**­­**

COMSATS University Islamabad

**Abbottabad Campus**

*Project Proposal*  
UNIVERSITY ADMISSION SYSTEM

***CSC392*** *OBJECT ORIENTED SOFTWARE ENGINEERING*

*By*

**SYED SHAH HUSSAIN FA21-BSE-172**

**MAHAD WAJID FA21-BSE-057**

**ABDUL AZIZ FA21-BSE-058**

**SOMAN AHMED FA21-BSE-150**

**DANYAL NAWAZ FA21-BSE-083**

**HAMZA BADAR FA21-BSE-055**

**AJWAH SARDAR FA21-BSE-073**

*Bachelor of Science in* ***SOFTWARE ENGINEERING***

***(2021-2025)***

Supervisor: ***MUKHTIAR ZAMIN***

Contents

[**CHAPTER 1:** 5](#_Toc140605968)

[INTRODUCTION: 5](#_Toc140605969)

[VISION AND SCOPE: 5](#_Toc140605970)

[BUSINESS CASES: 6](#_Toc140605971)

[SPECIAL REQUIREMENTS / EXPECTED QUALITY MEASURES: 6](#_Toc140605972)

[RISKS WITH MEGITATION PLAN: 6](#_Toc140605973)

[**CHAPTER 2:** 7](#_Toc140605974)

[USE CASES DISTRIBUTION. 7](#_Toc140605975)

[USE CASE DIAGRAM: 8](#_Toc140605976)

[BRIEF LEVEL USE CASES: 9](#_Toc140605977)

[SYED SHAH HUSSAIN (FA21-BSE-172): 9](#_Toc140605978)

[MAHAD WAJID (FA21-BSE-057) 10](#_Toc140605979)

[HAMZA BADAR (FA21-BSE-055) 11](#_Toc140605980)

[ABDUL AZIZ (FA21-BSE-058): 12](#_Toc140605981)

[SOMAN AHMAD (FA21-BSE-150) 14](#_Toc140605982)

[AJWAH SARDAR (FA21-BSE-073) 15](#_Toc140605983)

[Daniyal Nawaz(FA21-BSE-083): 17](#_Toc140605984)

[FULLY DRESSED USE CASES WITH UI PROTOPTYPE: 18](#_Toc140605985)

[SYED SHAH HUSSAIN BADSHAH (FA21-BSE-172) 18](#_Toc140605986)

[MAHAD WAJID (FA21-BSE-057) 22](#_Toc140605987)

[HAMZA BADAR (FA21-BSE-055) 28](#_Toc140605988)

[ABDUL AZIZ (FA21-BSE-058): 32](#_Toc140605989)

[Daniyal Nawaz (FA21-BSE-083): 41](#_Toc140605990)

[SOMAN AHMAD (FA21-BSE-150) 47](#_Toc140605991)

[AJWAH SARDAR (FA21-BSE-072) 51](#_Toc140605992)

[**CHAPTER 3:** 61](#_Toc140605993)

[**Domain Model:** 61](#_Toc140605994)

[Use Case: Publish and Create Merit List HAMZA BADAR (FA21-BSE-055) 61](#_Toc140605995)

[**CHAPTER 4:** 62](#_Toc140605996)

[**SSD:** 62](#_Toc140605997)

[Use Case: Confirm Admission SYED SHAH HUSSAIN (FA21-BSE-172) 62](#_Toc140605998)

[Use Case: Notify Students SYED SHAH HUSSAIN (FA21-BSE-172) 64](#_Toc140605999)

[Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057) 65](#_Toc140606000)

[Use Case: Scheduling Test MAHAD WAJID (FA21-BSE-057) 66](#_Toc140606001)

[Use Case: Announcement MAHAD WAJID (FA21-BSE-057) 67](#_Toc140606002)

[Use Case: Take Test ABDUL AZIZ (FA21-BSE-058) 68](#_Toc140606003)

[Use Case: Print Challan ABDUL AZIZ (FA21-BSE-058) 69](#_Toc140606004)

[Use Case: Make Payment ABDUL AZIZ (FA21-BSE-058) 70](#_Toc140606005)

[Use Case: Publish merit list by (Hamza Badar FA21-BSE-055) 70](#_Toc140606006)

[Use Case: Check merit list by (Hamza Badar FA21-BSE-055) 71](#_Toc140606007)

[**CHAPTER 5:** 72](#_Toc140606008)

[**Operation Contracts:** 72](#_Toc140606009)

[Use Case: Confirm Admission SYED SHAH HUSSAIN (FA21-BSE-172) 72](#_Toc140606010)

[Use Case: Notify Students SYED SHAH HUSSAIN (FA21-BSE-172) 73](#_Toc140606011)

[Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057) 73](#_Toc140606012)

[Use Case: Scheduling Test MAHAD WAJID (FA21-BSE-057) 74](#_Toc140606013)

[Use Case: Announcement MAHAD WAJID (FA21-BSE-057) 74](#_Toc140606014)

[Use Case: Take Test ABDUL AZIZ (FA21-BSE-058) 75](#_Toc140606015)

[Use Case: Print Challan ABDUL AZIZ (FA21-BSE-058) 75](#_Toc140606016)

[Use Case: Make Payment ABDUL AZIZ (FA21-BSE-058) 76](#_Toc140606017)

[Use Case: Publish merit list by (Hamza Badar FA21-BSE-055) 76](#_Toc140606018)

[Use Case: Check merit list by (Hamza Badar FA21-BSE-055) 76](#_Toc140606019)

[Use Case: LOGIN by Ajwa Sardar(FA21-BSE-073) 77](#_Toc140606020)

[Use Case: ENTER ACADEMIC INFO BY SOMAN AHMAD(FA21-BSE-150) 77](#_Toc140606021)

[Use Case: VERIFY DOCUMENT BY SOMAN AHMAD(FA21-BSE-150) 78](#_Toc140606022)

[**CHAPTER 6:** 79](#_Toc140606023)

[**Package Diagram:** 79](#_Toc140606024)

[**CHAPTER 7:** 80](#_Toc140606025)

[**Class Diagram:** 80](#_Toc140606026)

[**CHAPTER 8:** 81](#_Toc140606027)

[**Sequence Diagram:** 81](#_Toc140606028)

[Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057) 81](#_Toc140606029)

[Use Case: Announcement BY MAHAD WAJID (FA21-BSE-057) 82](#_Toc140606030)

[Use Case: Take Test ABDUL AZIZ (FA21-BSE-058) 83](#_Toc140606031)

[Use Case: Publish Merit list (FA21-BSE-055) 84](#_Toc140606032)

[Use Case: Check Merit list (FA21-BSE-055) 84](#_Toc140606033)

[Use Case: admission confirmation SYED SHAH HUSSAIN (FA21-BSE-172) 86](#_Toc140606034)

[Use Case: notify Students SYED SHAH HUSSAIN (FA21-BSE-172) 87](#_Toc140606035)

# **CHAPTER 1:**

## INTRODUCTION:

As the number of students appearing for the counseling are increasing rapidly every year, it requires much effort and time to handle the admission system with man power and paper system. So we are in need of a better system to make the process easier and serves better which could be done by Computerized Student Admission System that facilitates the work of the universities and at the same time it must reduce the work load of the organization with expected quality. Quality in the sense, the system tries to avoid the mistakes that are usually happen during the Admission Process. The University Admission System has been developed in order to automate the complete admission system starting from the notification to admission process.

The system enables online admissions saving the time of the geographically scattered students. It enables reducing time in activities, centralized data handling and paperless admission with reduced manpower. It improves the operational efficiency and reduces the cost. It also provides consist view of data and integration with other institutions for verification of marks and details.

## VISION AND SCOPE:

**Vision:**

The vision of a university admission system is to provide an accessible, fair, and efficient process for selecting and admitting the most qualified candidates to the academic programs offered by the university. The system should ensure that every applicant is given equal consideration and opportunities for success, regardless of their background, ethnicity, gender, or financial status.

**Scope:**

The scope of a university admission system includes the following:

**Application process:** This involves developing and implementing an online or paper-based application process that includes all necessary fields and information required to evaluate the applicant's qualifications.

**Evaluation process:** This includes the development of evaluation criteria that are aligned with the academic program's requirements, and the review of applications by the admissions committee or evaluators. The evaluation process may also include interviews, essays, and letters of recommendation.

**Admission decision**: This involves making the final admission decision based on the evaluation results and the requirements of the academic program.

**Communication and documentation:** This includes notifying applicants of the admission decision and providing any additional documentation required for enrollment, such as transcripts, financial aid forms, or health records.

**Continuous improvement:** The university admission system should continuously monitor and evaluate its processes and outcomes, and make improvements as necessary to ensure that it remains effective and aligned with the university's mission and goals.

## BUSINESS CASES:

## SPECIAL REQUIREMENTS / EXPECTED QUALITY MEASURES:

## RISKS WITH MEGITATION PLAN:

# **CHAPTER 2:**

## USE CASES DISTRIBUTION.

|  |  |  |
| --- | --- | --- |
| Name | Reg No | Assign Use cases |
| Hamza Badar | Fa21-bse-055 | 1. Published Merit list  2.. Check Merit List |
| Mahad Wajid | Fa21-bse-057 | 1.Enter Personal info  2.Announcement  3.Scheduling test |
| Abdul Aziz | Fa21-bse-058 | 1.Take Test  2. Print Challan  3. Make Test Payment  4.Make Admission payment |
| Soman Ahmed | Fa21-bse-150 | 1.Logout,  2.Academic info  3.verify document |
| Daniyal Nawaz | Fa21-bse-083 | 1.Login  2.Apply for test  3. Apply For Admission |
| Shah Hussain | Fa21-bse-172 | 1.Notify  2.Admission Confirmation |
| Ajwa Sardar | Fa21-bse-072 | 1.Registration  2.Arrange Test  3.Admission Criteria |

## USE CASE DIAGRAM:

­­­

## BRIEF LEVEL USE CASES:

### SYED SHAH HUSSAIN (FA21-BSE-172):

#### Use Case: NOTIFY STUDENTS

The university admission system retrieves the Merit List, Fee Submission, and Admission Confirmation details from the database. It then notifies the students about their Merit List status, reminds them about the fee submission deadline, and informs them about the admission confirmation process. By directly retrieving and notifying students based on the database records, the system ensures efficient and accurate communication throughout the admission process.

#### Use Case: CONFIRM ADMISSION

The university admission system checks the list of students who have paid the fee. It identifies the students who have successfully completed the fee submission process. Those students are then marked as "Admission Confirmed" in the system. The system automatically notifies the confirmed students about their admission status, providing them with further instructions and relevant information. This streamlined process ensures that only the students who have paid the fee are admitted and promptly informed, enhancing efficiency and communication within the admission process.

### MAHAD WAJID (FA21-BSE-057)

#### Use Case: Enter Personal Info

1. **Application processing**: Personal information is required to process and evaluate applications for admission. This includes basic information such as name, address, contact details, and academic records.
2. **Communication:** Personal information such as email address and phone number is needed to communicate with student about their application status, interviews, and other related matters.
3. **Financial aid:** Personal information is used to determine eligibility for financial aid, scholarships, and other forms of financial assistance. This includes information about income, assets, and family background.
4. **Accommodation**: Personal information is used to determine accommodation options for students, including preferences for living arrangements and any special needs.
5. **Health and safety:** Personal information is required for health and safety purposes, such as medical history, emergency contacts, and insurance information.
6. **Diversity and inclusion**: Personal information is used to promote diversity and inclusion in the university community, such as gathering information about an applicant's race, ethnicity, or socioeconomic background.

#### Use Case: Scheduling Test

1. **Standardized testing:** Many universities require applicants to take standardized tests such as the SAT, ACT, GRE, or GMAT. Test schedules are used to inform applicants of upcoming test dates, registration deadlines, and other important information related to the testing process.
2. **Subject-specific testing**: Some universities may require students to take subject-specific tests in addition to standardized tests. Test schedules are used to inform applicants of the specific tests required, the dates and locations of the tests, and any registration deadlines.
3. **Language proficiency testing:** For international students, universities may require proof of English proficiency through tests such as TOEFL or IELTS. Test schedules are used to inform students of the testing process, including available test dates, registration deadlines, and test fees.
4. **Accommodation for test takers:** Test schedules are used to ensure that appropriate accommodations are provided for test takers with disabilities or other special needs, such as extended testing time or special testing conditions.
5. **Test score submission deadlines:** Universities may have specific deadlines for receiving test scores from applicants. Test schedules are used to inform students of these deadlines, allowing them to plan accordingly and ensure that their scores are submitted on time.
6. **Alternate testing arrangements**: In some cases, universities may offer alternate testing arrangements for students who are unable to take the test on the scheduled date, such as makeup exams or remote testing options. Test schedules are used to inform students of these options and the associated deadlines and requirements.

#### Use Case: Announcement

1. **Application deadlines:** Universities may use announcements to inform prospective students of application deadlines, including early decision or early action deadlines, regular decision deadlines, and any applicable extension deadlines.
2. **Admission decision notifications:** Universities may use announcements to inform applicants of admission decisions, including acceptance, rejection, or waitlist status. These notifications may include instructions on next steps for admitted students, such as submitting a deposit or registering for classes.
3. **Scholarship and financial aid announcements**: Universities may use announcements to inform students of scholarship and financial aid opportunities, including deadlines for submitting applications and instructions for how to apply.
4. **Open house and campus visit announcements:** Universities may use announcements to promote campus visits and open house events, providing information on dates, times, and registration details.
5. **Program updates and changes:** Universities may use announcements to inform students of changes or updates to academic programs or requirements, as well as new program offerings or changes to admission requirements.
6. **Important dates and deadlines**: Universities may use announcements to highlight important dates and deadlines related to the admission process, such as when transcripts or test scores are due, or when students can expect to receive their admission decision.

### HAMZA BADAR (FA21-BSE-055)

#### Use Case: Published Merit List

1. **Generate Merit List:** This use case involves generating the merit list of admitted students based on the evaluation results. The system sorts the applications based on the criteria specified by the university and creates a list of students who have been admitted to the university.
2. **Notify Admitted Students:** This use case involves notifying the admitted students about their acceptance to the university. The system sends out notifications to the email addresses or phone numbers provided by the students, informing them of their admission status and any further steps they need to take.
3. **Publish Merit List:** This use case involves publishing the merit list on the university website. The system creates a webpage or PDF document containing the list of admitted students and makes it accessible to the public.
4. **Update Student Records:** This use case involves updating the student records in the university's student information system. The system adds the admitted students to the database, assigns them a student ID, and updates their personal and academic information.
5. **Manage Waitlist:** This use case involves managing the waitlist of students who have not yet been admitted. The system adds the remaining students to a waitlist and notifies them of their waitlist status. As spots become available, the system evaluates the students on the waitlist and admits them if they meet the admission criteria.
6. **Review Merit List:** This use case involves reviewing the merit list for accuracy and completeness. The system allows the admin to review the list and make any necessary corrections or adjustments.

#### Use Case: Check Merit List

1. **View Merit List:** This use case involves allowing users to view the published merit list of the admitted students. The system displays the merit list on the university website or provides a link to download the list in a readable format.
2. **Search Merit List:** This use case involves allowing users to search the merit list for specific students or criteria. The system allows users to search the merit list based on student name, program, or other relevant criteria.
3. **Check Application Status:** This use case involves allowing users to check the status of their application. The system retrieves the user's application status from the admission database and displays it to the user.
4. **Notify Waitlisted Students:** This use case involves notifying waitlisted students of any changes in their status. The system sends out notifications to the email addresses or phone numbers provided by the students, informing them of any updates to their waitlist status.
5. **Send Acceptance Letter:** This use case involves sending an acceptance letter to the admitted students. The system generates an acceptance letter with the relevant details and sends it to the admitted student's email or physical address.

### ABDUL AZIZ (FA21-BSE-058):

#### Use Case: Take Online Test

1. The Student logs into the university admission system using their credentials.
2. The system presents the available tests and the Student selects the one they want to take.
3. The system checks if the Guest meets the prerequisites for the selected test, such as Fsc marks, and documents verification.
4. If the Student meets the prerequisites, the system presents the test schedule to the Student.
5. The Student takes the test, answering the questions presented to them.
6. The Student submits the completed test to the test proctor and waiting for results.

#### Use Case: Print Challan

1. The student click on the print challan option presented by the system on Student’s dashboard.
2. The system calculates the fee based on the student program and any applicable discounts or scholarships.
3. The system generates a unique challan number and associates it with the student account.
4. The system presents the challan to the student, including the fee amount, the due date, and the payment instructions.
5. The student can choose to print the challan or save it as a PDF.
6. If the Student chooses to print the challan, the system generates a formatted document with the necessary information, including the challan number and a barcode for tracking.
7. The student then print challan.

#### Use Case: Make Test Payment

1. The system presents the available test payment methods, such as bank transfer or pay by challan.
2. The student selects the preferred test payment method and enters the required payment information, such as account number or printed challan.
3. The system simulates the payment information and notify the student to verify and confirm its payment information.
4. The Student can then analyze the response and make any necessary changes to the payment processing and verification functionalities in the code.
5. Once all payment functionalities are completed, the system shows a Thank You message to the Student as a confirmation.

#### Use Case: Make Admission Payment

1. The system presents the student with the total amount due for admission fees, along with any applicable taxes.
2. The student selects the preferred payment method, such as, bank transfer, or by challan.
3. The student enters the required payment information, such as bank account number or challan number.
4. The system verifies the payment information and processes the payment.
5. If the payment is successful, the system generates a payment receipt and associates it with the student account.
6. The system updates the student account to indicate that the payment has been received and processed and that the student has been officially admitted.
7. If the payment is unsuccessful, the system presents an error message and allows the student to try again or select a different payment method.

#### **Contact Admission Office:** This use case involves allowing users to contact the admission office for any questions or concerns regarding their application or admission status. The system provides contact information for the admission office, such as phone numbers and email addresses

### SOMAN AHMAD (FA21-BSE-150)

**BRIEF LEVEL USE CASE OF LOG OUT OF UNIVERSITY ADMISSION SYSTEM:**

1. The user logs in to the university admission system using their username and password.
2. After the user has completed their tasks within the system, they want to end their session and log out to prevent unauthorized access to their account.
3. The user clicks on the "log out" button, which is typically located in the upper right corner of the screen.
4. The system displays a confirmation message asking the user if they are sure they want to log out.
5. The user confirms their intention to log out by clicking on the "yes" button.
6. The system ends the user's session and returns them to the login screen.
7. The user is no longer able to access any of the features or functionality of the system until they log in again.

**BRIEF LEVEL USE CASE OF ACADEMIC INFO OF UNIVERSITY ADMISSION SYSTEM:**

1. The user logs in to the university admission system using their username and password.
2. The user navigates to the "Academic Info" section of the system.
3. The system displays the user's academic record, which may include information such as courses taken, grades received, and degree progress.
4. The user may have the ability to update their academic information or view additional details by clicking on links or buttons within the system.
5. The user may also be able to access information related to academic requirements, such as degree program requirements, course prerequisites, or graduation requirements.
6. The user may be able to download or print their academic information for their own records or to provide to others.
7. If the user encounters any issues or discrepancies in their academic information, they may be able to request assistance or submit a support ticket to the university administration.

**BRIEF LEVEL USE CASE OF VERIFY DOCUMENT OF UNIVERSITY ADMISSION SYSTEM:**

1. Prospective students submit their application online and upload scanned copies of their academic transcripts, certificates, and other relevant documents.
2. The admission system receives the documents and sends them to a document verification service for validation.
3. The document verification service checks the documents against its database of known valid documents and employs various security measures such as image analysis and watermark detection to ensure their authenticity.
4. The verification service then generates a report indicating whether each document is verified or not and the level of confidence in the verification.
5. The admission system receives the verification report and uses it to determine the eligibility of the prospective students.
6. If a document is found to be invalid or inaccurate, the admission system can reject the application or request the applicant to provide additional supporting documents.

### AJWAH SARDAR (FA21-BSE-073)

#### SIGNING UP FOR ADMISSION SYSTEM

#### Brief Use Cases:

Here are some brief use cases for the sign-up process of an admission system for a university:

1. Prospective students navigate to the university admission system website.
2. Prospective students click the "Sign up" button on the admission system homepage.
3. System prompts prospective students to enter personal information, such as name, email, and password.
4. Prospective students enter personal information and click the "Submit" button.
5. System validates personal information to ensure it meets required format and is not already registered.
6. If validated successfully, the system creates a new account for prospective students.
7. System sends confirmation emails to prospective student's email addresses.
8. Prospective students click the confirmation link in the email and are redirected to the admission system.
9. System displays a confirmation message, informing prospective students that their account has been created.
10. If there are errors or issues with the sign-up process, the system displays error messages and prompts the prospective student to correct the information or take other actions.

These use cases describe the main steps involved in the sign-up process for an admission system at a university, and can be further elaborated upon in a fully dressed use case.

#### INTAKE ACADEMIC INFORMATION

#### **Brief Use Case**

The academic information intake form page of a university admission system serves as a platform for prospective students to provide detailed academic information that may be required during the admission process. Some of the use cases of this page include:

1. Collecting Academic Information: The academic information intake form collects details of the prospective student's academic qualifications, including their GPA, standardized test scores, and transcripts.
2. Course Preferences: The page may also ask the student to provide their course preferences and select the program they are interested in pursuing.
3. Additional Information: The academic information intake form may also collect additional information, such as extracurricular activities, work experience, and personal statements.
4. Admission Eligibility: Based on the information provided, the system can determine the student's eligibility for admission to the program they have applied for.
5. Academic Evaluation: The information collected on this page can be used to evaluate the academic profile of the student and compare it with other applicants.
6. Application Processing: The academic information intake form provides the admission office with the necessary information to process the application and make a decision on admission.
7. Scholarship and Financial Aid: The page may also ask the student to provide details of any scholarship or financial aid they may require to pursue their studies.

#### VIEWING ELIGIBILITY CRITERIA

#### **Brief Use Case**

The eligibility criteria page of a university admission system serves as a platform for prospective students to understand the admission requirements and criteria they need to meet before applying for a particular program. Some of the use cases of this page include:

1. Clear Understanding of Admission Requirements: The eligibility criteria page provides a clear understanding of the academic qualifications and other requirements necessary to be eligible for admission to a particular program.
2. Standardized Tests: The page may specify the standardized tests required for admission, such as SAT, ACT, GRE, or GMAT, and the minimum score required.
3. Course Prerequisites: The eligibility criteria page may also specify the course prerequisites that the prospective student must have completed in high school or college.
4. Language Proficiency: The page may specify the language proficiency requirements for international students, such as TOEFL, IELTS, or other language proficiency tests.
5. Residency Requirements: The eligibility criteria page may specify the residency requirements for admission, such as the state or country of residence, citizenship, or visa status.
6. Age Limit: The eligibility criteria page may specify the age limit for admission, especially for programs that have an age limit.
7. Program Specific Criteria: The eligibility criteria page may also provide specific criteria for certain programs, such as work experience, portfolio, or auditions.
8. Avoiding Wasted Applications: The eligibility criteria page helps prospective students avoid wasting time and effort on applications if they do not meet the admission requirements.

### Daniyal Nawaz(FA21-BSE-083):

#### BRIEF LEVEL USE CASE OF LOGIN:

1. User opens the admission system login page and enters their credentials (username and password).
2. The system verifies the user's credentials and checks if the user is authorized to access the admission system based on their role and access level.
3. If the credentials are correct and the user is authorized, the system logs the user into the admission system and displays the appropriate dashboard or landing page based on the user's role.
4. If the credentials are incorrect, the system displays an error message and prompts the user to enter their credentials again or reset their password if necessary.
5. Once logged in, the user can perform various actions such as submitting applications, reviewing applications, managing student records, etc. based on their role and level of access.
6. If the user logs out of the system or if their session times out, the system logs the user out and redirects them to the login page to re-authenticate.

#### BRIEF LEVEL USE CASE OF APPLY FOR ADMISSION:

1. Prospective student opens the admission system application page and starts filling out the application form.
2. The system prompts the student to enter their personal information, educational background, test scores, and other required information.
3. The system validates the student's application and prompts them to review and verify the accuracy of the information entered.
4. Once the student has reviewed and verified their application, they submit the application to the university.
5. The system sends a confirmation email to the student, acknowledging receipt of the application and providing them with further instructions and updates about the application process.
6. The university admission office receives the application and reviews it to determine the student's eligibility for admission.
7. The admission office may request additional information or documents from the student to complete the application process.
8. The admission office makes a decision on the application, and the system sends an email notification to the student about the decision.
9. If the student is accepted, the system provides them with further instructions on how to enroll in the university.
10. If the student is rejected, the system provides them with information on the reasons for the rejection and any available options for appeal or reapplication.

#### BRIEF LEVEL USE CASE OF APPLY FOR TEST:

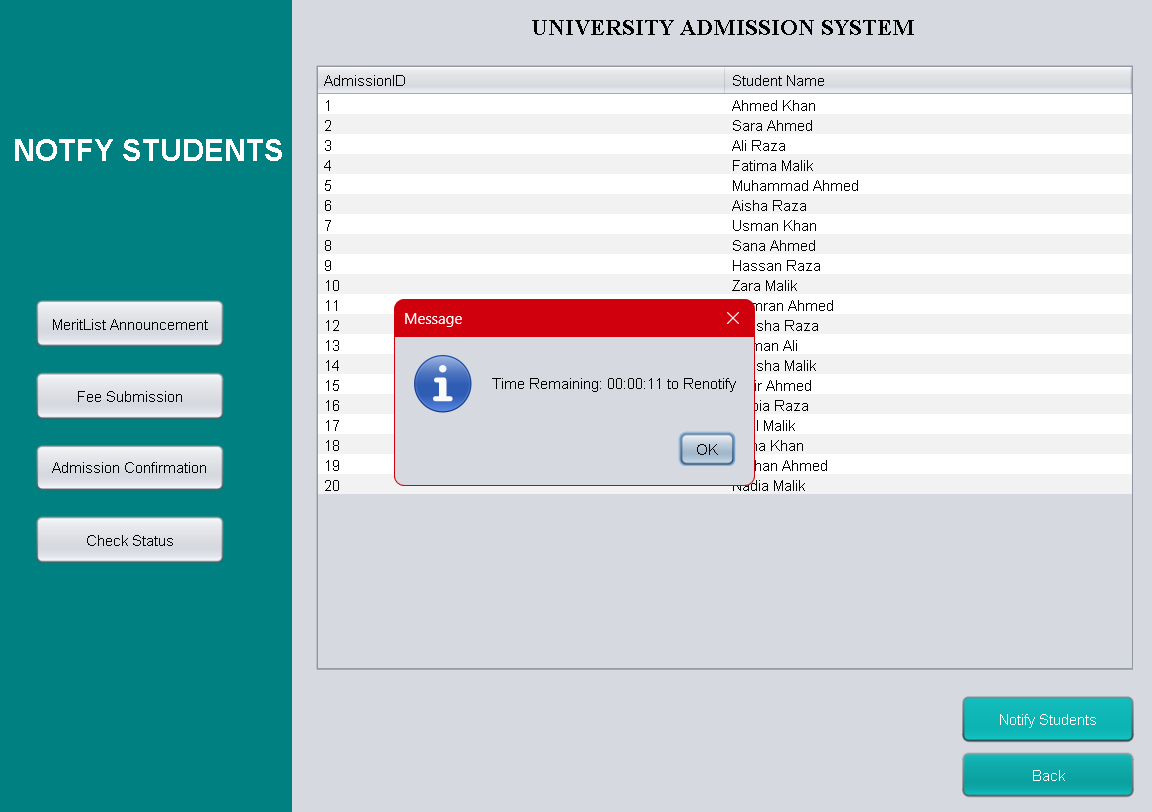
1. Prospective student logs in to the admission system and opens the apply for test page.
2. The system prompts the student to select the type of test they wish to take (e.g., SAT, ACT, GRE, etc.).
3. The system displays the available test dates, locations, and times for the selected test.
4. The student selects their preferred test date, location, and time.
5. The system prompts the student to pay the test registration fee.
6. The student pays the fee through the admission system's integrated payment gateway.
7. The system confirms the student's test registration and sends them an email confirmation with the test details and instructions on what to bring to the test center.
8. On the day of the test, the student arrives at the test center and presents the necessary identification and materials as instructed by the admission system.
9. The student takes the test and receives their score report through the admission system.
10. The university admission office receives the student's test scores and uses them as part of the application evaluation process.

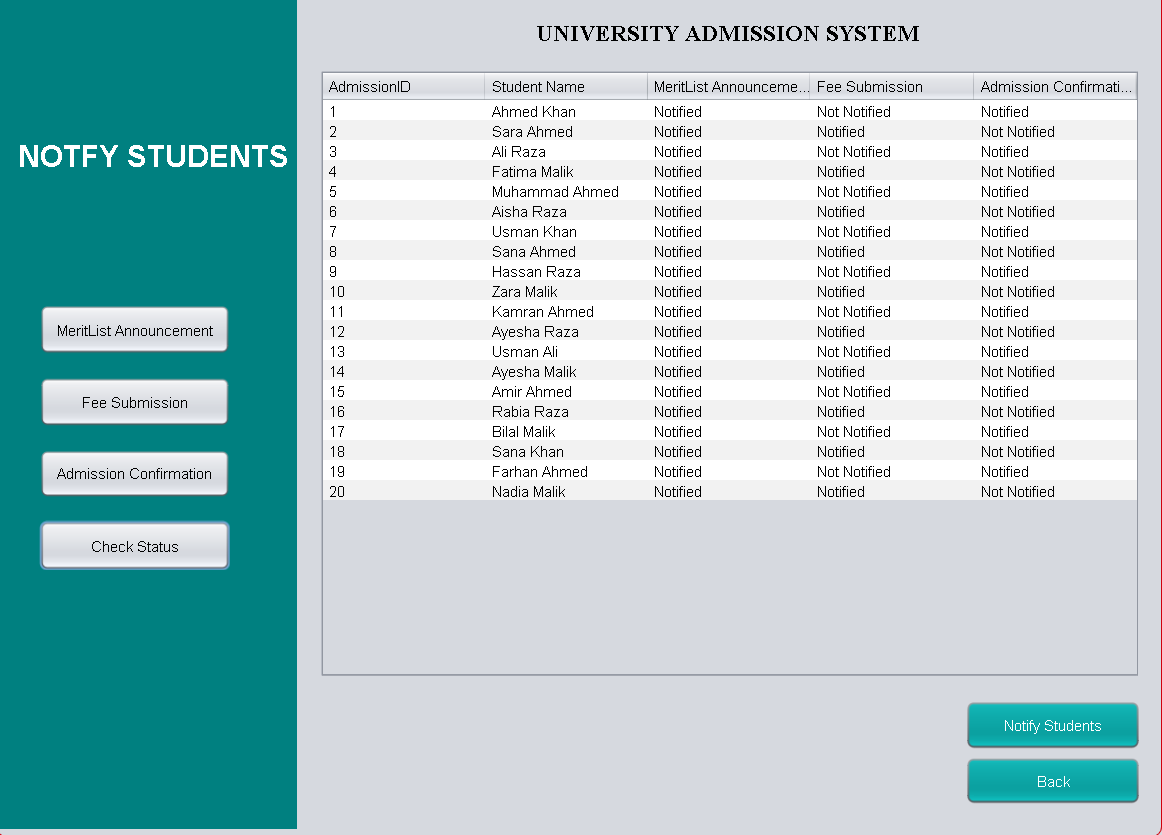
## FULLY DRESSED USE CASES WITH UI PROTOPTYPE:

### SYED SHAH HUSSAIN BADSHAH (FA21-BSE-172)

#### Use Case: Notify

|  |  |
| --- | --- |
| Use Case Name | Notify Students |
| Scope | This use case covers the process of notifying potential applicants of upcoming admissions for the university. |
| Primary Actors | University Administrators, Students |
| Stakeholders | University Administrators, Prospective Applicants |
| Pre-Conditions | The university has upcoming admissions, and the university administrators have prepared the admission advertisement. |
| Main Success Scenario | * + - 1. On Admin Dashboard clicks on Notify button & system navigate him to Notify Students Interface.       2. Admin Click on Merit List announcement button and System will provide/show List of all Students.       3. An Admin then click on Notify Students button which will add those students to the Notified (check Status) table in Database & notify the Students and show a message how much students are notified.       4. Admin Click on Fee submission button and System will Display/show List of those students who have not yet paid a Fee.       5. An Admin then click on Notify Students button which will add those students to the Notified (check Status) table in Database & notify the Students and show a message how much students are notified.       6. Admin Click on Admission Confirmation button and System will Display/show List of those students who have paid Fee.       7. An Admin then click on Notify Students button which will add those students to the Notified (check Status) table in Database & notify the Students and show a message how much students are notified.       8. An Admin then Click on check Status to check which student is notified regard which type notification and System Display all students in Table form with status notified or not notified.       9. And if Step 3, 5, 7 are performed and admin wants to Notify the Students again there will be time Restriction if specified time passed already then admin can re-notify the Students otherwise Notify functionality will remain restricted and show message Time remaining to re-notify and when remaining time completed system will display a message “you can notify now”. |
| Alternative Scenarios | - If there are technical issues with the admission system, the administrators will contact the IT department to resolve the issue.  - If the notification fails to reach the target audience, the administrators will investigate the issue and take appropriate actions to resolve it. |
| Success Guarantee | The notification process will be successful, and the target audience will receive the admission advertisement. |
| Expectations | The university administrators expect to reach a wide audience of potential applicants and to receive a high number of applications as a result of the admission advertisement. |
| Frequency of Occurrence | This use case occurs periodically, depending on the admission cycles of the university. |
| Technology and Data Variation List | - Operating System: Windows, Mac OS, Linux  - Web Browser: Chrome, Firefox, Safari, Edge  - Communication Channels: Email, Social Media, SMS |
| Prototype | A prototype of the notify feature will be developed to test the use case. |
| Miscellaneous | None. |
| Use Case Section | Admission |
| Special Requirements | The admission advertisement should include all necessary information for prospective applicants to apply for admission, including any required documents and deadlines. The notification should also be personalized and relevant to the target audience to increase the chances of attracting qualified applicants. |





#### Use Case: Confirm Admission

|  |  |
| --- | --- |
| Use Case Name | Confirm Admission |
| Scope | This use case covers the process of confirming admission for accepted applicants in the university admission system. |
| Primary Actors | Selected Applicants |
| Stakeholders | Selected Applicants, University Administrators |
| Pre-Conditions | The applicant must have been accepted into the university through the admission system. |
| Main Success Scenario | 1. Admin will retrieve the list of selected Students from merit list.  2. Admin then will check feeStatus of Selected students.  3. Admin will confirm the admission of those students whose fee status is paid/cleared.  4. An Admin can also modify the feeStatus of students Manually. |
| Alternative Scenarios | - If the applicant is unable to log in, they will contact the IT department for assistance.  - If there are any technical issues with the admission system, the applicant will contact the IT department to resolve the issue. |
| Success Guarantee | The admission confirmation process will be successful, and the applicant will receive confirmation of their admission to the university. |
| Expectations | The applicant expects to receive confirmation of their admission and to have clear instructions for the enrollment process. |
| Frequency of Occurrence | This use case occurs frequently during the enrollment period. |
| Technology and Data Variation List | - Operating System: Windows, Mac OS, Linux  - Web Browser: Chrome, Firefox, Safari, Edge  - Data Format: PDF, Email, Text Message |
| Prototype | A prototype of the admission confirmation process will be developed to test the use case. |
| Miscellaneous | None. |
| Use Case Section | Admission |
| Special Requirements | The admission confirmation letter should include all necessary information for the applicant to complete the enrollment process, including any required documents and deadlines. |



### MAHAD WAJID (FA21-BSE-057)

#### Use Case: Enter Personal Info

|  |  |
| --- | --- |
| Use case name | Personal Info |
| Scope | The Personal Info use case describes the functionality of the university admission system for gathering and managing the personal information of prospective students. |
| Main Success Scenario | |  |  | | --- | --- | | Student | System | | The student navigates to the Project University Admission system and clicks on the "Enter Personal Info" button. |  | |  | The system presents a form to the user, asking for personal information such as name, address, phone number, email address, and any other relevant details. | | The student fills out the form with accurate and complete information and submits it. |  | |  | The system verifies that all required fields have been filled out correctly and prompts the user to confirm their submission. | | The student confirms the submission |  | |  | The system displays a confirmation message indicating that the student's personal information has been successfully recorded. | |  | The system then redirects the user to the next step in the admission process, such as submitting transcripts or completing an application essay. | |
| Pre-Condition | The prospective student must have applied for admission to the university. |
| Primary Actors | Student |
| Level | Level 1: Basic Level: The student enters their personal information into the university admission system.  Level 2: Intermediate Level: The system validates the entered data and checks for any errors or missing information.  Level 3: Advanced Level: The system saves the information in the database and displays a confirmation message to the student. |
| Stake Holders | Prospective students: The primary stakeholders who benefit from the Personal Info use case by having their personal information accurately recorded and stored in the university admission system.  University administrators: The secondary stakeholders who benefit from the Personal Info use case by having access to accurate and up-to-date personal information for each prospective student. |
| Success Guarantee | The personal information of the prospective student is accurately recorded and stored in the university admission system. |
| Exceptions | The university admission system should be user-friendly and easy to navigate.  The system should validate the entered data and check for any errors or missing information.  The personal information of the student should be securely stored in the database.  The system should provide clear and concise feedback to the student on the status of their personal information. |
| Special Requirements | The university admission system must comply with all applicable data privacy laws and regulations. |
| Technology and data variation list | Technology: The university admission system is a web-based application that requires a modern web browser and an internet connection.  Data Variation: The personal information entered by the student may vary in format and content, depending on the student's background and cultural norms. |
| Frequency of occurrences | This use case occurs for each new prospective student who applies for admission to the university. |
| Miscellaneous | The Personal Info use case is one of several use cases in the university admission system, including Application Submission, Transcript Submission, Test Scores Submission, and Admissions Decision. |
| Prototyping | A prototype of the university admission system with the Personal Info use case can be developed and tested with a small group of prospective students before the system is deployed for the entire applicant pool. |

#### Interface:



#### Use Case: Scheduling Test

|  |  |
| --- | --- |
| Use case name | Test Schedule |
| Scope | The Test Schedule use case describes the functionality of the university admission system for managing and scheduling entrance exams for prospective students. |
| Main Success Scenario | * The guest navigates to the Project University Admission system and clicks on the "Schedule Test" button. * The system presents a list of available test dates and times for the user to choose from. * The guest selects a test date and time that works for them and confirms their selection. * The system verifies that the selected test date and time are available and prompts the user to confirm their reservation. * The guest confirms the reservation and the system displays a confirmation message indicating that their test has been successfully scheduled. * The system then provides the user with any additional instructions or details they may need to know before taking the test, such as the location of the test center or what materials to bring. |
| Pre-Condition | The prospective student must have applied for admission to the university and have completed all the necessary requirements to be eligible to take the entrance exam. |
| Primary Actors | Admin, students |
| Level | Level 1: Basic Level: The admission staff member schedules the entrance exam for the prospective student.  Level 2: Intermediate Level: The system displays a list of available dates and times for the exam.  Level 3: Advanced Level: The system confirms the date and time of the exam and notifies the student via email or other means. |
| Stake Holders | Admission staff - responsible for scheduling and managing entrance exams for prospective students.  Prospective students - responsible for attending the entrance exam as per the scheduled date and time. |
| Success Guarantee | The entrance exam is scheduled for the student on a convenient date and time, and the student is notified of the exam details. |
| Exceptions | If the admission staff member encounters any technical difficulties while scheduling the exam, they can contact the university helpdesk for assistance.  If the student is unable to attend the scheduled exam, they can contact the university to reschedule for a different date and time.  If there is an error with the database or system, the system will display an error message and prompt the staff member to try again later. |
| Special Requirements | The system should allow the staff member to view the scheduled exam details and make changes if necessary.  The system should be able to generate reports on the exam scheduling and attendance for administrative purposes. |
| Technology and data variation list | Technology: The university admission system is a web-based application that requires a modern web browser and an internet connection.  Data Variation: The list of available exam dates and times may vary based on the availability of exam proctors and facilities. |
| Frequency of occurrences | This use case occurs for each eligible student who has completed all the necessary admission requirements to take the entrance exam. |
| Miscellaneous | The university admission system should have a feature to automatically reschedule an exam if a student fails to attend the scheduled exam.  The system should have an option to cancel an exam and notify the student if the exam cannot be held due to unforeseen circumstances such as natural disasters or system outages.  The system should have a mechanism to prevent overbooking of exam dates and times to ensure the smooth conduct of exams. |
| Prototyping | A prototype of the Test Schedule use case can be developed to demonstrate the following functionalities:  The ability to display a list of eligible students who have completed all the necessary admission requirements to take the entrance exam.  The ability to select a student from the list and schedule an entrance exam for them.  The ability to display a list of available exam dates and times.  The ability to confirm the exam date and time and notify the student via email or other means.  The ability to handle errors and exceptions gracefully.  The ability to generate reports on the exam scheduling and attendance for administrative purposes. |

#### Use Case: Announcement

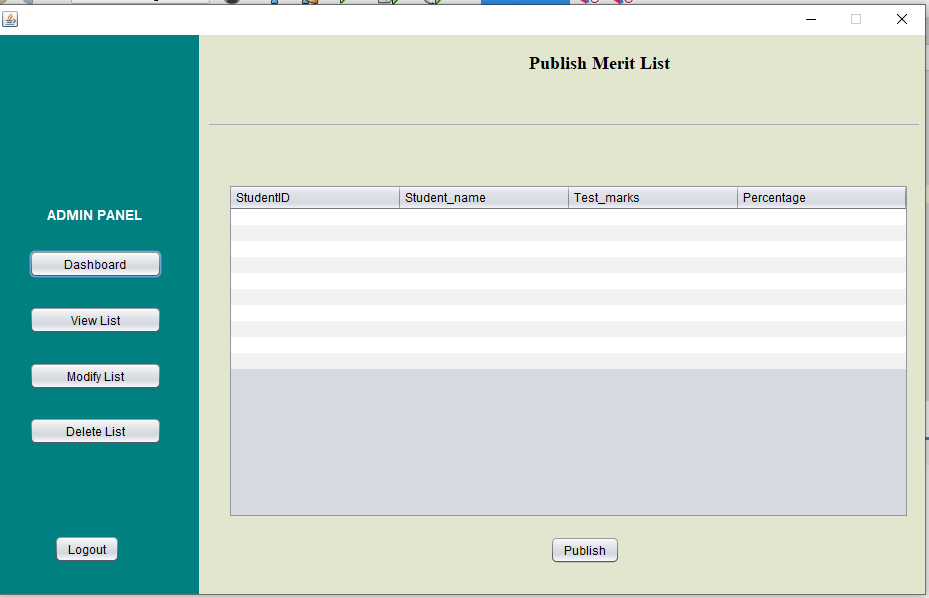
|  |  |
| --- | --- |
| Use Case Name: | Announcement |
| Scope: | University Admission System |
| Primary Actors: | Admin, Students |
| Stakeholders and Interests: | Admission Officer: Need to inform prospective students about the admission process and deadlines. Prospective Students: Need to receive timely and accurate information about the admission process to make informed decisions. |
| Precondition: | The admission officer has access to the announcement section of the admission system. |
| Success Guarantee: | Prospective students receive timely and accurate information about the admission process and deadlines. |
| Main Success Scenario: | * The administrator navigates to the announcement section. * The administrator creates a new announcement with a title, description, and deadline. The admission officer publishes the announcement to the admission system. * The prospective guest navigate to the announcement section. * The prospective guest read the announcement and take appropriate action. |
| Extensions: | If the admission officer fails to create a new announcement, the system displays an error message and the use case ends. If the admission officer fails to publish the announcement, the system displays an error message and the use case ends. If the prospective students fail to read the announcement, the system sends a reminder to their registered email address. |
| Frequency of Occurrence: | Multiple times a year, depending on the admission cycle. |
| Technology and Data Variation List: | The admission system should support multiple languages for the announcement text. The system should be accessible from different devices such as desktops, laptops, tablets, and smartphones. The system should be able to handle a large number of concurrent users during peak admission periods. |
| Prototype: | A prototype of the Announcement use case can be developed to demonstrate the following functionalities The ability to create and publish announcements. The ability to display announcements in a user-friendly format. The ability to send email reminders to students who have not read the announcements. |
| Special Requirements: | The announcement section should be easy to navigate and accessible to all prospective students. |
| Level: | User Goal |
| Use Case Section: | Main Flow, Extensions |
| Miscellaneous: | The admission officer should be able to edit or delete announcements if necessary. The system should have a mechanism to ensure that the announcements are displayed in chronological order. The system should be able to track the number of views and clicks on each announcement for administrative purposes. |

### HAMZA BADAR (FA21-BSE-055)

#### Use Case: Published Merit List

|  |  |
| --- | --- |
| Use case name | Publish merit list |
| Scope | This use case describes the actions and interactions involved in publishing the merit list of admitted students in the University Admission System. The merit list contains the names of students who have been admitted to the university, sorted by their academic achievements. |
| Primary actors | Admin |
| Pre-condition | The admin user has logged into the University Admission System.  Result list is in the system. |
| Main success scenario | 1. The admin user navigates to the “View List". 2. The system gives a list of all student results. 3. The admin reviews the list. 4. If data is irrelevant admin will change the data. 5. Then navigates to “Modify” button and modifies the data. 6. If admin wants to delete some data, the admin selects the specific data for deletion. 7. Then navigates to “delete” button and delete the data. 8. The admin navigates to “Publish” button and publishes the list. |
| Level | User goal. |
| Stakeholders and interest | **Admission Committee**: Needs to evaluate applications, calculate merit scores, and publish the merit list to make admission decisions.  **Applicants:** Expect timely and accurate updates on their application status and admission chances.  **University Staff:** Need to access student information for various administrative tasks, such as financial aid, registration, and housing. |
| Success guarantee | The merit list accurately reflects the admissions decisions and is published in a timely manner.  Admitted students receive timely notifications of their admission status. |
| Exceptions | If the university admission System is not operational, the use case cannot proceed.  If the list of students results is not available , then the admin cannot published the list. |
| Special requirement | The University Admission System must be secure and protect student information from unauthorized access or disclosure.  The system must be able to handle a large volume of admission decisions and students. |
| Technology and data variation list | The University Admission System may use different technologies for publishing the merit list on the university website.  The system may need to convert admission decision data into a format that is suitable for the merit list. |
| Frequency of occurrences | The Publish Merit List use case occurs once per admission cycle, which may occur multiple times per year. |
| Miscellaneous | The system should be able to handle large amounts of data efficiently.  The system should be user-friendly and easy to navigate.  The system should be able to generate a timetable within a reasonable amount of time. |
| Prototyping | A prototype of the University Admission System has been developed and is undergoing testing. |

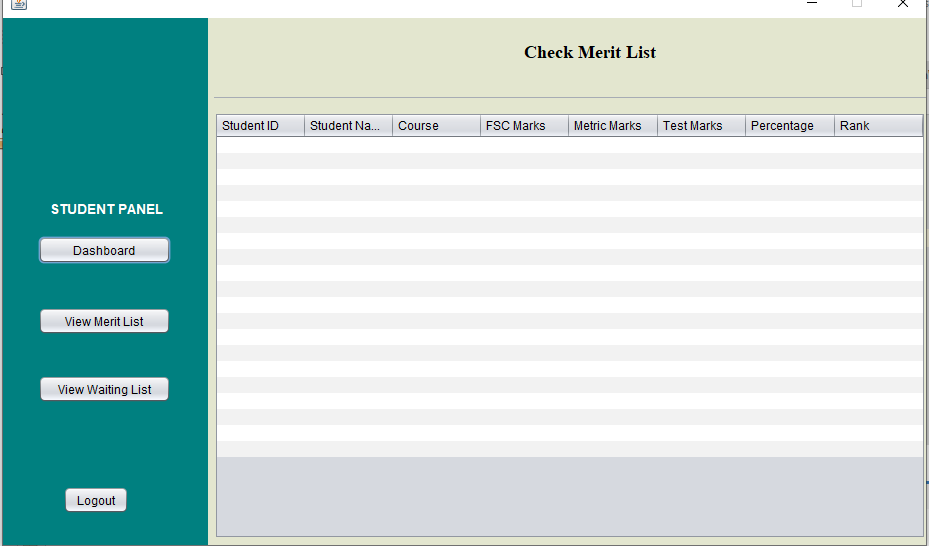
#### Interface:



#### Use Case: Check Merit List

|  |  |
| --- | --- |
| Use case name | Check merit list |
| Scope | This use case describes the actions and interactions involved in checking the merit list of admitted students in the University Admission System. The merit list contains the names of students who have been admitted to the university, sorted by their academic achievements. |
| Primary actors | Guest/student. |
| Pre-condition | The applicant has submitted an application to the university.  The university has published the merit list. |
| Main success scenario | 1. The applicant navigates to the university website and selects "View Merit List". 2. The system displays the merit list of admitted students results. 3. The applicant reviews the merit list to see if their name is listed. 4. If the applicant's name is not listed in merit list. 5. Then applicant navigates to “view wait list”. 6. The system provides the wait list. 7. The applicant reviews the wait list.  * If the applicant's name is not on the merit list, they receive notification that they have not been admitted. |
| Level | User goal |
| Stakeholders | **Faculty:** Professors and instructors who will be responsible for teaching and advising the admitted students.  **University Administration:** The higher-level administration who oversee the university as a whole. |
| Success guarantee | The merit list accurately reflects the admissions decisions and is published in a timely manner.  Applicants receive timely notifications of their admission status. |
| Exceptions | The University Admission System is expected to accurately and securely store admission decisions.  The system is expected to generate a clear and organized merit list.  The system is expected to notify applicants in a timely manner. |
| Special requirement | The University Admission System must be secure and protect student information from unauthorized access or disclosure.  The system must be able to handle a large volume of admission decisions and students. |
| Technology and data variation list | The University Admission System may use different technologies for displaying the merit list on the university website.  The system may need to convert admission decision data into a format that is suitable for the merit list. |
| Frequency of occurrences | The Check Merit List use case occurs once per admission cycle, which may occur multiple times per year. |
| Miscellaneous | • The university admission System should be regularly updated and maintained to ensure optimal performance and usability.  • The system should be able to send notifications to students regarding their academic achievements. |
| Prototyping | A prototype of the University Admission System has been developed and is undergoing testing. |

#### Interface:



### ABDUL AZIZ (FA21-BSE-058):

#### Use Case: Take Test

|  |  |  |
| --- | --- | --- |
| Use Case Section | Comments | |
| Use case name | Take Online Test | |
| Scope | This use case describes the interactions between the University Admission System and the Student who wants to take the admission test. | |
| Main success scenario | Student | System |
|  | 1. The student navigates to the "Take Test" section of the dashboard and selects the test they wish to take. 2. The student confirms that they understand the instructions and begins the test. 3. The Student answers all questions one by one and proceeds to the next one. 4. The student completes the entry test and submits it. | The system displays instructions for the test, including the allotted time and any special requirements (such as a secure testing environment or a proctoring service).  The system presents the entry test questions to the student one at a time, allowing them to answer each question before proceeding to the next.  The system tracks the time remaining and displays a warning when the time limit is approaching.  The system confirms that the test has been submitted and displays the result of the test and percentage. |
| Preconditions | The Student has logged into the system.  The Student has applied for admission to the university.  The admission test has been scheduled by the university.  The Student has registered for the admission test. | |
| Primary actors | Student | |
| Level | Level 1: Present identification and admission details  Level 2: Complete test within specified time  Level 3: Submit completed test  Level 4: Confirm test completion and send it for evaluation | |
| Stakeholders | Student: Wants to take the admission test.  University: Wants to evaluate the Student's knowledge and skills to determine their eligibility for admission.  Testing Center: Involved in facilitating the test-taking process. | |
| Success guarantee | The University Admission System will ensure that the Student can take the admission test, the test is evaluated correctly, and the application status is updated accordingly. | |
| Exceptions | If the Student is unable to answer the given question in specified time, then the system will skip that question and move to the next one.  If the Student is unable to attend their scheduled testing date due to unforeseen circumstances, they must contact the testing center to reschedule.  If the Student is found to be leaving the university website its test will be canceled. | |
| Special requirements | The test must be taken securely and efficiently.  The system must update the application status once the test has been completed and evaluated.  The test materials and format must be appropriate for the program and the Student's field of study. | |
| Technology and data variation list | The test materials may vary based on the program and the Student's field of study.  The test format may vary based on the program and the Student's field of study. | |
| Frequency of occurences | This use case will occur multiple times per year during the admission process. | |
| Miscellaneous | The University Admission System may provide study materials to the Student to prepare for the test.  The test may have a passing score that the Student must achieve to be considered for admission.  The system may generate a report of the test results for the Student and the university. | |
| Prototyping | A prototype of the test-taking process should be developed and tested with Student to ensure that the system meets their needs and is easy to use. | |

#### Use Case: Print Challan

|  |  |  |
| --- | --- | --- |
| Use Case Section | Comments | |
| Use case name | Take Online Test | |
| Scope | This use case describes the interactions between the University Admission System and the Student who wants to take the admission test. | |
| Main success scenario | Student | System |
| 1. The student navigates to the "Print Challan" section of the dashboard. 2. The student clicks the button to generate the challan. 3. The student then verify the challan and requests to download and print the generated challan. 4. The student downloads and prints the challan. | The system presents the student with an option to generate a challan.  The system validates the student's details, admission status and eligibility.  The system generates a challan with unique identification number and display to the student.  The system provides the challan in a printable and pdf format. |
| Preconditions | The Student has log into the UAS System.  The Student has submitted an application for admission.  The admission fee has been determined by the university. | |
| Primary actors | Student. | |
| Level | Level 1: Generate challan.  Level 2: Download and print challan. | |
| Stakeholders | Student: Wants to print a challan to pay the admission fee.  University: Wants the Student to pay his admission fee to proceed further with the admission process. | |
| Success guarantee | The University Admission System will allow the Student to easily generate and print a challan for the admission fee and the system will update the application status accordingly.  The System will validate the fee amount, student’s details, and unique identification number. | |
| Exceptions | If the Student has not downloaded a fee challan, then the system will not be able to update the status of Student and will not proceed further in the admission process.  If the Student encounters an issue with the payment channels or payment confirmation, they can contact the finance department for assistance.  If Student requests or revised challan due to errors and changes in the admission details. | |
| Special requirements | The challan must contain all required information, such as Student name, program, fee amount, and identification number.  The challan must be unique to prevent duplication or fraud. | |
| Technology and data variation list | The challan may be generated in different formats, such as PDF or Doc.  The challan may have different designs or logos for different programs or departments. | |
| Frequency of occurences | This use case will occur multiple times per day during the admission process. | |
| Miscellaneous | The University Admission System may integrate with a payment gateway to enable online payment of the admission fee.  The system may provide the option to track the status of the challan and the payment.  The challan may have an expiry date to ensure that it is used within a specified timeframe. | |
| Prototyping | A prototype of the challan generation feature should be developed and tested with Student or Guests to ensure that the system meets their needs and is easy to use. | |

#### Use Case: Make Test Payment

|  |  |  |
| --- | --- | --- |
| Use Case Section | Comments | |
| Use case name | Make Payment | |
| Scope | This use case describes the interactions between the University Admission System and the Student who wants to pay the admission fee. | |
| Main success scenario | Student | System |
| 1. The student navigates to the "Make Payment" section of the dashboard and clicks on the make payment button. 2. The Student selects a payment method such as bank transfer or by challan number and click on next button. 3. The Student then enters the specified information and clicks on pay button. 4. The Student then verify and confirms the payment. | The system presents the applicant with a list of available payment methods.  The system then prompts the Student to enter the required information according to the payment method such as account number, challan number, branch number.  The system then send a verification and confirmation message to the Student.  The system confirms the payment and present a thank you message on the screen.  The system now update the admission status according to the payment. |
| Preconditions | The Student has logged into the UAS.  The Student has been accepted for admission to the university.  The admission fee has been determined by the university.  The payment deadline has been specified by the university. | |
| Primary actors | Student. | |
| Level | Level 1: Enter payment information and select payment method.  Level 2: Confirm payment | |
| Stakeholders | Student: Wants to pay the admission fee.  University: Wants to ensure that the admission fee is paid by the Student. | |
| Success guarantee | The University Admission System will allow the Student to easily pay the admission fee and the system will update the application status accordingly. | |
| Exceptions | If the Student did not confirm the payment in the notification then the system will restart this process and the Student will be asked to try again.  If the payment transaction fails or is declined by the payment gateway or processor, the Student is prompted to correct their payment information or select a different payment method.  If the payment transaction is successful but there is a delay or error in recording the payment transaction details, the Student or system administrators can contact the finance department for assistance.  If the Student will not make the admission payment in the given deadline then the system will cancel the admission of the student and the admission status will be set to Zero. | |
| Special requirements | The payment must be processed securely and efficiently.  The system must update the application status once the payment has been completed.  The payment deadline must be enforced to ensure that the Student completes the payment within the specified timeframe. | |
| Technology and data variation list | The payment may be processed through different payment gateways.  The admission fee may vary based on the program and the Student status (domestic/international). | |
| Frequency of occurences | This use case will occur multiple times per year during the admission process. | |
| Miscellaneous | The University Admission System may provide a list of payment methods available for the Student to select.  The payment may have an expiry date to ensure that it is completed within a specified timeframe.  The system may generate a receipt for the Student once the payment has been completed. | |
| Prototyping | A prototype of the make payment process should be developed and tested with Student to ensure that the system meets their needs and is easy to use. | |

#### Use Case: Make Admission Payment

|  |  |
| --- | --- |
| Use Case Section | Comments |
| Use case name | Make Admission Payment |
| Scope | This use case describes the interactions between the University Admission System and the Student who wants to pay the admission fee. |
| Main success scenario | 1. The system presents the Student with an option to pay the admission fee. 2. The Student selects the option. 3. The system presents the applicant with a list of available payment methods. 4. The Student selects a payment method such as bank transfer or by challan number. 5. The system then ask the Student to enter the required information according to the payment method such as account number or challan number. 6. The Student then enters the specified information and clicks on pay button. 7. The system then send a verification and confirmation message to the Student. 8. The Student then verify and confirms the payment. 9. The system confirms the payment and updates the application status to indicate that the admission fee has been paid and update the admission status to 100%. |
| Preconditions | The Student has logged into the UAS.  The Student has been accepted for admission to the university.  The admission fee has been determined by the university.  The payment deadline has been specified by the university. |
| Primary actors | Student. |
| Level | Level 1: Enter payment information and select payment method  Level 2: Confirm payment |
| Stakeholders | Student: Wants to pay the admission fee.  University: Wants to ensure that the admission fee is paid by the Student. |
| Success guarantee | The University Admission System will allow the Student to easily pay the admission fee and the system will update the application status accordingly. |
| Exceptions | If the Student did not confirm the payment in the notification then the system will restart this process and the Student will be asked to try again.  If the payment transaction fails or is declined by the payment gateway or processor, the Student is prompted to correct their payment information or select a different payment method.  If the payment transaction is successful but there is a delay or error in recording the payment transaction details, the Student or system administrators can contact the finance department for assistance.  If the Student will not make the admission payment in the given deadline then the system will cancel the admission of the student and the admission status will be set to Zero. |
| Special requirements | The payment must be processed securely and efficiently.  The system must update the application status once the payment has been completed.  The payment deadline must be enforced to ensure that the Student completes the payment within the specified timeframe. |
| Technology and data variation list | The payment may be processed through different payment gateways.  The admission fee may vary based on the program and the Student status (domestic/international). |
| Frequency of occurences | This use case will occur multiple times per year during the admission process. |
| Miscellaneous | The University Admission System may provide a list of payment methods available for the Student to select.  The payment may have an expiry date to ensure that it is completed within a specified timeframe.  The system may generate a receipt for the Student once the payment has been completed. |

#### Interface:







### Daniyal Nawaz (FA21-BSE-083):

***Fully Dressed Use Cases of University Admission System***

|  |  |
| --- | --- |
| USE CASE NAME: | LOGIN |
| Scope: | University admission system |
| Primary Actor(s): | Student, Faculty, Administrator |
| Stakeholders and Interests: | Student: Wants to access their application status, view admission requirements, and submit their application.  Faculty: Wants to access course materials, view student records, and communicate with students.  Administrator: Wants to manage the admission process, access university data, and maintain the system. |
| Preconditions: | The user has a valid username and password. |
| Main Success Scenario: | The user opens the admission system login page and enters their credentials.  The system validates the user's credentials and checks their access level.  The system logs the user into the admission system and displays their appropriate dashboard or landing page. |
| Success Guarantee: | The user is logged into the admission system with the appropriate level of access. |
| Extensions: | If the user enters invalid credentials, the system displays an error message and prompts them to re-enter their credentials or reset their password.  If the user's access level is insufficient, the system displays an error message and denies access to the admission system.  If the user's session times out, the system logs the user out and redirects them to the login page. |
| Frequency of Occurrence: | High |
| Technology and Data Variation List: | The admission system may use different authentication technologies, such as LDAP, OAuth, or custom authentication. The system may store user login data in a database, file system, or other storage technologies. |
| Prototype: | A prototype of the login feature can be created using web application frameworks such as Ruby on Rails, Django, or Laravel. |
| Level: | User goal |
| Use Case Section: | Core |
| Special Requirements: | The system should ensure that user passwords are secure and stored using appropriate encryption techniques. |
| Miscellaneous: | None |

|  |  |
| --- | --- |
| USE CASE NAME: | APPLY FOR ADMISSION |
| Scope: | University admission system |
| Primary Actor(s): | Student |
| Stakeholders and Interests: | Student: Wants to apply for admission to the university, view admission requirements, and submit their application. |
| Preconditions: | The user has a valid login and has accessed the admission system. |
| Main Success Scenario: | The user selects the option to apply for admission.  The system presents the admission application form to the user.  The user fills out the application form, including personal information, academic records, and supporting documents.  The user submits the application form.  The system validates the application form and checks for any missing information or errors.  The system confirms the successful submission of the application and provides the user with a confirmation message. |
| Success Guarantee: | The user has successfully submitted their admission application. |
| Extensions: | If the user fails to fill out the application form correctly, the system displays an error message and prompts them to correct the errors.  If the user fails to provide all necessary supporting documents, the system displays a warning message and prompts them to upload the missing documents.  If the user encounters technical issues while filling out the application form, the system provides a support contact for assistance. |
| Frequency of Occurrence: | Moderate |
| Technology and Data Variation List: | The admission system may use different application form formats, such as online forms or downloadable PDF forms.  The system may store application data in a database, file system, or other storage technologies. |
| Prototype: | A prototype of the apply for admission feature can be created using web application frameworks such as Ruby on Rails, Django, or Laravel. |
| Level: | User goal |
| Use Case Section: | Core |
| Special Requirements: | The system should ensure that the admission application form is secure and that user data is stored using appropriate encryption techniques. |
| Miscellaneous: | None |

|  |  |
| --- | --- |
| Use Case Name: | APPLY FOR TEST |
| Scope: | University admission system |
| Primary Actor(s): | Student |
| Stakeholders and Interests: | Student: Wants to apply for a university admission test, view test requirements, and receive confirmation of their test registration. |
| Preconditions: | The user has a valid login and has accessed the admission system. |
| Main Success Scenario: | The user selects the option to apply for a test.  The system presents the available tests to the user.  The user selects the desired test and confirms their choice.  The system validates the user's eligibility for the test, based on their admission status and other relevant criteria.  If the user is eligible, the system presents the test registration form to the user.  The user fills out the test registration form, including personal information and payment details.  The user submits the test registration form.  The system validates the test registration form and checks for any missing information or errors.  The system confirms the successful registration of the test and provides the user with a confirmation message. |
| Success Guarantee: | The user has successfully registered for the university admission test. |
| Extensions: | If the user is not eligible for the test, the system displays an error message and prompts them to check their admission status or contact support.  If the user fails to fill out the test registration form correctly, the system displays an error message and prompts them to correct the errors.  If the user fails to provide valid payment details, the system displays a warning message and prompts them to update their payment information.  If the user encounters technical issues while registering for the test, the system provides a support contact for assistance. |
| Frequency of Occurrence: | Moderate |
| Technology and Data Variation List: | The admission system may use different payment gateways or payment processing systems to handle test registration fees.  The system may store test registration data in a database, file system, or other storage technologies. |
| Prototype: | A prototype of the apply for test feature can be created using web application frameworks such as Ruby on Rails, Django, or Laravel. |
| Level: | User goal |
| Use Case Section: | Core |
| Special Requirements: | The system should ensure that test registration data is secure and that payment information is stored using appropriate encryption techniques. |
| Miscellaneous: | None |
|  |  |



### SOMAN AHMAD (FA21-BSE-150)

|  |  |
| --- | --- |
| **Use Case Name** | **LOG OUT** |
| **Scope** | This use case describes the process of logging out of the university admission system, which is used by students and staff members to manage the admission process. |
| **Primary Actors** | Student |
| **Stakeholders** | Students: They want to be able to securely log out of the system to protect their personal information.  Staff Members: They want to ensure that the system is secure and that students can log out without any issues. |
| **Precondition** | The user is logged in to the university admission system. |
| **Main Success Scenario** | 1. The user clicks on the "Log Out" button in the system.  2. The system prompts the user to confirm that they want to log out.  3. The user confirms that they want to log out.  4. The system logs the user out and redirects them to the login page. |
| **Alternative Flow** | If the user cancels the log out confirmation, the system remains on the current page. |
| **Success Guarantee** | The user is successfully logged out of the system and their session is terminated. |
| **Expectations** | The user expects that their personal information will be protected after logging out of the system. |
| **Technology and Data Variation List** | 1. The university admission system is accessed via a web browser  2. The user's personal information is stored securely in a database.  3. The system is built using object-oriented programming principles. |
| **Frequency of Occurrence** | This use case occurs whenever a user wants to log out of the university admission system. |
| **Use Case Section** | This use case is part of the "Account Management" section of the university admission system. |
| **Special Requirements** | The user's session must be terminated securely to protect their personal information. |
| **Prototype** | A prototype of the university admission system is available for testing. |
| **Miscellaneous** | None. |

|  |  |
| --- | --- |
| **Use Case Name** | **ENTER ACADEMIC INFO** |
| **Scope** | This use case describes the process of managing academic information in the university admission system, which is used by students and staff members to manage the admission process. |
| **Primary Actors** | Student: A person who is applying for admission to the university<br>- Staff Member: A person who is responsible for managing the admission process |
| **Stakeholders** | Students: They want to be able to provide accurate academic information to be considered for admission.  Staff Members: They want to ensure that the academic information is accurate and up-to-date for the admission process. |
| **Precondition** | The user is logged in to the university admission system. |
| **Main Success Scenario** | 1. The user selects the "Academic Info" option in the system.  2. The system displays a form for the user to enter their academic information.  3. The user fills out the form with their academic information.  4. The system saves the academic information to the user's profile. |
| **Alternative Flow** | If the user cancels the form, the system returns the user to the previous page. |
| **Success Guarantee** | The user's academic information is saved to their profile in the system. |
| **Expectations** | The user expects that their academic information will be accurately recorded in the system for consideration in the admission process. |
| **Technology and Data Variation List** | 1. The university admission system is accessed via a web browser.  2. The user's academic information is stored securely in a database.  3. The system is built using object-oriented programming principles. |
| **Frequency of Occurrence** | This use case occurs whenever a user needs to enter or update their academic information in the university admission system. |
| **Use Case Section** | This use case is part of the "Application Management" section of the university admission system. |
| **Special Requirements** | The academic information entered by the user must be validated to ensure that it is accurate and complete. |
| **Prototype** | A prototype of the university admission system is available for testing. |
| **Miscellaneous** | None. |

|  |  |
| --- | --- |
| **Use Case Name** | **VERIFY DOCUMENT** |
| **Scope** | This use case describes the process of verifying documents in the university admission system, which is used by staff members to manage the admission process. |
| **Primary Actors** | Staff Member: A person who is responsible for verifying documents submitted by students for the admission process. |
| **Stakeholders** | Students: They want to ensure that their documents are accurately verified and processed for the admission process.  Staff Members: They want to ensure that the documents submitted by students are accurate and complete before being processed for the admission process. |
| **Precondition** | The staff member is logged in to the university admission system and has access to the submitted documents. |
| **Main Success Scenario** | 1. The staff member selects the "Verify Documents" option in the system.  2. The system displays a list of submitted documents for the staff member to verify.  3. The staff member reviews each document and verifies that it is accurate and complete.  4. The staff member updates the status of each document as verified or not verified in the system.  5. The system saves the updated status of the documents. |
| **Alternative Flow** | If the staff member identifies an error in a document, they can mark it as "rejected" and provide a reason for rejection. The student will be notified and given a chance to submit a corrected document. |
| **Success Guarantee** | The documents submitted by students are accurately verified and processed for the admission process. |
| **Expectations** | The staff member expects that the documents submitted by students are accurate and complete before being processed for the admission process. |
| **Technology and Data Variation List** | 1. The university admission system is accessed via a web browser.  2. The documents submitted by students are stored securely in a database.  3. The system is built using object-oriented programming principles. |
| **Frequency of Occurrence** | This use case occurs whenever a staff member needs to verify documents submitted by students for the admission process. |
| **Use Case Section** | This use case is part of the "Application Management" section of the university admission system. |
| **Special Requirements** | The staff member must have access to the submitted documents and must be able to verify their accuracy and completeness. |
| **Prototype** | A prototype of the university admission system is available for testing. |
| **Miscellaneous** | None. |

**INTERFACE:**



### AJWAH SARDAR (FA21-BSE-072)

**Fully Dressed Use Cases:**

|  |  |
| --- | --- |
| USE CASE NAME | COMMENTS |
| Scope | This use case describes the process of a prospective student signing up for an admission system of a university. |
| Primary Actor | Student |
| Stakeholders | 1. Admission staff 2. IT staff 3. University faculty |
| Preconditions | 1. The admission system is accessible. 2. The prospective student has an internet connection. 3. The prospective student is not already registered with the admission system. |
| Main Success Scenario | 1. The admin/student navigates to the admission system's homepage. 2. The admission system displays the login screen. 3. The admin/student put info in email and password fields. 4. Then the admin/student navigates to the login button. 5. If credentials are correct, then shows the admin/student dashboard. 6. If credentials are incorrect then shows an error of invalid credentials. |
| Alternate Scenarios | 1. If the prospective student enters invalid information in step 5, the system displays an error message and prompts them to correct the invalid fields. 2. If the prospective student's email address is already registered, the system displays an error message and prompts them to log in or reset their password. 3. If the confirmation email is not received, the prospective student can click the "Resend Confirmation Email" button on the admission system to receive a new email. |
| Success Guarantee | The prospective student has an active account on the admission system, and can log in to access additional features and information. |
| Frequency of Occurrence | .This use case occurs frequently, as prospective students sign up for the admission system throughout the year. |
| Technology and Data Variation List | 1. The admission system website may be accessed from various devices and web browsers. 2. The personal information required for sign-up may vary depending on the university's requirements |
| Prototype | A prototype of the admission system website may be used to test the sign-up process. |
| Miscellaneous | 1. The university may have additional policies and requirements for signing up for the admission system. 2. The university may provide support or resources for prospective students who have difficulty signing up |
| Use Case Section | Sign up for Admission System |
| Special Requirements | 1. The admission system must be accessible to prospective students. 2. The confirmation email must be sent promptly and reliably to the email address provided by the prospective student |

**Prototype**



**Fully Dressed Use Case**

|  |  |
| --- | --- |
| Use Case Name | Academic Information Intake |
| Scope | University Admission System |
| Primary Actor | Student |
| Stakeholders and Interests | 1. Student: Wants to provide their academic information accurately and conveniently. 2. Admissions Office: Wants to manage the intake of academic information efficiently. |
| Preconditions | The applicant has an account in the university admission system. |
| Main Success Scenario | 1. The applicant logs in to the university admission system. 2. The applicant selects the "Academic Information Intake" option. 3. The system presents a form for the applicant to enter their academic information, including their GPA, standardized test scores, and transcripts. 4. The applicant fills in the form accurately and submits it. 5. The system validates the information provided by the applicant. 6. If the information is valid, the system saves it to the applicant's record. 7. The system sends a confirmation email to the applicant. |
| Alternate Scenarios | 1. If the applicant provides invalid or incomplete information, the system prompts the applicant to correct the errors or fill in the missing information. 2. If the applicant encounters technical difficulties while filling in the form, the system provides support and assistance. |
| Success Guarantee | The system will accurately collect and validate the academic information provided by the applicant and save it to their record. The applicant will receive a confirmation email. |
| Frequency of Occurrence | This use case occurs for each applicant during the admissions process. |
| Technology and Data Variation List | 1. The university admission system may be accessed through a web portal or a mobile application. 2. The system may use different databases for storing applicant records and academic information. |
| Prototype | A prototype of the academic information intake feature can be tested by a group of applicants during the admissions process. |
| Miscellaneous | 1. The system should provide clear instructions and guidelines for filling in the form. 2. The system should ensure the security of applicant information. |
| Use Case Section | This use case is part of the admissions section of the university admission system. |
| Special Requirements | 1. The system must comply with data privacy laws and regulations. 2. The system must be able to handle a large volume of submissions during peak periods. |

**Prototype**

****

**Fully Dressed Use Case**

|  |  |
| --- | --- |
| Use Case Name | Eligibility Criteria |
| Scope | University Admission System |
| Primary Actor | Admissions Office |
| Stakeholders and Interests | Admissions Office: Wants to define the eligibility criteria for applicants to be considered for admission |
| Preconditions | The admissions office has access to the university admission system. |
| Main Success Scenario | 1. The admissions office logs in to the university admission system. 2. The admissions office selects the "Eligibility Criteria" option. 3. The system presents a form for the admissions office to define the eligibility criteria. 4. The admissions office fills in the form accurately, including minimum GPA requirements, standardized test score requirements, and any other relevant criteria. 5. The system validates the information provided by the admissions office. 6. If the information is valid, the system saves it to the eligibility criteria section of the admission system. 7. The system sends a confirmation email to the admissions office. |
| Alternate Scenarios | If the admissions office provides invalid or incomplete information, the system prompts the admissions office to correct the errors or fill in the missing information. If the admissions office encounters technical difficulties while filling in the form, the system provides support and assistance. |
| Success Guarantee | The system will accurately collect and validate the eligibility criteria defined by the admissions office and save it to the eligibility criteria section of the admission system. The admissions office will receive a confirmation email. |
| Frequency of Occurrence | This use case occurs periodically, usually before the start of the admissions cycle or whenever there is a need to update the eligibility criteria. |
| Technology and Data Variation List | 1. The university admission system may be accessed through a web portal or a mobile application. 2. The system may use different databases for storing eligibility criteria and applicant records. |
| Prototype | A prototype of the eligibility criteria feature can be tested by the admissions office before the start of the admissions cycle. |
| Miscellaneous | 1. The system should provide clear instructions and guidelines for defining the eligibility criteria. 2. The system should ensure the security of eligibility criteria information. |
| Use Case Section | This use case is part of the admissions section of the university admission system. |
| Special Requirements | 1. The system must comply with university policies and regulations regarding eligibility criteria. 2. The system must be able to handle multiple eligibility criteria for different programs and departments. |

**Prototype**

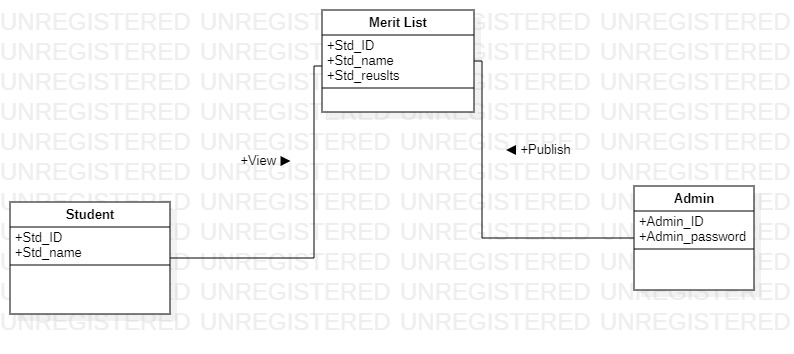


# **CHAPTER 3:**

## Domain Model:

### HAMZA BADAR (FA21-BSE-055)

#### Use Case: Publish and Create Merit List



### SYED SHAH HUSSAIN (FA21-BSE-172);

#### Use Case: Notify Students

# **CHAPTER 4:**

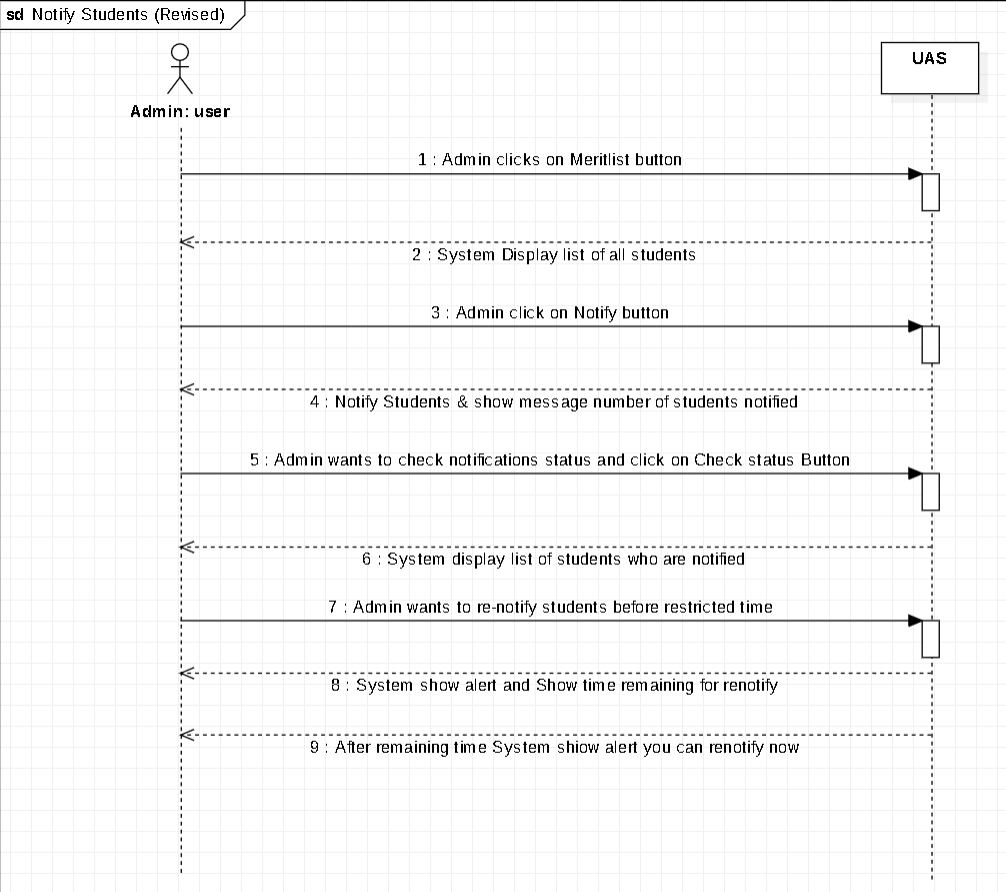
## SSD:

### SYED SHAH HUSSAIN (FA21-BSE-172)

#### Use Case: Notify Students



#### Use Case: Notify Students (REVERSE ENGINEERED)



#### Use Case: Confirm Admission SYED SHAH HUSSAIN (FA21-BSE-172)



REVERSE ENDINEERED (Modified);



### Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057)



### Use Case: Scheduling Test MAHAD WAJID (FA21-BSE-057)



### Use Case: Announcement MAHAD WAJID (FA21-BSE-057)



### Use Case: Take Test ABDUL AZIZ (FA21-BSE-058)

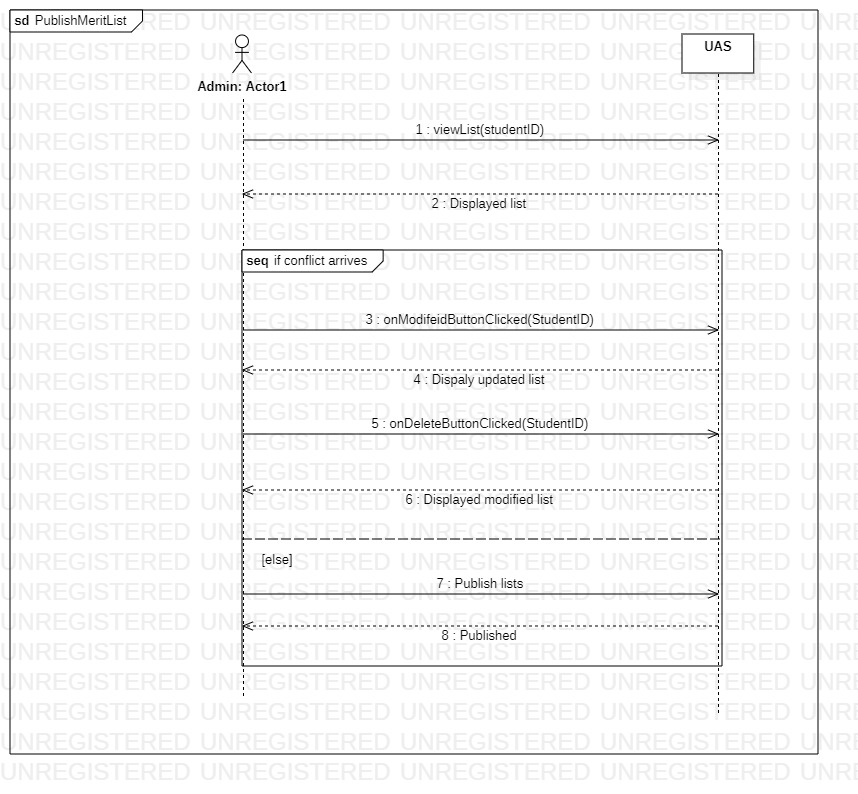
### Use Case: Print Challan ABDUL AZIZ (FA21-BSE-058)



### Use Case: Make Payment ABDUL AZIZ (FA21-BSE-058)



### Use Case: Publish merit list by (Hamza Badar FA21-BSE-055)



### Use Case: Check merit list by (Hamza Badar FA21-BSE-055)

A screen shot of a computer

Description automatically generated with low confidence

# **CHAPTER 5:**

# **Operation Contracts:**

### Use Case: Notify Students SYED SHAH HUSSAIN (FA21-BSE-172)

|  |  |
| --- | --- |
| Operations | notifyStudents(  admissionID: int,  StudentName: String,  notificationType: String) |
| Cross Reference | Use Case : Notify Students |
| Pre conditions | * The admission ID provided exists in the university admission system. * There are students associated with the provided admission ID. |
| Post conditions | * The notifyStudent instance is created according to the notification type. * The notification is sent to the relevant students. * The notification is associated with the students. * The notification & notified students are/is recorded in the university admission system. * System also show number of Students Modified * System also display the Notifeid Students List. |

### Use Case: Confirm Admission SYED SHAH HUSSAIN (FA21-BSE-172)

|  |  |
| --- | --- |
| Operation | admissionConfirmation ( admissionID:int, std\_name:String, feeStatus:boolean ) |
| Cross Reference | Use Case: Confirm Admission |
| Pre conditions | * The student has submitted an application for admission. * The admission ID provided exists in the university admission system. * The admission status for the provided admission ID is "Pending". |
| Post Conditions | * The admission status for the provided admission ID is updated to "Confirmed". * • The student's enrollment status is set to "Enrolled" in the university admission system. * • The student's personal and academic information is saved in the university database. * • The student is officially registered as a student of the university for the specified program. |

### Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057)

|  |  |
| --- | --- |
| Operation | enterPersonalInfo(firstName: string, lastName: string, email: string, phone: string, address: string) |
| Cross Reference | Use Case: Enter Personal Info |
| Pre conditions | The student is in the process of submitting their personal information for university admission. |
| Post Conditions | * A new instance of the Student class is created. * The newly created Student instance is associated with the university admission process. * The personal information of the student is updated with the provided details. * The student instance is associated with the admission application. * The personal information is available for further processing and review by the admissions committee. |

### Use Case: Scheduling Test MAHAD WAJID (FA21-BSE-057)

|  |  |
| --- | --- |
| Operations | schedule Test(test Name: string, date: Date, time: Time, location: string) |
| Cross reference | Use case :Scheduling Test |
| Pre-conditions | * The student has the necessary authorization to schedule tests. * The university admission system is in the test scheduling mode. * The specified test exists in the system. |
| Post conditions | * A test session for the specified test with the given date, time, and location is scheduled in the university admission system. * The test session is associated with the appropriate test and identified by a unique identifier. * The scheduled test session becomes available for registration by eligible applicants. * The test session details, including the test name, date, time, and location, are saved in the system. |

### Use Case: Announcement MAHAD WAJID (FA21-BSE-057)

|  |  |
| --- | --- |
| Operations | Announcement(title: string, content: string, author: string, date: Date) |
| Cross Reference | Use Case : Announcement |
| Pre conditions | * The admin has appropriate authorization to create an announcement. * The system is in the announcement management mode. |
| Post conditions | * An announcement with the provided title, content, author, and date is created and saved in the university admission system. * The announcement is available for viewing by all users. * The announcement is associated with the appropriate category or target audience, if applicable. * The announcement is timestamped with the creation date. |

### Use Case: Take Test ABDUL AZIZ (FA21-BSE-058)

|  |  |
| --- | --- |
| Operations | takeTest () |
| Cross reference | Use Case: Take Test. |
| Pre-conditions | * The university admission process is underway. * The student has an entry test to take admission. |
| Post-conditions | * The takeTest instance is created for the student. * The takeTest instance is associated with the student. * The system load the test from DB. * The studentProgress became progress. |

### Use Case: Print Challan ABDUL AZIZ (FA21-BSE-058)

|  |  |
| --- | --- |
| Operations | printChallan (studentId: int, challanNo: String) |
| Cross reference | Use Case: Print Challan. |
| Pre-conditions | * The university admission process is underway. * The student has a fee to be submitted. |
| Post-conditions | * The printChallan instance is created for the student. * The printChallan instance is associated with the student instance. * The new challan no is assigned to the challan. * The Challan is presented to the student. |

### Use Case: Make Payment ABDUL AZIZ (FA21-BSE-058)

|  |  |
| --- | --- |
| Operations | makePayment (studentId: int, accountNo: String, branchNo: string, paymentType: String, amount: double, challanNo: String, challanImage: File) |
| Cross reference | Use Case: Make Payment. |
| Pre-conditions | * The university admission process is underway. * The student has a remaining payment to pay and proceed in admission process. * The payment has to be made according to the student admission status. |
| Post-conditions | * The system initiates the payment process with specified payment type. * The payment instance is created according to the payment type for the student. * The payment transaction is associated with the student. * The system verifies the payment transaction for studentID and bank verification. |

### Use Case: Publish merit list by (Hamza Badar FA21-BSE-055)

|  |  |
| --- | --- |
| Operation | viewList () |
| Cross reference | Publish merit list |
| Pre-condition | * The result list is in the system. * The merit list must not be empty. * The merit list data must contain valid student objects with marks. * All high percentage students are at topmost. |
| Post condition | * + The merit list was published.   + The published merit list is accurate, reflecting the correct rankings. |

### Use Case: Check merit list by (Hamza Badar FA21-BSE-055)

|  |  |
| --- | --- |
| Operation | getMeritList () |
| Cross reference | Check merit list |
| Pre-condition | * The merit list has been published. * Applicants have the access to check the list. |
| Post condition | * + The merit list is displayed. |

### Use Case: LOGIN by Ajwa Sardar(FA21-BSE-073)

|  |  |
| --- | --- |
| **Operation** | login () |
| **Cross Reference** | Use Case - Login |
| **Preconditions** | * The user data must be in the backend. * The user has valid email address. |
| **Post Conditions** | * The user is logged in to the university admission system. * The user can perform actions according to their view. |

### Use Case: ENTER ACADEMIC INFO BY SOMAN AHMAD(FA21-BSE-150)

|  |  |
| --- | --- |
| **Operations** | EnterAcademicInfo (institutionname: String, totalmarks: int, obtainmarks: int, percentage: double, session: int) |
| **Cross Reference** | Use Case - Enter Academic Info |
| **Pre conditions** | * The student is currently logged into the university admission system. * The student has already provided their personal information. * The required fields for academic information are available. |
| **Post conditions** | * The academic information of the student is successfully entered into the university admission system. * The student's academic details are associated with their admission application. |

### Use Case: VERIFY DOCUMENT BY SOMAN AHMAD(FA21-BSE-150)

|  |  |
| --- | --- |
| **Operations** | verifyDocument(document: File) |
| **Cross reference** | Use case - Verify Document |
| **Pre-conditions** | * The document to be verified is provided by the student. * The document is in a supported file format. |
| **Post conditions** | * The document is successfully verified for authenticity and validity. |

# **CHAPTER 6:**

# **Package Diagram:**



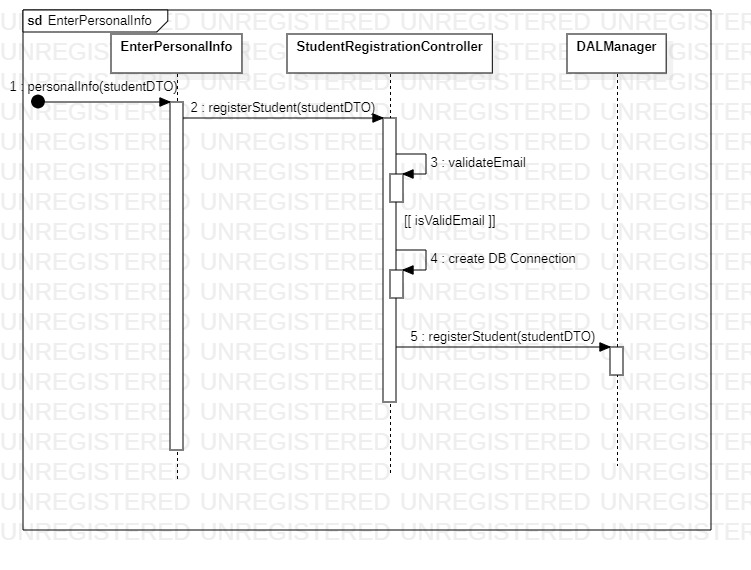
# **CHAPTER 7:**

# **Class Diagram:**

# **CHAPTER 8:**

## Sequence Diagram:

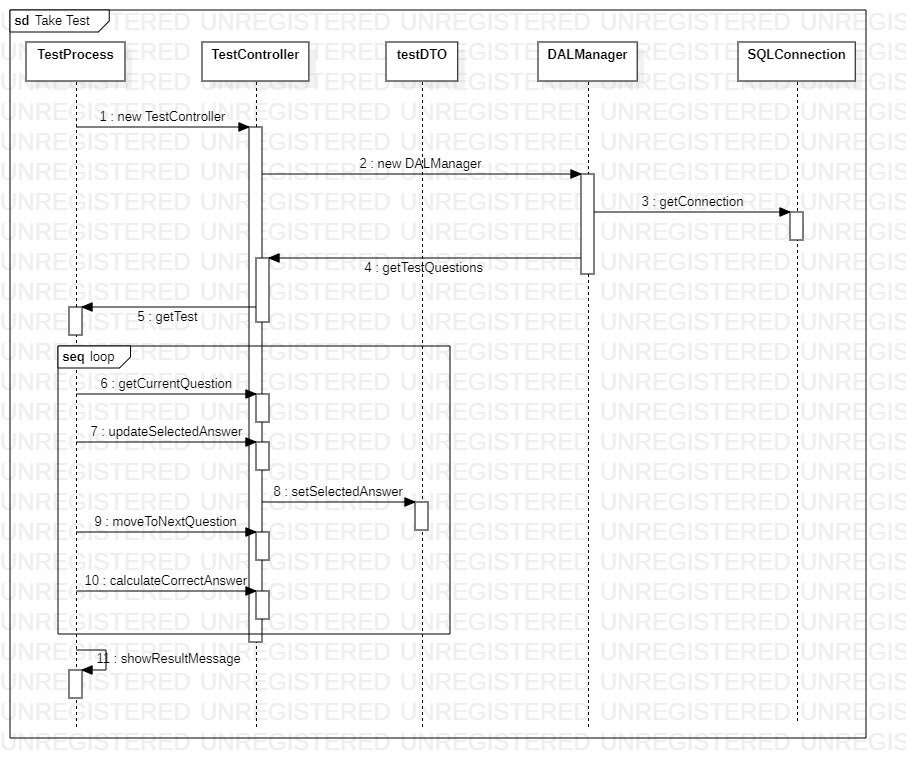
### Use Case: Enter Personal Info MAHAD WAJID (FA21-BSE-057)



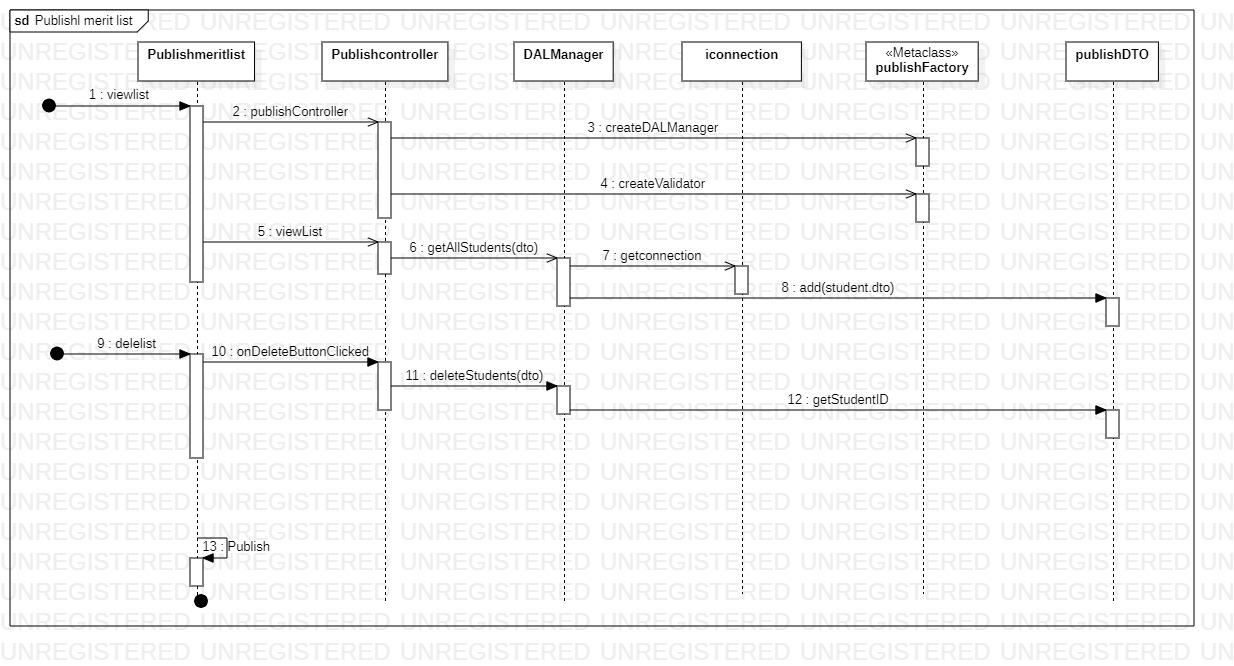
### Use Case: Announcement BY MAHAD WAJID (FA21-BSE-057)



### Use Case: Take Test ABDUL AZIZ (FA21-BSE-058)



### Use Case: Publish Merit list (FA21-BSE-055)



### Use Case: Check Merit list (FA21-BSE-055)

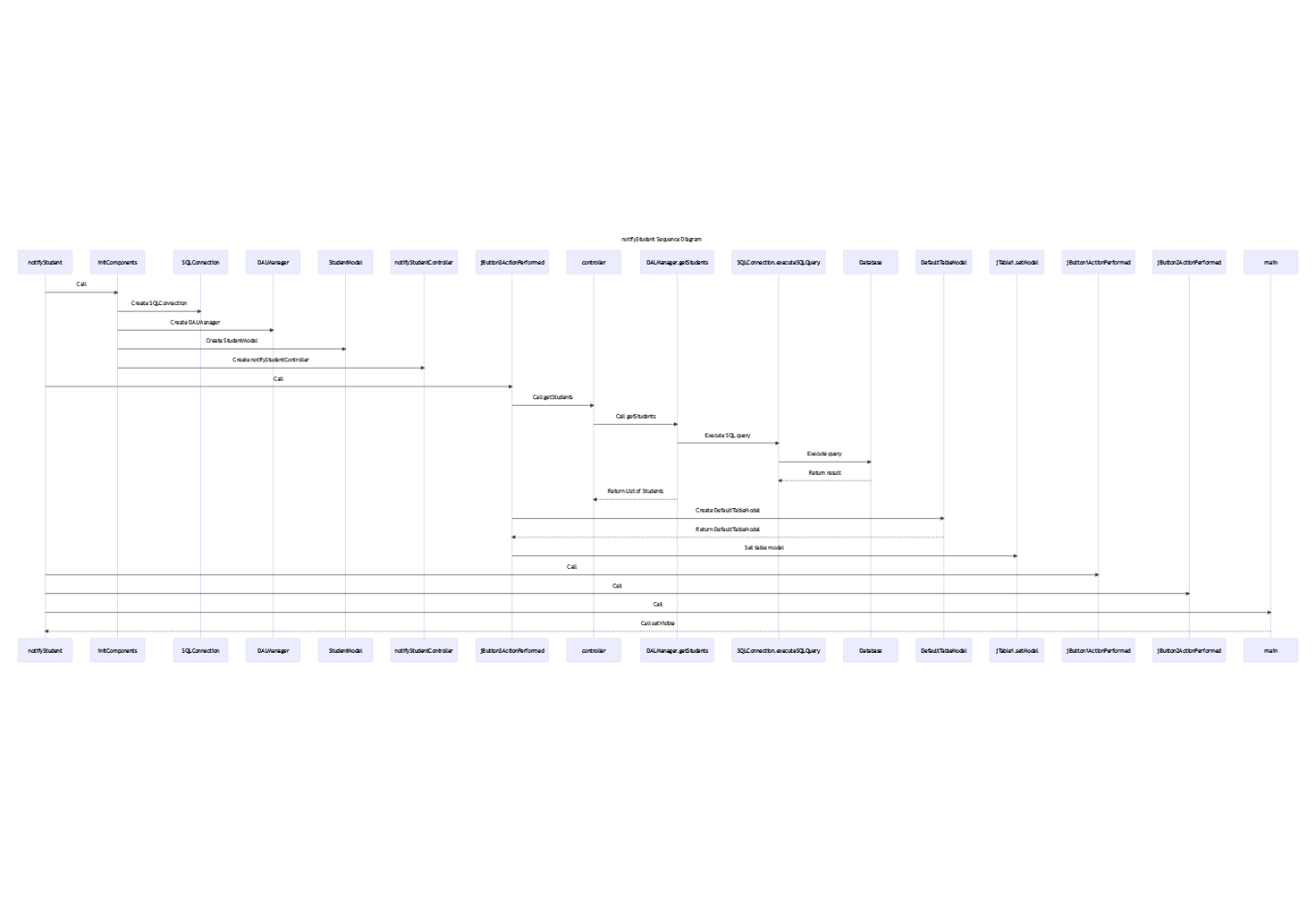
A diagram of a diagram

Description automatically generated

### Use Case: admission confirmation SYED SHAH HUSSAIN (FA21-BSE-172)



### Use Case: notify Students SYED SHAH HUSSAIN (FA21-BSE-172)



## Communication Diagram:

### SYED SHAH HUSSAIN (FA21-BSE\_172):

#### Use Case: Notify Students

