **4a.2** Staff corrects the information.
    - **Use Case Returns to Step 4** for re-validation.
* **7b. Duplicate Student Detected:**
  + **At Step 5**, if a duplicate record is found:
    - **5b.1** System alerts the staff member of the potential duplicate with details.
    - **5b.2** Staff reviews the duplicate information.
      * **If it's a duplicate:**
        + **5b.3** Staff cancels the operation or updates the existing record if necessary.
        + **Use Case End**

**Use Case : View Assigned Classes (6)**

**1. Use Case Name: View Assigned Classes**

**2. Primary Actor: Faculty Member**

**3. Stakeholders and Interests:**

* **Faculty Member:** Wants to know which classes they are teaching, including schedules and locations.
* **Students:** Depend on accurate scheduling for class attendance.
* **Administrative Staff:** Needs faculty to be informed of their assignments for planning purposes.

**4. Preconditions:**

* The faculty member is authenticated and logged into the system.
* Class assignments have been made and are up-to-date in the system.

**5. Postconditions:**

* **Success Postcondition:** Faculty member has viewed their teaching schedule with all relevant details.
* **Failure Postcondition:** Faculty member is informed if no classes are assigned or if information is unavailable.

**6. Main Success Scenario (Basic Flow):**

1. **Faculty member logs into the system** and is directed to their dashboard.
2. **Faculty selects the "My Classes"** option from the menu.
3. **System displays a list of classes** the faculty member is assigned to teach for the current term.
4. **For each class, the system shows**:
   * Course code and title
   * Class schedule (days and times)
   * Classroom location
   * Enrollment numbers
5. **Faculty reviews the list** to understand their teaching commitments.
6. **Faculty selects a class** to view more detailed information if desired.
7. **System displays detailed class information**, including:
   * Course description
   * Student roster
   * Links to upload course materials or communicate with students

**7. Alternative Flows:**

* **7a. No Classes Assigned:**
  + **At Step 3**, if the faculty member has no assigned classes:
    - **3a.1** System displays a message indicating no classes are currently assigned.
    - **3a.2** Faculty may contact the department head for clarification.
    - **Use Case Ends.**
* **7b. Future or Past Terms:**
  + **At Step 2**, if the faculty wants to view classes for a different term:
    - **2b.1** Faculty selects the desired term from a dropdown menu.
    - **2b.2** System displays classes for the selected term.
    - **Use Case Continues from Step 3.**
* **7c. System Error Occurs:**
  + **At any step**, if a system error occurs:
    - **System displays an error message** and logs the issue.
    - **Faculty may retry the action** or contact support.
    - **Use Case Ends or Retries from the Appropriate Step.**

**8. Special Requirements:**

* The system should ensure data is real-time and reflects any recent changes in class assignments.
* The interface must be intuitive, allowing faculty to easily navigate between classes and terms.
* Mobile accessibility could be beneficial for faculty on the go.

SOLUTION Q-4

**Use Case: Register for Classes**

**Acceptance Criteria:**

1. **Authentication and Authorization:**
   * Students must be able to log in using valid credentials.
   * Unauthorized users cannot access the registration system.
2. **Registration Period:**
   * Students can register only during the designated registration periods.
   * The system should display appropriate messages if the registration period is closed.
3. **Course Search and Display:**
   * Students can search and filter courses by code, name, department, schedule, and other relevant criteria.
   * The system displays up-to-date course details, including prerequisites, capacity, schedule, location, and instructor.
4. **Prerequisite Verification:**
   * The system must check and confirm that students meet all prerequisites before allowing registration.
   * If prerequisites are not met, a clear and informative message is displayed.
5. **Schedule Conflict Detection:**
   * The system detects any schedule conflicts with the student's existing classes.
   * Students are notified of conflicts and cannot proceed until conflicts are resolved.
6. **Account Holds Check:**
   * The system checks for any holds (financial, disciplinary, etc.) on the student's account.
   * If holds exist, the student is informed and provided with instructions on how to resolve them.

**Use Case: Drop a Class**

**Acceptance Criteria:**

1. **Authentication and Authorization:**
   * Only authenticated students can access the drop class functionality.
   * Unauthorized access is prevented.
2. **Add/Drop Period Enforcement:**
   * The system allows dropping of classes only within the designated add/drop period.
   * If the period has passed, an informative message is displayed.
3. **Current Schedule Display:**
   * Students can view their current, up-to-date schedule with all enrolled classes.
4. **Eligibility Verification:**
   * The system verifies if the class is eligible to be dropped (e.g., no restrictions, not a corequisite).
   * Ineligible classes cannot be dropped, and an explanation is provided.
5. **Confirmation Process:**
   * A confirmation prompt is presented before finalizing the drop action.
   * Students must explicitly confirm to proceed.
6. **Successful Drop:**
   * The class is removed from the student's schedule upon confirmation.
   * The system updates the class's enrollment count.
   * A confirmation message is displayed and sent via email.

**Use Case: View Course Information**

**Acceptance Criteria:**

1. **Accessibility:**
   * Students can access course information without unnecessary barriers.
   * If login is required, authentication mechanisms are in place.
2. **Course Search and Filtering:**
   * The system allows searching and filtering of courses by various criteria.
   * Search results are accurate and relevant.
3. **Comprehensive Course Details:**
   * Course pages display all relevant information:
     + Course code and title
     + Description and objectives
     + Prerequisites and corequisites
     + Credit hours
     + Schedule and location for each offering
     + Instructor information
     + Enrollment capacity and current availability
4. **Real-Time Data:**
   * Information is up-to-date, reflecting any recent changes.
5. **Usability:**
   * The interface is intuitive and user-friendly.
   * Navigation between courses and search results is seamless.
6. **Error Handling:**
   * Informative messages are provided if no courses match the search criteria.
   * System errors are handled gracefully.
7. **Performance:**
   * Search and page loading times are acceptable (e.g., under 3 seconds).

**Use Case: Assign Grades to Students**

**Acceptance Criteria:**

1. **Authentication and Authorization:**
   * Only authenticated faculty members can access grade entry.
   * Faculty can only enter grades for classes they are assigned to teach.
2. **Class Roster Access:**
   * Faculty can view an accurate and complete roster of enrolled students.
3. **Grade Entry and Validation:**
   * Faculty can enter grades for each student.
   * The system validates grades against acceptable formats (e.g., letter grades, numerical ranges).
4. **Review and Confirmation:**
   * Faculty can review all entered grades before submission.
   * A confirmation step is required to finalize grades.
5. **Grading Period Enforcement:**
   * Grades can only be submitted within the designated grading period.
   * The system informs faculty of submission deadlines.
6. **Successful Submission:**
   * Grades are securely recorded in the system.
   * Students' academic records are updated.
   * Confirmation messages are provided to the faculty member.
7. **Student Notification:**
   * Optionally, students are notified when grades are posted.
8. **Data Integrity and Security:**
   * Grade data is stored securely to maintain confidentiality.
   * Unauthorized access or modification is prevented.

**Use Case: Add a New Student**

**Acceptance Criteria:**

1. **Authentication and Authorization:**
   * Only authorized administrative staff can add new students.
   * Access to this functionality is secured.
2. **Data Entry and Validation:**
   * Staff can enter all required student information.
   * Mandatory fields are enforced.
   * Data validation checks are performed (e.g., email format, date ranges).
3. **Duplicate Prevention:**
   * The system checks for existing records to prevent duplicates.
   * Staff are alerted to potential duplicates with options to proceed or cancel.
4. **Confirmation Before Creation:**
   * Staff can review all entered data before finalizing the addition.
5. **Successful Account Creation:**
   * A unique student ID is generated.
   * Login credentials are securely created.
6. **Notification to Student:**
   * A welcome email with login instructions is sent to the student's email address.
7. **Security and Compliance:**
   * Personal data is stored in compliance with privacy laws (e.g., FERPA).
   * Access to student data is appropriately restricted.

**Use Case: View Assigned Classes**

**Acceptance Criteria:**

1. **Authentication and Authorization:**
   * Faculty must log in with valid credentials.
   * Only their own assigned classes are accessible.
2. **Accurate Class Listing:**
   * The system displays all classes assigned to the faculty member for the selected term.
   * Information is accurate and up-to-date.
3. **Detailed Class Information:**
   * Faculty can access additional details for each class, including:
     + Course description
     + Schedule and location
     + Enrollment numbers
     + Student roster
     + Links to class materials and communication tools
4. **Term Selection:**
   * Faculty can view assignments for current, past, and future terms.
5. **Usability:**
   * The interface is intuitive, with clear navigation.
   * Information is presented in a readable format.
6. **Error Handling:**
   * If no classes are assigned, an informative message is displayed.
   * System errors are communicated effectively.
7. **Performance:**
   * Pages load promptly, with minimal delays.
8. **Security and Privacy:**
   * Access to sensitive information (e.g., student data) is secure.
   * Faculty cannot access classes or data not assigned to them.
9. **Accessibility:**
   * The system adheres to accessibility standards.

SOLUTION- Q5

**Use Case: Register for Classes**

**Primary Actor:** Student

**User Stories:**

1. **As a student, I want to search for available courses, so that I can find classes that fit my interests and schedule.**
2. **As a student, I want to view detailed information about a course, so that I can understand prerequisites and course content.**
3. **As a student, I want to add a course to my registration cart, so that I can enroll in it if I meet all requirements.**
4. **As a student, I want the system to check for schedule conflicts, so that I do not enroll in overlapping classes.**
5. **As a student, I want to register for selected classes, so that I can secure my spot in them.**
6. **As a student, I want to receive notifications upon successful registration, so that I have a record of my enrollment.**
7. **As a student, I want to be informed if a class is full, so that I can choose to join a waitlist or select another class.**
8. **As a student, I want the ability to resolve holds on my account, so that I can proceed with registration.**

**Use Case: Drop a Class**

**Primary Actor:** Student

**User Stories:**

1. **As a student, I want to view my current class schedule, so that I can see which classes I am enrolled in.**
2. **As a student, I want to drop a class from my schedule, so that I can adjust my course load.**
3. **As a student, I want to receive a confirmation after dropping a class, so that I know the action was successful.**
4. **As a student, I want to be informed if I cannot drop a class, so that I understand any restrictions.**

**Use Case: View Course Information**

**Primary Actor:** Student

**User Stories:**

1. **As a student, I want to search for courses by various criteria, so that I can find classes that match my interests.**
2. **As a student, I want to view detailed course information, so that I can make informed decisions about enrolling.**
3. **As a student, I want to see the current availability of a course, so that I know if seats are open.**
4. **As a student, I want to save courses of interest, so that I can easily access them later.**

**Use Case: Assign Grades to Students**

**Primary Actor:** Faculty Member

**User Stories:**

1. **As a faculty member, I want to view the roster of students in my class, so that I can see who is enrolled.**
2. **As a faculty member, I want to enter grades for my students, so that I can record their academic performance.**
3. **As a faculty member, I want to review all grades before submission, so that I can ensure accuracy.**
4. **As a faculty member, I want to submit grades securely, so that student records are updated appropriately.**
5. **As a faculty member, I want to be notified of grading deadlines, so that I submit grades on time.**
6. **As a faculty member, I want to save grade entries as drafts, so that I can complete them at a later time.**

**Use Case: Add a New Student**

**Primary Actor:** Administrative Staff

**User Stories:**

1. **As an administrative staff member, I want to add a new student's personal information, so that they can be registered in the system.**
2. **As an administrative staff member, I want the system to check for duplicate student records, so that data integrity is maintained.**
3. **As an administrative staff member, I want to assign academic program details to the student, so that their academic profile is complete.**
4. **As an administrative staff member, I want to generate login credentials for the new student, so that they can access the system.**
5. **As an administrative staff member, I want to receive confirmation that the student was added successfully, so that I can proceed with other tasks.**

QUESTION-6

**Business Model Canvas**

**1. Customer Segments**

* **Primary Customers:**
  + Small to medium-sized **Universities and Colleges** needing efficient registration systems.
  + **Vocational Schools** and **Online Educational Platforms** requiring streamlined administrative tools.
* **End Users:**
  + **Administrative Staff**
  + **Faculty Members**
  + **Students**

**2. Value Propositions**

* **For Institutions:**
  + **Streamlined Operations**: Simplifies administrative tasks and reduces errors.
  + **Cost-Effective Solution**: Affordable pricing with essential features.
  + **Scalable and Customizable**: Adapts to institution size and specific needs.
  + **Regulatory Compliance**: Adheres to educational regulations and data protection laws.
  + **Robust Security**: Protects sensitive academic and personal data.
* **For Users:**
  + **User-Friendly Interface**: Easy navigation for staff, faculty, and students.
  + **Real-Time Access**: Immediate updates on schedules, enrollments, and grades.
  + **Efficient Communication**: Built-in tools for notifications and messaging.

**3. Channels**

* **Marketing and Sales:**
  + **Direct Sales Team**
  + **Digital Marketing** (website, social media)
  + **Educational Conferences and Trade Shows**
  + **Partnerships** with educational consultants
* **Distribution:**
  + **Software as a Service (SaaS)**
  + **On-Premises Installation**

**4. Customer Relationships**

* **Dedicated Support:**
  + **Onboarding Assistance**
  + **24/7 Customer Service**
* **Community Engagement:**
  + **User Forums**
  + **Feedback Mechanisms**
* **Account Management:**
  + **Personalized Account Managers**

**5. Revenue Streams**

* **Primary Revenue:**
  + **Subscription Fees**: Monthly or annual plans based on user count or features.
* **Secondary Revenue:**
  + **Customization Services**
  + **Training Programs**
  + **Premium Support**
  + **Add-On Features**

**6. Key Resources**

* **Human Resources:**
  + **Development Team**
  + **Customer Support**
  + **Sales and Marketing Team**
* **Technological Resources:**
  + **Software Platform**
  + **Cloud Infrastructure**
  + **Security Systems**

**7. Key Activities**

* **Product Development and Maintenance**
* **Marketing and Sales Efforts**
* **Customer Support and Training**
* **Regulatory Compliance and Security Management**

**8. Key Partnerships**

* **Technology Partners:**
  + **Cloud Service Providers**
  + **Integration Partners**
* **Educational Organizations:**
  + **Accreditation Bodies**
  + **Educational Consultants**
* **Resellers and Distributors**

**9. Cost Structure**

* **Fixed Costs:**
  + **Employee Salaries**
  + **Office Expenses**
  + **Software Licenses**
* **Variable Costs:**
  + **Marketing and Advertising**
  + **Customer Support Scaling**
  + **Hosting and Bandwidth**
* **Operational Costs:**
  + **Research and Development**
  + **Training**
  + **Compliance Expenses**

QUESTION 7

**Course Management (Core)**

* **Purpose:** Manage all course-related information.
* **Responsibilities:**
  + Maintain the course catalog with details like course codes, titles, descriptions, and prerequisites.
  + Update and modify course information as needed.
* **Type:** Core sub-domain essential for the system's operation.

 **Student Enrollment (Core)**

* **Purpose:** Facilitate student registration and enrollment processes.
* **Responsibilities:**
  + Enable students to register for and drop classes.
  + Check prerequisites and schedule conflicts during registration.
  + Manage waitlists for full classes.
* **Type:** Core sub-domain critical for student interaction.

 **Schedule Management (Core)**

* **Purpose:** Organize class schedules and offerings.
* **Responsibilities:**
  + Schedule classes for each term.
  + Assign faculty to class offerings.
  + Allocate classrooms and resources.
* **Type:** Core sub-domain necessary for coordinating classes.

 **User Management (Supporting)**

* **Purpose:** Handle user accounts and roles.
* **Responsibilities:**
  + Add, update, and remove student and faculty accounts.
  + Assign roles and permissions.
  + Manage authentication credentials.
* **Type:** Supporting sub-domain aiding core functionalities.

 **Grade Management (Core)**

* **Purpose:** Manage grading and academic records.
* **Responsibilities:**
  + Allow faculty to submit grades for students.
  + Update student transcripts.
  + Provide students access to view their grades.
* **Type:** Core sub-domain integral for academic tracking.

 **Authentication and Authorization (Infrastructure)**

* **Purpose:** Secure system access.
* **Responsibilities:**
  + Authenticate users during login.
  + Authorize user actions based on roles and permissions.
* **Type:** Infrastructure sub-domain supporting security.

 **Reporting and Analytics (Supporting)**

* **Purpose:** Provide data insights and reports.
* **Responsibilities:**
  + Generate enrollment statistics and reports.
  + Produce class capacity and utilization reports.
  + Provide academic performance analytics.
* **Type:** Supporting sub-domain enhancing decision-making.

 **Notification System (Supporting)**

* **Purpose:** Communicate important information to users.
* **Responsibilities:**
  + Send alerts about registration periods and deadlines.
  + Notify users of schedule changes or holds on accounts.
  + Deliver grade postings and other announcements.
* **Type:** Supporting sub-domain improving user engagement.

 **Financial Management (Optional)**

* **Purpose:** Handle billing and financial transactions.
* **Responsibilities:**
  + Process tuition payments and fees.
  + Manage financial holds affecting registration.
  + Integrate with accounting systems.
* **Type:** Optional sub-domain depending on system scope.