Admission Management

**Functional Requirements**

1. **User Registration and Authentication**
   * Allows new users to register and existing users to log in, with options for password recovery.
2. **Applicant Profile Management**
   * Enables applicants to create and update their profiles, storing personal and academic information.
3. **Course Management**
   * Administrators can add, update, and delete courses, while applicants can view course details.
4. **Application Submission**
   * Applicants can apply for courses and upload required documents, with validation for completeness.
5. **Payment Processing**
   * Supports online payment for application fees with multiple payment options and receipt generation.
6. **Application Review and Decision**
   * Administrators can review applications, make decisions, and notify applicants of the outcomes.
7. **Communication and Notifications**
   * Automated email notifications and messaging system for communication between applicants and administrators.
8. **Reporting and Analytics**
   * Generates reports on application statistics and supports data export for analysis.
9. **User Roles and Permissions**
   * Supports different user roles with restricted access based on role-specific permissions.

**Non-Functional Requirements**

1. **Performance**
   * Supports high concurrency and processes application submissions swiftly under normal load.
2. **Scalability**
   * Designed to handle increasing numbers of users and applications seamlessly.
3. **Security**
   * Implements strong encryption and protective measures against common security threats, ensuring regulatory compliance.
4. **Availability**
   * Ensures high availability with minimal downtime and robust backup and recovery mechanisms.
5. **Usability**
   * Features an intuitive interface and provides user help and documentation.
6. **Compatibility**
   * Compatible with major web browsers and supports mobile access through responsive design.
7. **Maintainability**
   * Modular design for easy updates and maintenance, with comprehensive developer documentation.
8. **Reliability**
   * Ensures data integrity and consistency with effective error handling mechanisms.
9. **Interoperability**
   * Supports integration with third-party services and provides APIs for data exchange.

2. User Story And Acceptance Criteria

Story - 1

|  |  |
| --- | --- |
| Functionality | **Applicant Registration** |
| Story - 1 | As a new user, I want to register an account, so that I can apply for courses. |
| Scenario – 1.1 | Successful User Registration |
| Acceptance criteria | **Given:** A new user navigates to the registration page.  **When:** The user fills in all the required fields (username, email, password, confirm password, etc.)  **Then:**   * The system validates the provided information. * If the information is valid, the system creates a new user account. * The user receives a confirmation email. * The user is redirected to a confirmation page or the login page with a success message. |

|  |  |
| --- | --- |
| Functionality | **Applicant Registration** |
| Story - 1 | As a new user, I want to register an account, so that I can apply for courses. |
| Scenario – 1.2 | Registration with Missing Information |
| Acceptance criteria | **Given:** A new user navigates to the registration page.  **When:** The user fills in only some of the required fields (e.g., username and email, but not password and confirm password)  **Then:**   * The system detects the missing information. * The system displays error messages next to the missing fields indicating that these fields are required. * The user remains on the registration page and is prompted to fill in the missing information. |

|  |  |
| --- | --- |
| Functionality | **Applicant Registration** |
| Story - 1 | As a new user, I want to register an account, so that I can apply for courses. |
| Scenario – 1.3 | Registration with Invalid Information |
| Acceptance criteria | **Given:** A new user navigates to the registration page.  **When:** The user fills in all the required fields, but enters invalid information (e.g., an email in an incorrect format, a password that does not meet the security criteria)  **Then:**   * The system detects the invalid information. * The system displays error messages next to the fields with invalid information indicating the nature of the errors * The user remains on the registration page and is prompted to correct the invalid information. |

**Story -2**

|  |  |
| --- | --- |
| Functionality | **Payment functionality** |
| Story - 2 | As an applicant, I want to choose from multiple payment options, so that I can select the most convenient method for me. |
| Scenario – 2.1 | Successful Payment with Credit Card |
| Acceptance criteria | **Given:** An applicant has selected a course and is on the payment page.  **When:** The applicant selects the credit card payment option. The applicant enters the required credit card details.  **Then:**   * The system validates the credit card details. * If the details are valid, the system processes the payment. * The applicant receives a payment confirmation and a receipt via email. * The applicant is redirected to a confirmation page or the course dashboard with a success message. |

|  |  |
| --- | --- |
| Functionality | **Payment functionality** |
| Story - 2 | As an applicant, I want to choose from multiple payment options, so that I can select the most convenient method for me. |
| Scenario – 2.2 | Payment with PayPal |
| Acceptance criteria | **Given:** An applicant has selected a course and is on the payment page.  **When:** The applicant selects the PayPal payment option. The applicant is redirected to the PayPal login page.  **Then:**   * The system processes the payment through PayPal. * The applicant receives a payment receipt and a confirmation email. * If the payment is successful, the system confirms the enrollment in the course. |

|  |  |
| --- | --- |
| Functionality | **Payment functionality** |
| Story - 2 | As an applicant, I want to choose from multiple payment options, so that I can select the most convenient method for me. |
| Scenario – 2.3 | Payment Failure Due to Insufficient Funds |
| Acceptance criteria | **Given:** An applicant has selected a course and is on the payment page.  **When:** The applicant selects the credit card payment option.  **Then:**   * The system validates the credit card details. * The system attempts to process the payment but detects insufficient funds. * The applicant remains on the payment page to retry the payment. |

**Story -3**

|  |  |
| --- | --- |
| Functionality | Course Management |
| Story - 3 | As an administrator, I want to add, update and delete courses, so that I can manage the course offerings efficiently and ensure that applicants have accurate and up-to-date information. |
| Scenario – 3.1 | Successful Addition of a New Course |
| Acceptance criteria | **Given:** An administrator is logged into the course management system.  **When:** The administrator navigates to the "Add Course" section. The administrator fills in the course details.  **Then:**   * The system validates the provided information. * If the information is valid, the system saves the new course to the database. * The new course is displayed in the course list for applicants. |

|  |  |
| --- | --- |
| Functionality | Course Management |
| Story - 2 | As an administrator, I want to add, update and delete courses, so that I can manage the course offerings efficiently and ensure that applicants have accurate and up-to-date information. |
| Scenario – 3.2 | Addition of a Course with Missing Information |
| Acceptance criteria | **Given:** An administrator is logged into the course management system.  **When:** The administrator navigates to the "Add Course" section. The administrator fills in only some of the required course details.  **Then:**   * The system detects the missing information. * The system displays error messages next to the missing fields indicating that these fields are required. * The administrator remains on the "Add Course" page and is prompted to fill in the missing information. |

|  |  |
| --- | --- |
| Functionality | Course Management |
| Story - 2 | As an administrator, I want to add, update and delete courses, so that I can manage the course offerings efficiently and ensure that applicants have accurate and up-to-date information. |
| Scenario – 3.3 | Addition of a Course with Invalid Information |
| Acceptance criteria | **Given:** An administrator is logged into the course management system.  **When:** The administrator navigates to the "Add Course" section. The administrator fills in the course details, but enters invalid information.  **Then:**   * The system detects the invalid information. * The system displays error messages next to the fields with invalid information indicating the nature of the errors * The administrator remains on the "Add Course" page and is prompted to correct the invalid information. |

3. Implement equivalence partitions

1.

Username : (numeric value length 5-10)

Password : (password length 8-12)

Solution :

For username:

Valid class : username length must be between 5 to 10 character.

Invalid class : username length is less than 5 character.

Invalid class : username length is greater than 10 character.

For password:

Valid class: password length must be between 8 to 12 character.

Invalid class: username length is less than 8 character.

Invalid class : username length is greater than 12 character.

2.

Percentage: (numeric value length 50-90)

Solution:

For percentage:

Valid class: percentage must be between 50-90.

Invalid class: percentage value is less than 50.

Invalid class: percentage value is greater than 90.

3.

OTP number : (number length is 6)

Solution:

For OTP number :

Valid class: number length must be 6.

Invalid class: number length is less than 6.

Invalid class: number length is greater than 6.

4.

Mobile number: (number length must be 10)

Solution:

For mobile number:

Valid class: number length must be 10.

Invalid class: number length is less than 10.

Invalid class: number length is greater than 10.

5.

Account balance: ( balance range greater than 0)

For account balance

Valid class : balance between 1 to 200 dollar

Valid class : balance between 201 to 900 dollar

Valid class : balance greater than 900 dollar

Invalid class : balance less than 1 dollar

5. Boundary value Analysis

1.

Age: (value range 18 to 19)

|  |  |  |
| --- | --- | --- |
| Invalid Test Case | Valid Test | Invalid Test Case |
| 17 | 18,19,55,56 | 57 |

Test case 1: Enter the value 17(18-1) – Invalid

Test case 2: Enter the value 18 – valid

Test case 3: Enter the value 19 – valid

Test case 4: Enter the value 55 – valid

Test case 5: Enter the value 56 – valid

Test case 6: Enter the value 57(56+1) – Invalid

2.

Name: (value range 6 to 12 character)

|  |  |  |
| --- | --- | --- |
| Invalid Test Case | Valid Test | Invalid Test Case |
| 5 | 6,7,11,12 | 13 |

Test case 1: Enter the value 5(6-1) – Invalid

Test case 2: Enter the value 6 – valid

Test case 3: Enter the value 7 – valid

Test case 4: Enter the value 11 – valid

Test case 5: Enter the value 12 – valid

Test case 6: Enter the value 13(12+1) – Invalid

3.

|  |  |  |
| --- | --- | --- |
| Invalid Test Case | Valid Test | Invalid Test Case |
| -1 | 0,1,9,10 | 11 |
| 10 | 11,12,19,20 | 21 |
| 20 | 21,22,29,30 | 31 |

**For shipping cast $10:**

Test case 1: Enter the value -1(0 -1) – Invalid

Test case 2: Enter the value 0 – valid

Test case 3: Enter the value 1(0+1) – valid

Test case 4: Enter the value 9(10-1) – valid

Test case 5: Enter the value 10 – valid

Test case 6: Enter the value 11(10+1) – Invalid

**For shipping cast $20:**

Test case 1: Enter the value 10(11 -1) – Invalid

Test case 2: Enter the value 11 – valid

Test case 3: Enter the value 12(11+1) – valid

Test case 4: Enter the value 19(20-1) – valid

Test case 5: Enter the value 20 – valid

Test case 6: Enter the value 21(20+1) – Invalid

**For shipping cast $30:**

Test case 1: Enter the value 20(21 -1) – Invalid

Test case 2: Enter the value 21 – valid

Test case 3: Enter the value 22(21+1) – valid

Test case 4: Enter the value 29(30-1) – valid

Test case 5: Enter the value 30 – valid

Test case 6: Enter the value 31(30+1) – Invalid

6. Prepare Decision Table Testing

1.

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Username | Password | Output |
| Case 1 | False | True | Error massage |
| Case 2 | True | False | Error massage |
| Case 3 | False | False | Error massage |
| Case 4 | True | True | Home Screen |

Case 1: username are wrong, but password was correct. The user is show an error.

Case 2: username was correct but password was wrong. Then user is show an error.

Case 3: username and password both are wrong. Then user show an error.

Case 4: username and password both are true or correct then user is redirect on home page.

2.

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Percentage | Activity | Output |
| Case 1 | <90 | >2 | Scholarship will be given |
| Case 2 | Not >90 | >2 | Not given |
| Case 3 | >90 | Not >2 | Not given |
| Case 4 | Not >90 | Not >2 | Not given |

Case 1: percentage is >90 and activity > 2 then Scholarship will be given.

Case 2: percentage is not >90 and activity > 2 then Scholarship will be not given.

Case 3: percentage is >90 and activity not > 2 then Scholarship will be not given.

Case 4: percentage is not >90 and activity not > 2 then Scholarship will be not given.