**UMT Admission System Documentation**



**UMT Admission System**

Requirements Specification

Software Engineering

Faculty of Computer Science and IT

Department of Software Engineering

**Requirements Specification prepared by:**

Joana Sulçe

Trejzi Xhaho

Metjon Metuli

Martin Shëngjergji

Eduart Bajo

**Received by**: Igli Hakrama

**Table of Contents**

1. **EXECUTIVE SUMMARY…………………………………….…...........................3**
   1. PROJECT OVERVIEW………………………………………………………………….....3
   2. PURPOSE AND SCOPE OF THIS SPECIFICATION…………………………………….3
2. **PRODUCT/SERVICE DESCRIPTION…………………………………………..3**
   1. PRODUCT CONTEXT…………………………………………………………..…………3
   2. USER CHARACTERISTICS……………………………………………….........................5
   3. ASSUMPTIONS…………………………………………………………………………….6
   4. CONSTRAINTS…………………………………………………………………………….7
   5. DEPENDENCIES…………………………………………………………………...............7
3. **REQUIREMENTS………...………………………………………..........................9**
   1. FUNCTIONAL REQUIREMENTS………………………………………….......................9
   2. NON-FUCTIONAL REQUIREMENTS………………………………………..………....12
      1. User Interface Requirements………………………………………………………..….12
      2. Usability……………………………………………………………………..................17
      3. Performance……………………………………………………………………………17
4. **ANALYSIS MODEL……………………………………………………………...18**
   1. USER SCENARIOS……………………………………………………………………….18
   2. USER SCENARIOS EXTENDED……………………………………………………...…20
   3. USE CASES…………………………………………………………………………….…27
   4. Behavioral Diagrams………………………………………………………………………31
      1. Use Case Diagrams…………………………………………………………………...31
      2. Activity Diagrams…………………………………………………………………….36
      3. State Diagrams……………………………………………………………………..…42
   5. Interaction Diagrams………………………………………………………………………47
      1. Sequence Diagrams…………………………………………………………………...47
      2. Collaboration Diagrams………………………………………………………...…….48
   6. Data Flow Diagrams……………………………………………………………………….49
   7. Structural Diagrams……………………………………………………………………….50.
      1. ER Diagram…………………………………………………………………………..50
5. **PROJECT PLANNING…………………………………………………………..51**

**APPENDIX………………………………………………………………………..52**

APPENDIX A SKETCHES…………………………………………………………….…52

**UMT Admission System Documentation**

1.Executive Summary

**1.1 Project Overview**

The University Admission System is a comprehensive solution designed to facilitate the registration process for new students at the university. The system caters to multiple user roles, including students, the Student Relations Office, the Finance Office, the Secretary Office, and professors/staff.

Students can easily apply to the university by accessing the online application form, which does not require them to create a login. The application form captures essential information necessary for admission consideration.

The Student Relations Office plays a crucial role in managing the admission process. Upon logging into the system, they are presented with a user-friendly interface that displays a table of pending applications. This table serves as a centralized repository of all students who have submitted their applications through the university's website. The office has full access to the list and can open individual applications to review the details provided by the students.

To enhance collaboration and communication, the Student Relations Office has the ability to add comments and provide feedback within each application. This feature ensures efficient coordination between the office and prospective students during the evaluation process. Once the office confirms the application with the student, they can update the student's status to the applicant list, indicating that the application is being processed.

The Finance Office is responsible for managing the financial aspects of the admission process. After logging into the system, they can view the complete student list, including pending applications. However, they are only authorized to make changes to the applicant list. The office maintains a separate column for transactions, where they can record students who have made the first payment to the university. This column serves as a record of financial transactions and helps track the status of student payments.

Once the transaction is confirmed by the Finance Office, the Student Relations Office is notified and updates the student's status to the approved list. This step indicates that the student has successfully completed the financial requirements and is ready to proceed with registration.

The Secretary Office, upon logging into the system, can view all the student lists. However, they are only permitted to make changes to the approved list. Once the Secretary Office verifies that the necessary registration procedures have been completed, they update the student's status to "registered." This status signifies that the student is officially enrolled at the university.

Professors and other staff members have restricted access to the system. They can log in to view student data and lists for statistical purposes. However, they are not authorized to make any changes or edits to the data.

The University Admission System streamlines the registration process by providing a centralized platform for managing student applications, tracking financial transactions, and facilitating communication between different offices involved in the admission process. The system enhances efficiency, improves collaboration, and ensures a smooth transition from application to registration for new students at the university.

**1.2 Purpose and Scope of this Specification**

The purpose of this specification is to provide a comprehensive understanding of the University Admission System and outline its requirements and functionalities.

The specification covers the student application process, allowing students to easily apply online without requiring a login. It outlines the responsibilities of the Student Relations Office, including the review of applications, provision of feedback, and updating application statuses. The Finance Office's role is also defined, encompassing tasks related to financial aspects such as verifying student payments and maintaining transaction records.

Additionally, the specification includes the responsibilities of the Secretary Office in managing the final registration process for approved students. It outlines the access permissions and limitations for professors and staff, who have read-only access to student data for statistical analysis. Communication and collaboration features, such as enabling comments, feedback, and status updates within the system, are also addressed.

The specification further highlights the system interfaces, including the user interfaces and interactions required for each user role. It also considers any constraints or limitations that may impact the system's development and functionality.

2.Product/Service Description

**2.1 Product Context**

The University Admission System operates within the university's administrative framework, serving as a centralized web-based platform for managing the registration of new students. It integrates with existing systems, databases, and workflows, facilitating seamless data exchange. The system enhances collaboration and communication among students, the Student Relations Office, the Finance Office, the Secretary Office, and professors/staff. It provides role-specific interfaces and functionalities, enabling efficient application processing, financial management, registration, and statistical analysis. The system plays a vital role within the university's administrative infrastructure, streamlining the admission process and fostering effective communication among stakeholders.

**2.2 User Characteristics**

There are five type of users that will use the system:

* Students
* Student Relations Office (SRO)
* Finance Office (FO)
* Secretary Office (SO)
* Proffesors/Staff

**Students** are able of the following functionalities:

* Fill out the application form
* Submit the application
* Apply online without requiring a login
* Request support or assistance
* View comments and feedback
* Respond to comments and provide additional information
* Pay application fees

**SRO** are able of the following functionalities:

* Log into the system
* View pending applications
* Open and review individual applications
* Add comments and provide feedback
* Update application status
* Communicate with students
* Receive notifications
* Generate reports
* Access student data
* Collaborate with other offices

**FO** are able of the following functionalities:

* Log into the system
* View complete student list
* Access pending applications
* Update applicant list
* Record financial transactions
* Track student payments
* Confirm transactions
* Collaborate with other offices
* Generate financial reports
* View student data

**SO** are able of the following functionalities:

* Log into the system
* View all student lists
* Access approved list
* Update student status to "registered"
* Collaborate with other offices
* Generate reports
* Access student data
* Communicate registration status
* Handle registration-related inquiries
* Coordinate registration procedures

**Professors/ Staff** are able of the following functionalities:

* Log into the system
* View student data
* Access student lists
* Generate reports
* View admission statistics
* Access for research purposes
* View for academic planning
* Stay informed

**2.3 Assumptions**

* It is assumed that the log in for the staff will be from UMT University email.
* It is assumed that the interface and database is part of the UMT University.
* It is assumed that all applicants are eligible to apply for the university and the admission process is based on their application and payment fees.
* It is assumed that the system will be used by authorized personnel only and will be secured with appropriate security measures such as user authentication and authorization.
* It is assumed that the system will be able to handle a large volume of data related to student applications, student registrations and will be scalable as per the university’s needs.
  1. **Constraints**
* Users are expected to run the software on the well-known platforms Windows or Mac. Because of the complexity of the solution the device should have at least average computation power.
* The users should make sure to have reliable internet connection in order for the software to work.
* The intended audience (students, proffesors&staff, student relations office, finance office, secretary office) are expected to be at least semi proficient in the English language to understand the utilities provided.
* The audience should know how to use the application.
* The back-end structure should be ready to respond to all requests at any time which might be highly concurrent peaking at university operating hours.
  1. **Dependencies**

There are some dependencies between **Students and the Student Relations Office**:

* Application Submission:Students depend on the SRO to receive and process their applications. The SRO acts as the central point of contact for students during the admission process.
* Application Review: The SRO reviews and evaluates the applications submitted by students. This evaluation determines whether the student meets the admission criteria and requirements set by the university.
* Feedback and Communication: The SRO provides feedback to students regarding their applications. This feedback may include requests for additional information, clarification, or documentation required for the evaluation process. Students rely on the SRO to receive timely and clear feedback to ensure their application progresses smoothly.
* Collaboration and Coordination: Students may need to collaborate and coordinate with the SRO during the admission process. This collaboration includes responding to queries, providing additional information or documentation, and addressing any concerns or issues raised by the SRO.
* Confirmation of Application: Once the SRO confirms the application with the student, indicating that it is being processed further, students rely on this confirmation to understand the next steps and requirements for registration.

There are some dependencies between **Students and Financial Office:**

* Financial Requirements: Students depend on the Finance Office to provide information on the financial requirements and payment procedures necessary for admission to the university. This includes tuition fees, payment deadlines, and any additional financial documentation that needs to be submitted.
* Payment Submission: Students rely on the Finance Office to process their payments and record them accurately in the system. They need to make the required payments within the specified deadlines to fulfill the financial obligations for admission.
* Financial Record-keeping: Students depend on the Finance Office to maintain accurate financial records related to their admission. These records may include payment history, outstanding balances, or any financial documentation provided by the students.

There are some dependencies between **SRO and FO:**

* Financial Information Exchange: The SRO and the Finance Office depend on each other to exchange relevant financial information related to student applications. The SRO provides the Finance Office with necessary details about the students, such as their application status and any financial requirements or payments they need to fulfill.
* Payment Confirmation: The SRO depends on the Finance Office to confirm whether students have made the required payments. This confirmation is crucial for the SRO to update the application status and proceed with further evaluation or processing.
* Collaboration on Financial Matters: The SRO and the Finance Office collaborate to address any financial-related queries or concerns raised by students. They work together to provide students with accurate and timely information, ensuring a smooth and coordinated approach to financial matters.
* Financial Record-keeping: The Finance Office maintains financial records related to student payments and transactions. The SRO may rely on the Finance Office to access and retrieve these records as needed for administrative purposes, evaluation, or reporting.
* Application Status Updates: The Finance Office communicates with the SRO to update the application status once the required payments are confirmed. This update allows the SRO to proceed with the evaluation process or update the student's status accordingly.

There are some dependencies between **Student Relations Office and the Secretary Office:**

* Application Status Update: The SRO depends on the Secretary Office to update the application status to "registered" once the necessary registration procedures have been completed. The SRO communicates with the Secretary Office to confirm that the student has fulfilled all the requirements and is officially enrolled at the university.
* Communication on Registration Matters: The SRO and the Secretary Office collaborate and communicate to address any registration-related queries or concerns raised by the students. They work together to provide students with accurate and timely information regarding registration procedures, deadlines, or any changes in registration requirements.
* Access to Student Information: The Secretary Office may need access to student information or details provided by the SRO to verify the completion of registration procedures. They depend on the SRO to provide accurate information and updates about each student's application status.
* Collaboration for Administrative Tasks: The SRO and the Secretary Office collaborate on administrative tasks related to student registration. This may include coordinating course selection, issuing student identification cards, or updating student records within the system.

3. Requirements

**3.1 Functional Requirements**

The requirement numbering has a scheme – AS## (AS for Admission System)

|  |  |  |  |
| --- | --- | --- | --- |
| **Req#** | **Requirement** | **Comments** | **Priority** |
| AS\_01 | The system must be supported by a web application. | This is the main platform for using the software. | 2 |
| AS\_02 | The software should have different views for different level of users. | A view with different functionalities for the students , another view with different functionalities for the SRO (admin) , another view for the FO , another view for the SO, and another view for the professors &staff. | 2 |
| AS\_03 | All accounts of the users should be secured by a password. | Only users know their password.  Passwords are going to be hashed in the database. | 1 |
| AS\_04 | Users should be able to register by providing necessary information (name, email address, role). | The system should validate user information and ensure unique usernames and email addresses. | 1 |
| AS\_05 | The system should provide an intuitive and user-friendly online application form. | The form should capture essential information for admission consideration (personal details, academic history, supporting documents). | 2 |
| AS\_06 | The system should allow applicants to submit the application form without the need for creating a login. | The student should be able to submit the form without needing to login to the system. | 1 |
| AS\_07 | The system should store and manage all submitted student applications in a centralized repository. | All the applicant form that have been submitted by the students should be stored. | 2 |
| AS\_08 | The SRO should be able to view a table of pending list, applicant list, transactions list, approved list and registered list. | SRO should be able to view the list of the students when he/she logs in. | 2 |
| AS\_09 | The SRO should be able to open individual applications and access the details provided by the students. | SRO can view a specific student application and view all the detailed information of the student. | 2 |
| AS\_10 | The SRO should be able to add comments and provide feedback within each application. | SRO should be able to add comments next to the student name for feedback and review later on. | 2 |
| AS\_11 | The system should track and update the status of student applications, | The system should be able to update the status as pending, approved, rejected. | 2 |
| AS\_12 | The SRO should be able to update the application status based on their evaluation and coordination with other offices. | SRO should be able to update the status of the students giving the feedback from the other offices. | 2 |
| AS\_13 | The system should enable the Finance Office to view the student list, including pending applications. | FO should be able to view all the lists of the students even the ones he/she can’t update or change. | 2 |
| AS\_14 | The Finance Office should have a separate column for recording student payments and financial transactions. | FO should have new columns where he/she will update the new changes. | 2 |
| AS\_15 | The system should notify the SRO upon confirmation of student payments to update the application status. | SRO should be able to get a notification when the FO has updated the new changes of the students list. | 1 |
| AS\_16 | The Secretary Office should be able to view all student lists. | SO should be able to view all the lists of the students , even the one he/she can’t update. | 2 |
| AS\_17 | The Secretary Office should be able to verify the completion of registration procedures. | SO should be able to check that the registration procedure has been completed. | 2 |
| AS\_18 | The Secretary Office should be able to update the student's status to "registered" upon successful verification. | SO update the student status after the verification has been completed. | 2 |
| AS\_19 | The system should have role-based access control. | Role-based access control ensures secure and controlled access to the University Admission System, granting users appropriate privileges based on their roles. It enhances data security, privacy, and operational efficiency by enforcing access restrictions and maintaining proper segregation of duties. | 2 |
| AS\_20 | Professors and staff members should have restricted access to view student data for statistical purposes only. | Restricting access to student data for statistical purposes only ensures privacy and confidentiality while providing professors and staff members with valuable information for analysis and research. | 1 |
| AS\_21 | Editing capabilities should be limited to authorized personnel. | Only the SRO, FO and SO can edit the students data. | 2 |
| AS\_22 | The system should generate reports and analytics on admission trends, student demographics, and other relevant statistics. | The system should generate data in standard formats (CSV, Excel, etc.) for reporting or integration with other systems. | 3 |
| AS\_23 | System administrators should be able to manage user accounts, roles, and permissions. | Granting system administrators the ability to manage user accounts, roles, and permissions is essential for maintaining the University Admission System's security and functionality. | 1 |
| AS\_24 | The system should ensure performance, security, and regular backups. | The system operates efficiently, with optimal speed and responsiveness. It also includes robust security measures to protect sensitive data from unauthorized access or breaches. Regular backups are essential for data protection and disaster recovery, minimizing the risk of data loss or system failures. | 1 |
| AS\_25 | An administrative interface should be provided for managing system settings and configurations. | A user-friendly interface specifically designed for administrators to easily modify system parameters, such as application deadlines, user roles, email templates, and other configurable settings. | 2 |
| AS\_26 | The system should allow setting and managing application deadlines for different admission cycles. | This functionality enables administrators to define specific timeframes for accepting applications, ensuring that the admissions process adheres to predefined schedules. | 3 |
| AS\_27 | The system should support online payment processing for application fees. | It should integrate with payment gateways to securely handle transactions and provide payment confirmation to applicants. | 3 |
| AS\_28 | The system should allow applicants to upload supporting documents securely. | It should provide verification mechanisms to ensure the authenticity and validity of uploaded documents. | 2 |

**3.2 Non-Fuctional Requirements**

**3.2.1 Product Requirements**

**3.2.1.1 User Interface Requirements**

The application where the main focus will be put on is the web application. It should be compatible for all Chrome, Microsoft Edge and other browsers.

The sketches of the interfaces are going to be attended to the Appendix C.

* The main page of the application is just a login interface.
* It will contain UMT logo

Below the user will choose his identification:

* Student Relations Office
* Finance Office
* Secretary Office
* Proffesors & Staff

After choosing the identifaction role it will direct to the log in interface

* It will contain the UMT logo

Below the user is asked to enter:

* Their email
* Their password
* Followed by a login button

Two links follow the login button:

* The “Forgot your password?” link which will direct to the specific interfaces
* “Don’t have an account?” link which will direct to the specific interfaces
* The “Forgot your password?” link directs the users to another page where:
* where they are asked to enter e-mail address where a code is going to be send to reset the password
* and a send button
* The send button directs the users to another page where:
* the users are asked to enter the code they received in their email address
* and a verify button
* The verify button directs the users to another page where:
* the users are asked to enter their password twice
* and a reset password button that directs the users to the login interface

**The student interface** contains( The student doesn’t require a login so the application forms will be added to the university page)

Two applications form where the student will fill them with their data. They appear in this order:

* The first form of applications where the student will choose their program. It contains:
* A choose your program form
* A menu that contains the type of program the student can choose from such as bachelor application, professional studies application and master&phd application
* A field where they can type or select the program they want to study
* A filed where they will choose the academic year
* A button Continue that sends the students to the next form of application
* The button Continue send the student to the second application form ,it contains the student personal information. It is shown in this order:
* First there is the student information part that contains:
* Name
* Surname
* Birthday
* Place of Birth
* Mobile Phone
* Email address
* The family information part that contains:
* Father name
* Father phone
* Father profession
* Mother name
* Mother profession
* Family address
* The education information part that contains:
* Graduated country
* High school name
* School profikle
* D1(Literature)
* D2(Math)
* D3(Language)
* Z1(Optional exam)
* Language certifications that also has a note below that says : you have to provide the certificate of your English exam
* A button Send that send the application form to the university

**The Student Relations Office Interface** contains:

* A table with all the students list with all the needed information. It is in this order
* First column it is the pending list which contains:
* The number of students that have applied
* A view button where they can view the list of the students
* An edit button where they can edit and make the changes for a specific student
* Second column it is the applicant list which contains:
* The number of students
* A view button
* An edit button
* Third column it is the transactions list which contains:
* The number of students
* A view button
* Fourth column it is the approved list which contains:
* The number of students
* A view list button
* An edit list button
* Fifth column it is the registered list which contains:
* The number of students
* A view list button
* The view button send the user to another page where it contains a table :
* Name
* Surname
* Program
* The edit button send the user to another page where it contains a table:
* Name
* Surname
* Program
* Rejected
* Accepted
* Comments

**The Finance Office** **interface** contains:

* A table with all the students list with all the needed information. It is in this order
* First column it is the pending list which contains:
* The number of students that have applied
* A view button where they can view the list of the students
* Second column it is the applicant list which contains:
* The number of students
* A view button
* Third column it is the transactions list which contains:
* The number of students
* A view button
* An edit list button
* Fourth column it is the approved list which contains:
* The number of students
* A view list button
* Fifth column it is the registered list which contains:
* The number of students
* A view list button
* The view button send the user to another page where it contains a table :
* Name
* Surname
* Transactions
* The edit button send the user to another page where it contains a table:
* Name
* Surname
* Transaction
* Total
* Comments

**The Secreatary Office** **interface** contains:

* A table with all the students list with all the needed information. It is in this order
* First column it is the pending list which contains:
* The number of students that have applied
* A view button where they can view the list of the students
* Second column it is the applicant list which contains:
* The number of students
* A view button
* Third column it is the transactions list which contains:
* The number of students
* A view button
* Fourth column it is the approved list which contains:
* The number of students
* A view list button
* Fifth column it is the registered list which contains:
* The number of students
* A view list button
* An edit button
* The view button send the user to another page where it contains a table :
* Name
* Surname
* Program
* The edit button send the user to another page where it contains a table:
* Name
* Surname
* Program
* Accepted
* Rejected
* Comments

**The Proffesors & Staff** **interface** contains:

* A table with all the students list with all the needed information. It is in this order
* First column it is the pending list which contains:
* The number of students that have applied
* A view button where they can view the list of the students
* Second column it is the applicant list which contains:
* The number of students
* A view button
* Third column it is the transactions list which contains:
* The number of students
* A view button
* Fourth column it is the approved list which contains:
* The number of students
* A view list button
* Fifth column it is the registered list which contains:
* The number of students
* A view list button
* The view button send the user to another page where it contains a table :
* Name
* Surname
* Program

**3.2.2 Usability**

**3.2.2.1 Learnability:**

* The application is easy to use hence no specific training will be needed for the users.
* The application will know the specific users that are allowed to use the application (by the username), so it will not allow random users to login and use the application.

**3.2.3 Performance Requirements**

* The software will be based on web and has to be run from a web server.
* The software should support the SRO, SO, FO , Proffesors&Staff and student that have access to the system.
* The application’s time of execution will depend on the user’s internet connection strength.
* The performance of the application will depend on the number of active user accessing the website

**3.2.3.1 Capacity**

* Every user will use the same database. If multiple requests are made to the server, the request will form a query.
* The application will be stored in a web server.
* The database will not be very large and complex.

4. Analysis Model

**4.1 User Scenarios**

|  |  |  |
| --- | --- | --- |
| **Number** | **User Story Name** | **Description** |
| 1 | Successful Login | User logs in successfully by entering his email and password |
| 2 | Login failed | User fails to login by using his email and password |
| 3 | Password forgotten | User forgets his password, changes it by using a code received to his email account |
| 4 | Language chosen | User changes the language from English to Albania or the other way around |
| 5 | Dark mode on | User switches the display to dark mode |
| 6 | Student applies to the university | Student goes to the university page and fills out the application form |
| 7 | SRO chooses to view the pending list | SRO logs in with their account and chooses to view the pending list that the student have applied from the application form |
| 8 | SRO chooses to edit the pending list | SRO logs in with their account and chooses to edit the pending list where they can make comments about a student or accept or reject their application |
| 9 | SRO chooses to view the applicant list | SRO logs in with their account and chooses to view the applicant list of students |
| 10 | SRO chooses to edit the applicant list | SRO logs in with their account and chooses to edit the applicant list where they can make comments about a student or accept or reject their application |
| 11 | SRO chooses to view the transactions list | SRO logs in with their account and chooses to view the transactions list of students where they have made their payment |
| 12 | SRO chooses to view the approved list | SRO logs in with their account and chooses to view the approved list of students |
| 13 | SRO chooses to edit the approved list | SRO logs in with their account and chooses to edit the approved list where they can make comments about a student or accept or reject their application |
| 14 | SRO chooses to view the registered list | SRO logs in with their account and chooses to view the registered list of students |
| 15 | FO chooses to view the pending list | FO logs in with their account and chooses to view the pending list that the student have applied from the application form |
| 16 | FO chooses to view the applicant list | FO logs in with their account and chooses to view the applicant list of students |
| 17 | FO chooses to view the transactions list | FO logs in with their account and chooses to view the transactions list of students where they have made their payment |
| 18 | FO chooses to edit the transactions list | FO logs in with their account and chooses to edit the transactions list of students where they put if the student has made the transactions or not and the total sum and comment for every student |
| 19 | FO chooses to view the approved list | FO logs in with their account and chooses to view the approved list of students |
| 20 | FO chooses to view the registered list | FO logs in with their account and chooses to view the registered list of students |
| 21 | SO chooses to view the pending list | SO logs in with their account and chooses to view the pending list that the student have applied from the application form |
| 22 | SO chooses to view the applicant list | SO logs in with their account and chooses to view the applicant list of students |
| 23 | SO chooses to view the transactions list | SO logs in with their account and chooses to view the transactions list of students where they have made their payment |
| 24 | SO chooses to view the approved list | SO logs in with their account and chooses to view the approved list of students |
| 25 | SO chooses to view the registered list | SO logs in with their account and chooses to view the registered list of students |
| 26 | SO chooses to edit the registered list | SO logs in with their account and chooses to edit the list where they chooses to accept or reject the student |
| 27 | Proffesors & Staff chooses to view the pending list | Proffesors & Staff logs in with their account and chooses to view the pending list that the student have applied from the application form |
| 28 | Proffesors & Staff chooses to view the applicant list | Proffesors & Staff logs in with their account and chooses to view the applicant list of students |
| 29 | Proffesors & Staff chooses to view the transactions list | Proffesors & Staff logs in with their account and chooses to view the transactions list of students where they have made their payment |
| 30 | Proffesors & Staff chooses to view the approved list | Proffesors & Staff logs in with their account and chooses to view the approved list of students |
| 31 | Proffesors & Staff chooses to view the registered list | Proffesors & Staff logs in with their account and chooses to view the registered list of students |

**4.2 User Scenarios Extended**

1. **User Scenario 1 – Successful Login**
2. The user is asked to enter the username
3. The user is asked to enter the password
4. The user presses the “Login” button
5. If the credentials of the users match in the database, user is authorized to be redirected.
6. The user logs in and is redirected to the main page of the application
7. **User Scenario 2 – Login Failed**
8. The user is asked to enter the username
9. The user is asked to enter the password
10. The user presses the “Login “ button
11. The user credentials don’t match with any on the database
12. An error message is displayed to the user
13. The user tries to enter his credentials again
14. **User Scenario 3 – Password forgotten**
15. The user forgets the password and clicks on the link “Forgot Password?”.
16. The user is redirected to another page where he is asked to enter an email address so a code to reset the password is going to be sent.
17. The user presses the “Send” button.
18. The user is redirected to another page where he is asked to enter the code that was received in his email address.
19. The user presses the “Verify” button.
20. The user is redirected to another page where he is asked to enter the new password twice.
21. The user presses the “Reset Password” button.
22. The user is redirected to the login page.
23. The user enters his email address and the new password.
24. The user presses the “Login” button.
25. The user is redirected to the main page of the application.
26. **User Scenario 4 - Language choosen**
27. The user is logged in the system.
28. The user’s main page will be the notification page.
29. The user clicks on the Settings menu.
30. The user clicks on the Language drop down menu.
31. English, Turkish, Albanian appears.
32. The user chooses the language they want.
33. **User Scenario 5 – Dark mode**
34. The user is logged in the system.
35. The user’s main page will be the notification page.
36. The user clicks on the Settings menu.
37. The user switches display to dark mode.
38. **User Scenario 6 - Student applies to the university**
39. Student goes to the university page
40. Students opens the application form
41. Student chooses the application program (bachelor , professional studies or master&phd application)
42. Student types or selects the degree they want
43. Student chosses the academic year
44. Student presses the “Continue” button to be send to the next application form
45. Student fills out his information suh as name , surname, birthday, place of birth, mobile phone and email
46. Student fills out the family information
47. Student fills out the education information
48. Student send the application
49. **User Scenario 7- SRO chooses to view the pending list**
50. SRO logs into the system
51. SRO has a display of all the list of students
52. SRO clicks the “View” button to see the pending list of students
53. SRO can see the list with the information of the student such as name , surname and the program
54. SRO close the list
55. SRO logs out
56. **User Scenario 8 - SRO chooses to edit the pending list**
57. SRO logs into the system
58. SRO has a display of all the list of students
59. SRO clicks the “View” button to edit the pending list of students
60. If the students completes the criteria SRO accepts their application
61. If the students don’t meet the criteria SRO rejects their application
62. SRO can make comments for each students to help with the rejection and acceptness
63. SRO saves the edits
64. SRO close the list
65. SRO logs out
66. **User Scenario 9 - SRO chooses to view the applicant list**
67. SRO logs into the system
68. SRO has a display of all the list of students
69. SRO clicks the “View” button to see the applicant list of students
70. SRO can see the list with the information of the student such as name , surname and the program
71. SRO close the list
72. SRO logs out
73. **User Scenario 10 - SRO chooses to edit the applicant list**
74. SRO logs into the system
75. SRO has a display of all the list of students
76. SRO clicks the “View” button to edit the applicant list of students
77. If the students completes the criteria SRO accepts their application
78. If the students don’t meet the criteria SRO rejects their application
79. SRO can make comments for each students to help with the rejection and acceptness
80. SRO saves the edits
81. SRO close the list
82. SRO logs out
83. **User Scenario 11 - SRO chooses to view the transactions list**
84. SRO logs into the system
85. SRO has a display of all the list of students
86. SRO clicks the “View” button to view the transactions list
87. SRO can see the transactions that each students has made
88. SRO closes the list
89. SRO logs out
90. **User Scenario 12 - SRO chooses to view the approved list**
91. SRO logs into the system
92. SRO has a display of all the list of students
93. SRO clicks the “View” button to see the approved list of students
94. SRO can see the list with the information of the student such as name , surname and the program
95. SRO close the list
96. SRO logs out
97. **User Scenario 13 - SRO chooses to edit the approved list**
98. SRO logs into the system
99. SRO has a display of all the list of students
100. SRO clicks the “View” button to edit the approved list of students
101. If the students completes the criteria SRO accepts their application
102. If the students don’t meet the criteria SRO rejects their application
103. SRO can make comments for each students to help with the rejection and acceptness
104. SRO saves the edits
105. SRO close the list
106. SRO logs out
107. **User Scenario 14 - SRO chooses to view the registered list**
108. SRO logs into the system
109. SRO has a display of all the list of students
110. SRO clicks the “View” button to see the registered list of students
111. SRO can see the list with the information of the student such as name , surname and the program
112. SRO close the list
113. SRO logs out
114. **User Scenario 15 - FO chooses to view the pending list**
115. FO logs into the system
116. FO has a display of all the list of students
117. FO clicks the “View” button to see the pending list of students
118. FO can see the list with the information of the student such as name , surname and the program
119. FO close the list
120. FO logs out
121. **User Scenario 16 - FO chooses to view the applicant list**
122. FO logs into the system
123. FO has a display of all the list of students
124. FO clicks the “View” button to see the applicant list of students
125. FO can see the list with the information of the student such as name , surname and the program
126. FO close the list
127. FO logs out
128. **User Scenario 17 - FO chooses to view the transactions list**
129. FO logs into the system
130. FO has a display of all the list of students
131. FO clicks the “View” button to view the transactions list
132. FO can see the transactions that each students has made
133. FO closes the list
134. FO logs out
135. **User Scenario 18 - FO chooses to edit the transactions list**
136. FO logs into the system
137. FO has a display of all the list of students
138. FO clicks the “View” button to edit the approved list of students
139. If the students has made the payment FO puts “Yes” for that application
140. If the students hasn’t made the payment FO puts “No” for that application
141. FO can make comments for each students to help with the rejection and acceptness
142. FO puts the total of the transactions the student has made
143. FO saves the edits
144. FO close the list
145. FO logs out
146. **User Scenario 19 - FO chooses to view the approved list**
147. FO logs into the system
148. FO has a display of all the list of students
149. FO clicks the “View” button to see the approved list of students
150. FO can see the list with the information of the student such as name , surname and the program
151. FO close the list
152. FO logs out
153. **User Scenario 20 - FO chooses to view the registered list**
154. FO logs into the system
155. FO has a display of all the list of students
156. FO clicks the “View” button to see the registered list of students
157. FO can see the list with the information of the student such as name , surname and the program
158. FO close the list
159. FO logs out
160. **User Scenario 21 - SO chooses to view the pending list**
161. SO logs into the system
162. SO has a display of all the list of students
163. SO clicks the “View” button to see the pending list of students
164. SO can see the list with the information of the student such as name , surname and the program
165. SO close the list
166. SO logs out
167. **User Scenario 22 - SO chooses to view the applicant list**
168. SO logs into the system
169. SO has a display of all the list of students
170. SO clicks the “View” button to see the applicant list of students
171. SO can see the list with the information of the student such as name , surname and the program
172. SO close the list
173. SO logs out
174. **User Scenario 23 - SO chooses to view the transactions list**
175. SO logs into the system
176. SO has a display of all the list of students
177. SO clicks the “View” button to view the transactions list
178. SO can see the transactions that each students has made
179. SO closes the list
180. SO logs out
181. **User Scenario 24 - SO chooses to view the approved list**
182. SO logs into the system
183. SO has a display of all the list of students
184. SO clicks the “View” button to see the approved list of students
185. SO can see the list with the information of the student such as name , surname and the program
186. SO close the list
187. SO logs out
188. **User Scenario 25 - SO chooses to view the registered list**
189. SO logs into the system
190. SO has a display of all the list of students
191. SO clicks the “View” button to see the registered list of students
192. SO can see the list with the information of the student such as name , surname and the program
193. SO close the list
194. SO logs out
195. **User Scenario 26 - SO chooses to edit the registered list**
196. SO logs into the system
197. SO has a display of all the list of students
198. SO clicks the “View” button to edit the registered list of students
199. If the students completes the criteria SO accepts their application
200. If the students don’t meet the criteria SO rejects their application
201. SO can make comments for each students to help with the rejection and acceptness
202. SO saves the edits
203. SO close the list
204. SO logs out
205. **User Scenario 27 - Proffesors & Staff chooses to view the pending list**
206. Proffesors & Staff logs into the system
207. Proffesors & Staff has a display of all the list of students
208. Proffesors & Staff clicks the “View” button to see the pending list of students
209. Proffesors & Staff can see the list with the information of the student such as name , surname and the program
210. Proffesors & Staff close the list
211. Proffesors & Staff logs out
212. **User Scenario 28 - Proffesors & Staff chooses to view the applicant list**
213. Proffesors & Staff logs into the system
214. Proffesors & Staff has a display of all the list of students
215. Proffesors & Staff clicks the “View” button to see the applicant list of students
216. Proffesors & Staff can see the list with the information of the student such as name , surname and the program
217. Proffesors & Staff close the list
218. Proffesors & Staff logs out
219. **User Scenario 29 - Proffesors & Staff chooses to view the transactions list**
220. Proffesors & Staff logs into the system
221. Proffesors & Staff has a display of all the list of students
222. Proffesors & Staff clicks the “View” button to view the transactions list
223. Proffesors & Staff can see the transactions that each students has made
224. Proffesors & Staff closes the list
225. Proffesors & Staff logs out
226. **User Scenario 30 - Proffesors & Staff chooses to view the approved list**
227. Proffesors & Staff logs into the system
228. Proffesors & Staff has a display of all the list of students
229. Proffesors & Staff clicks the “View” button to see the approved list of students
230. Proffesors & Staff can see the list with the information of the student such as name , surname and the program
231. Proffesors & Staff close the list
232. Proffesors & Staff logs out
233. **User Scenario 31 - Proffesors & Staff chooses to view the registered list**
234. Proffesors & Staff logs into the system
235. Proffesors & Staff has a display of all the list of students
236. Proffesors & Staff clicks the “View” button to see the registered list of students
237. Proffesors & Staff can see the list with the information of the student such as name , surname and the program
238. Proffesors & Staff close the list
239. Proffesors & Staff logs out

**4.3 Use cases**

|  |  |
| --- | --- |
| **Name** | **User Login** |
| Summary | The user enters valid credentials and they log in the system. |
| Actor | Student Relations Office, Finance Office, Registration Office and Proffesors & Staff |
| Description | 1. The user is asked to enter the username 2. The user is asked to enter the password 3. The user presses the “Login” button 4. If the credentials of the users match in the database, user is authorized to be redirected. 5. The user logs in and is redirected to the main page of the application |
| Precondition | The user must have signed up before and must have an existing account |
| Alternative | - |
| Post condition | Gain access to the application |

|  |  |
| --- | --- |
| **Name** | **Submit new application** |
| Summary | Student wants to apply to the university and provide necessary information to the admission system. |
| Actor | Student , Student Relations Office |
| Description | 1. The student navigates to the admission system website and selects the "Apply Now" button. 2. The admission system presents the student with a form requesting personal information, academic history, and other relevant details. 3. The student fills out the application form, ensuring that all required fields are completed and all information is accurate. 4. The student reviews the application to ensure all information is correct and complete. 5. The student submits the application. |
| Precondition | The student has access to the admission system website.  The student has not yet submitted an application for the current academic term. |
| Alternative | Student Relations Office can apply for the student |
| Post condition | The admission system has recorded the student's application information. |

|  |  |
| --- | --- |
| **Name** | **View and edits the pending applications** |
| Summary | SRO wants to view and edit the pending applications for all the student that have applied |
| Actor | Student Relations Office |
| Description | 1. The Student Relations Office selects the option to view pending student applications 2. The system presents a list of pending applications with the student's information and status 3. The Student Relations Office selects a pending application to review and add comments 4. The system displays the student's application and provides a comment section for the Student Relations Office to add their opinion 5. The Student Relations Office fills out the comment section and submits it |
| Precondition | The Student Relations Office is logged into the system  There are pending student applications in the system  The Student Relations Office has access to view and edit pending applications |
| Alternative | - |
| Post condition | The pending application is updated with the Student Relations Office's comments and the status is changed to "Applicant" |

|  |  |
| --- | --- |
| **Name** | **View and edit the applicant applications** |
| Summary | SRO wants to view and edit the applicant applications for all the student that have applied |
| Actor | Student Relations Office |
| Description | 1. SRO navigates to the 'Applicant Applications' list. 2. SRO selects the application that they want to review. 3. SRO reviews the application and decides whether to approve or reject it. 4. If SRO accepts the application, they change the status to 'Accepted'. 5. If SRO rejects the application, they change the status to 'Rejected'. 6. SRO saves the changes. |
| Precondition | SRO is logged into the admission system  There is at least one application with a status of 'Pending' |
| Alternative | - |
| Post condition | The application status is updated in the admission system.  The applicant is notified of the status change if necessary.  The application is removed from the 'Pending Applications' list. |

|  |  |
| --- | --- |
| **Name** | **Approve application** |
| Summary | Student Relations Office changes the status of an application from applicant to approved |
| Actor | Student Relations Office, Finance Office |
| Description | 1. SRO navigates to the 'Applicant Applications' list. 2. SRO selects the application that they want to review. 3. SRO verifies that the applicant has paid the required fees by checking the status of the application with the Finance Office. 4. If SRO accepts the application, they change the status to 'Accepted'. 5. If SRO rejects the application, they change the status to 'Rejected'. 6. SRO saves the changes. |
| Precondition | SRO is logged into the admission system  There is at least one application with a status of 'Applicant' |
| Alternative | - |
| Post condition | The application status is updated in the admission system.  The Finance Office is notified of the status change.  The application is removed from the 'Applicant Applications' list. |

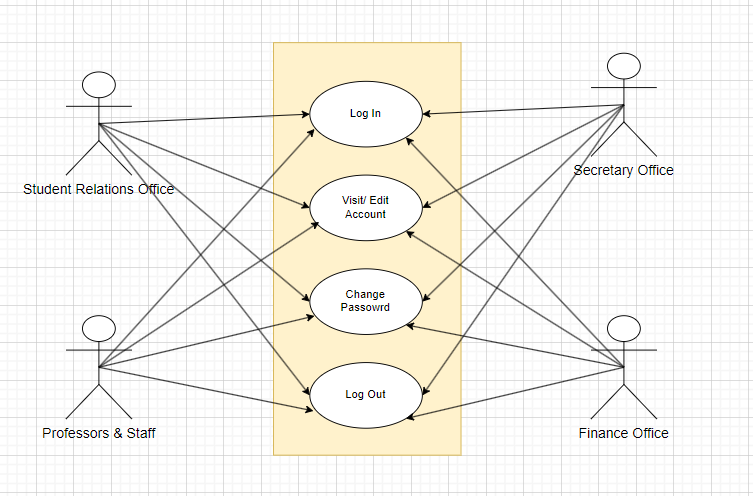
|  |  |
| --- | --- |
| **Name** | **View and edit the transactions applications** |
| Summary | Finance Office verifying that the student has paid the application fee |
| Actor | Finance Office |
| Description | 1. The finance office logs in to the University Admission System. 2. The finance office accesses the list of transactions list of students 3. For each student in the list, the finance office verifies that the student has paid the application fee. 4. If the student has paid the fee, the finance office update it to “Yes” 5. If the student has not paid the fee, the finance office updates it to “No” |
| Precondition | The finance office has access to the list of transactions applications. |
| Alternative | - |
| Post condition | The Finance Office has confirmed that the student has paid the application fee.  The Student Relations Office is notified of the student's payment status. |

|  |  |
| --- | --- |
| **Name** | **Register New Students** |
| Summary | Secretary Office register new students |
| Actor | Secretary Office |
| Description | 1. The Secretary Office logs into the system. 2. The Secretary Office accesses the list of approved students. 3. The Secretary Office selects an approved student from the list. 4. If SO accepts the application, they change the status to 'Accepted'. 5. If SO rejects the application, they change the status to 'Rejected'. |
| Precondition | The Secretary Office has received a list of approved students from the Student Relations Office. |
| Alternative | - |
| Post condition | The new students are registered in the system |

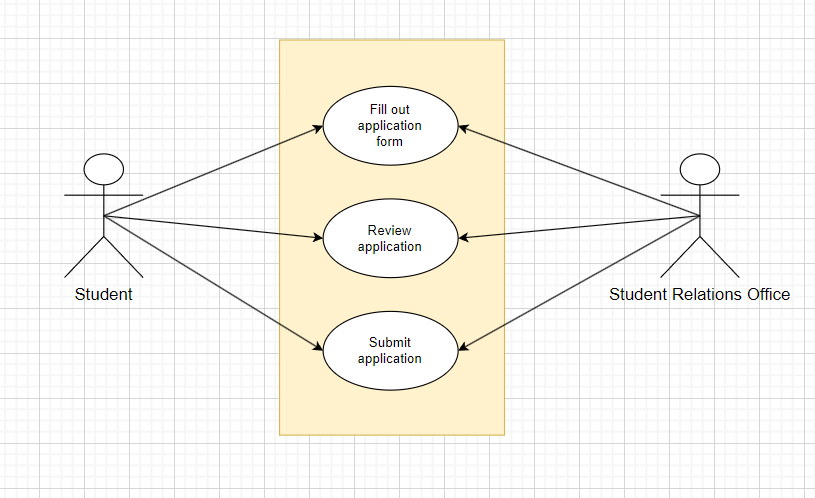
**4.4 Behavioral Diagrams**

**4.4.1 Use Case Diagrams**

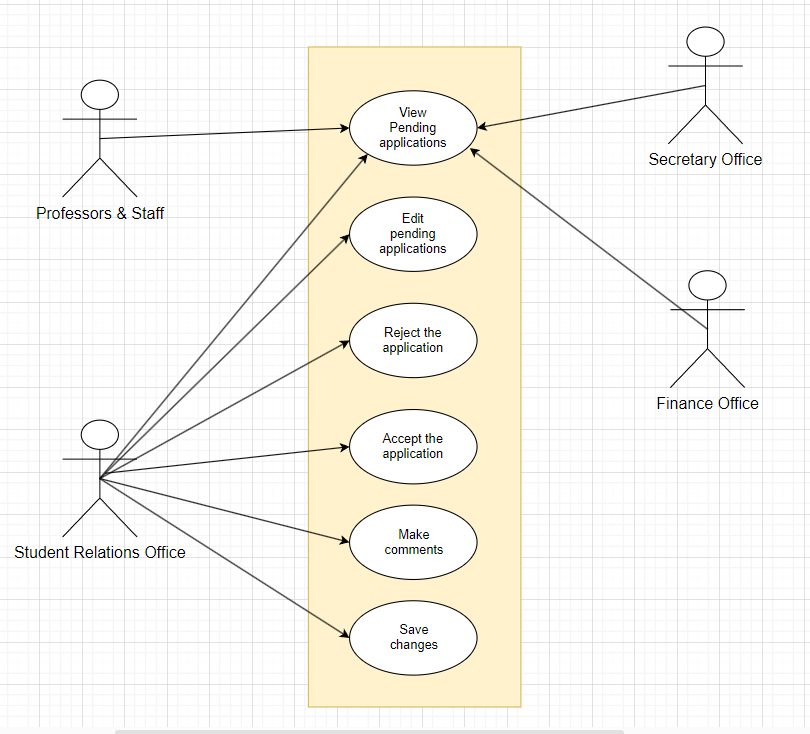
**User Login**

****

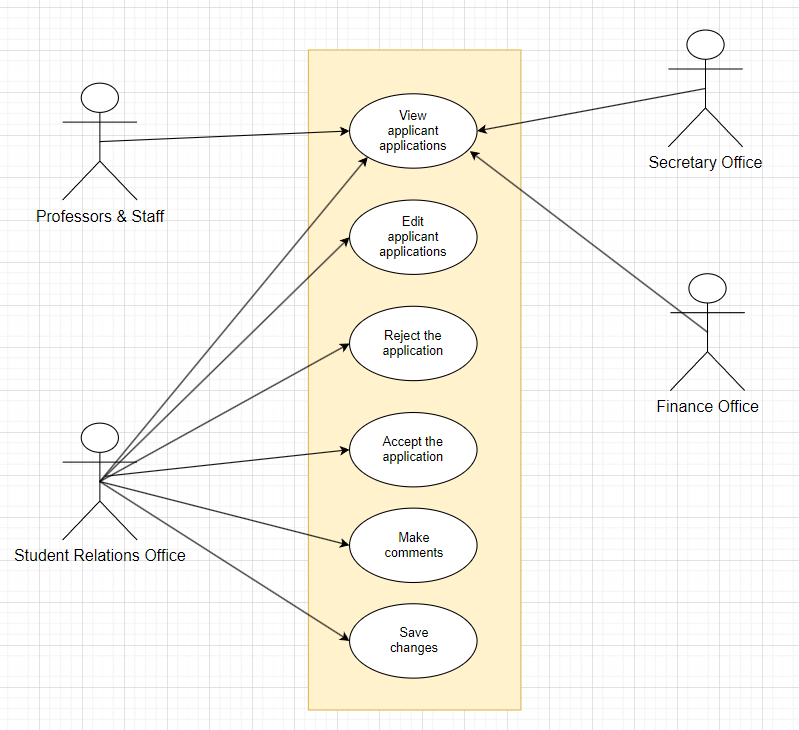
**Submit new application**

****

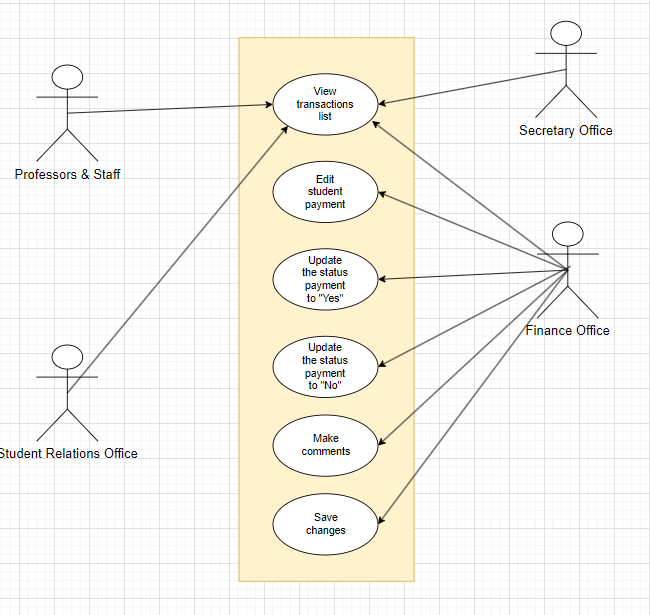
**View and edits the pending applications**

****

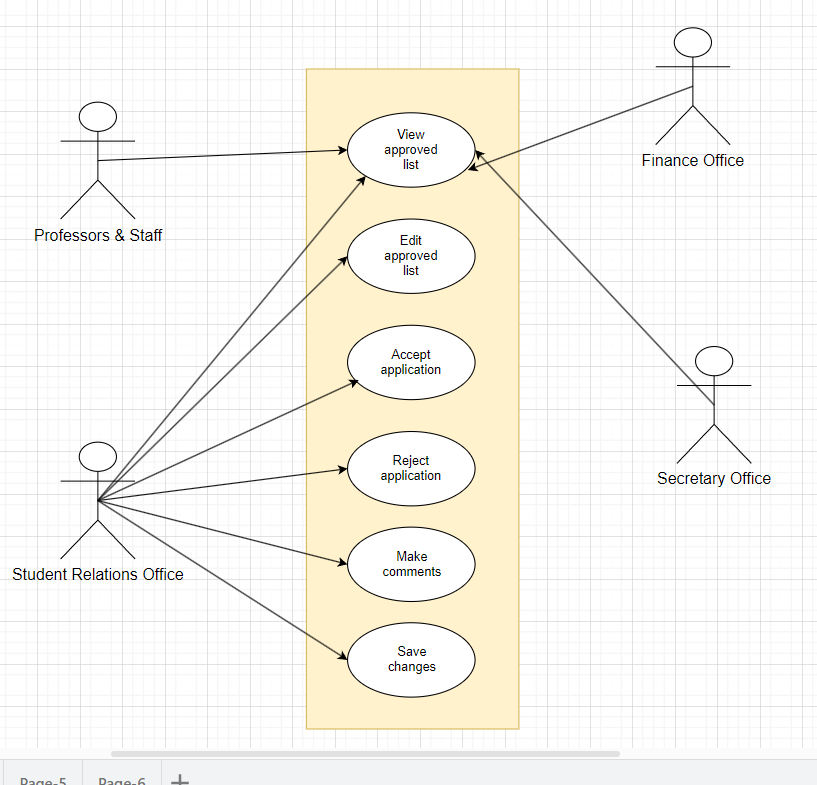
**View and edit the applicant applications**

****

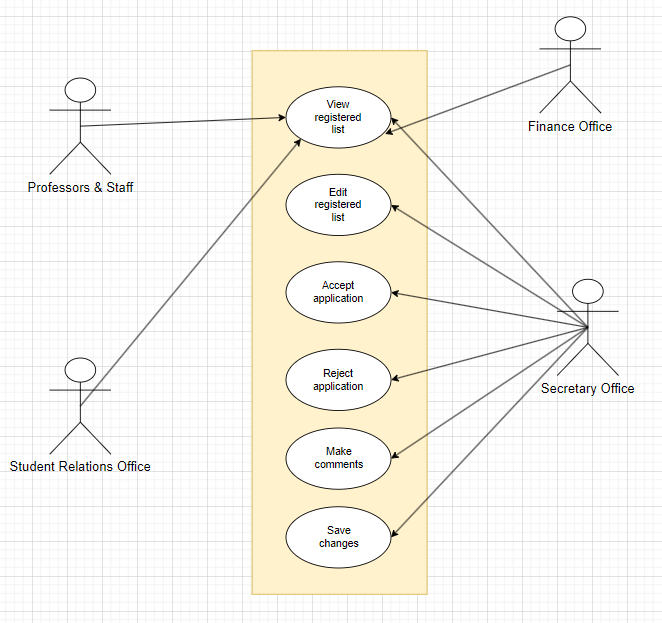
**View and edit the transactions applications**

****

**Approve application**

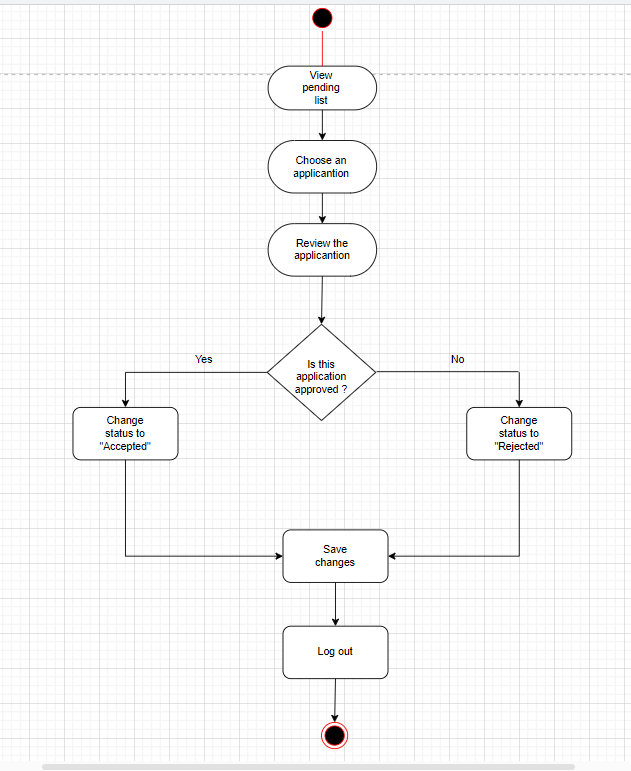
****

**Register New Students**

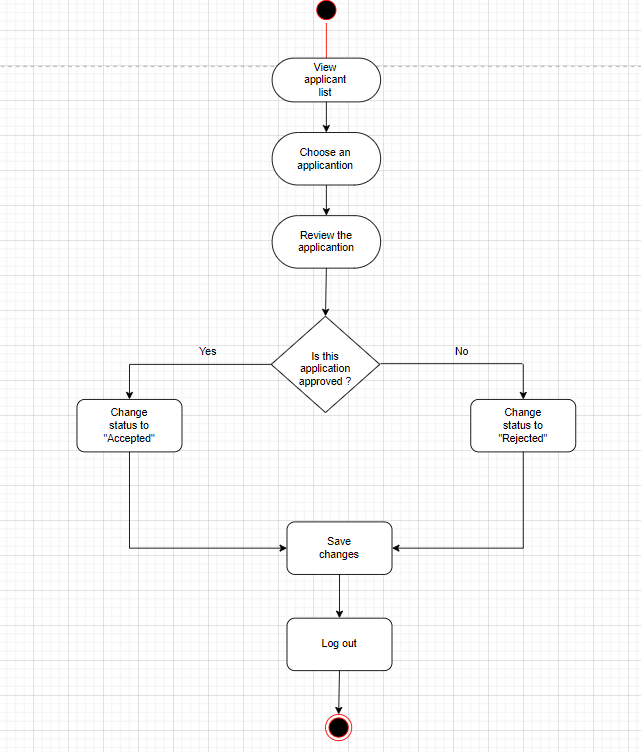
****

**4.4.2 Activity Diagrams**

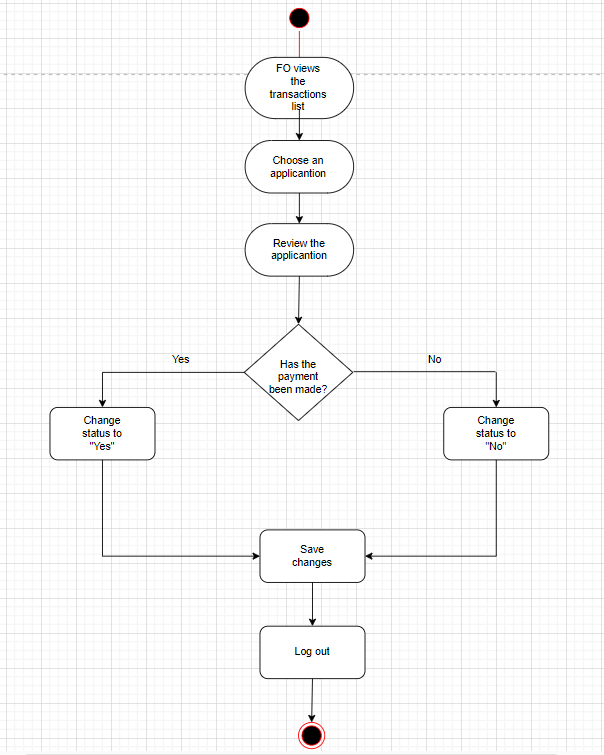
**View and edits the pending applications**

****

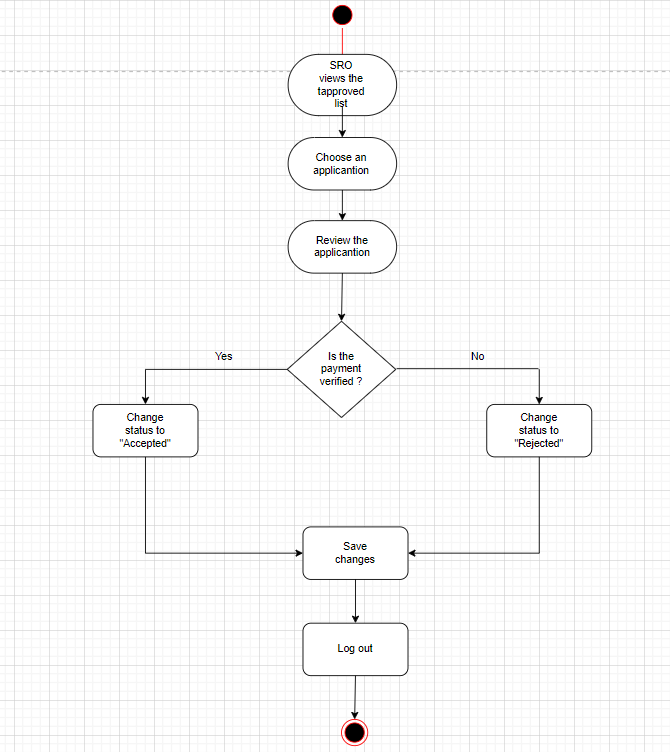
**Applicant applications**

****

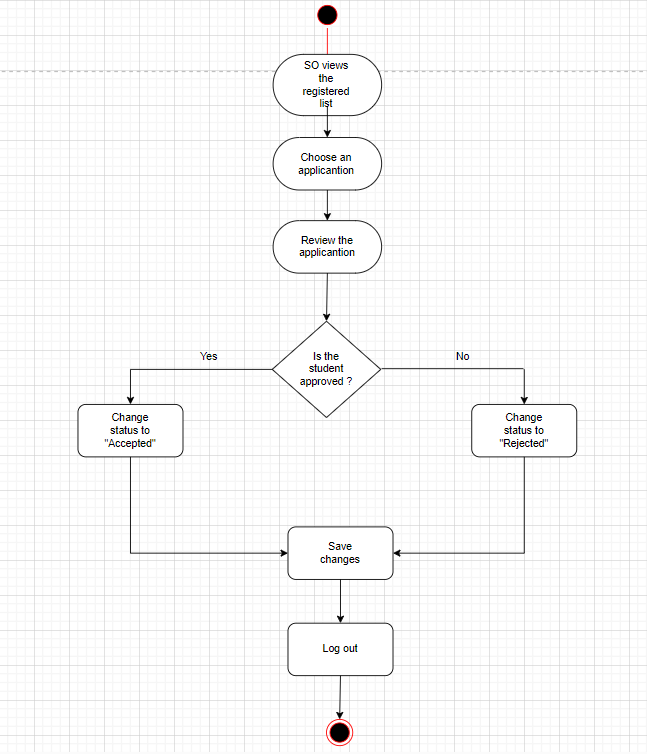
**Transactions applications**

****

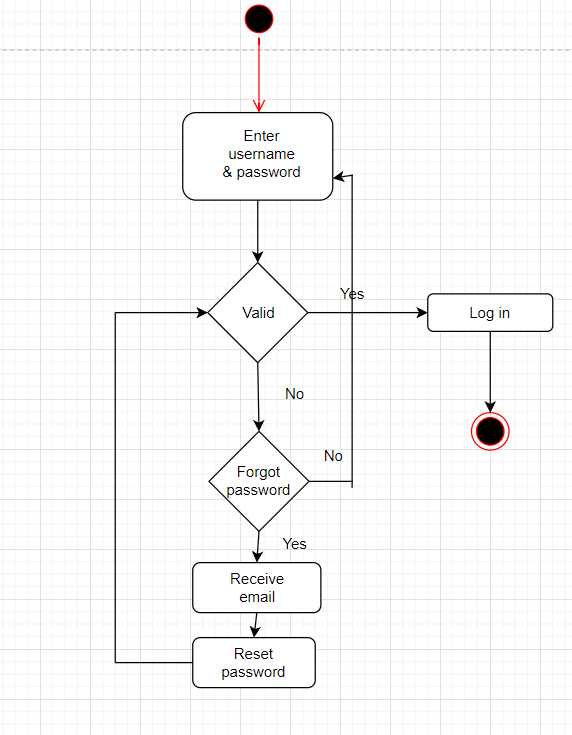
**Approve application**

****

**Register New Students**

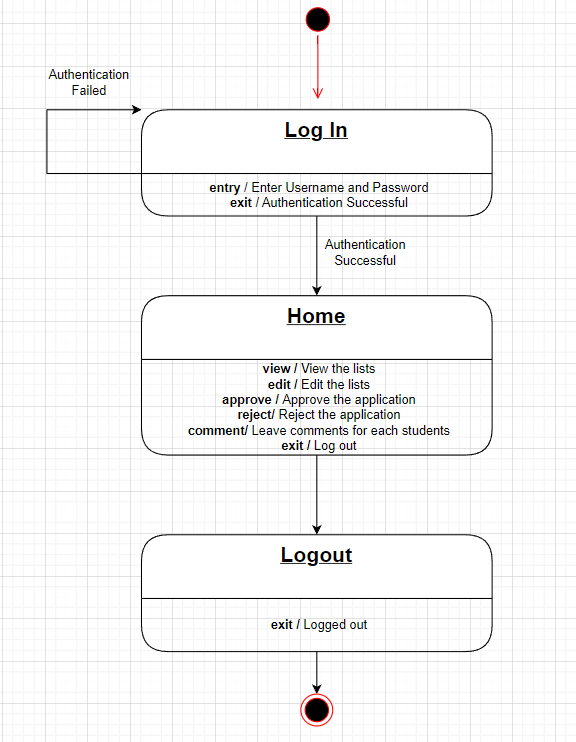
****

**Login**

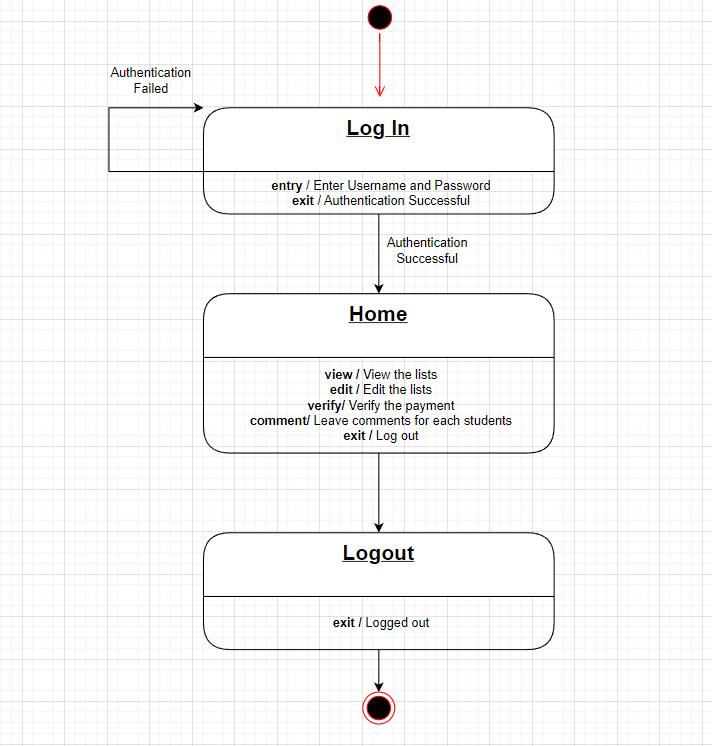
****

**4.4.3 State Diagram**

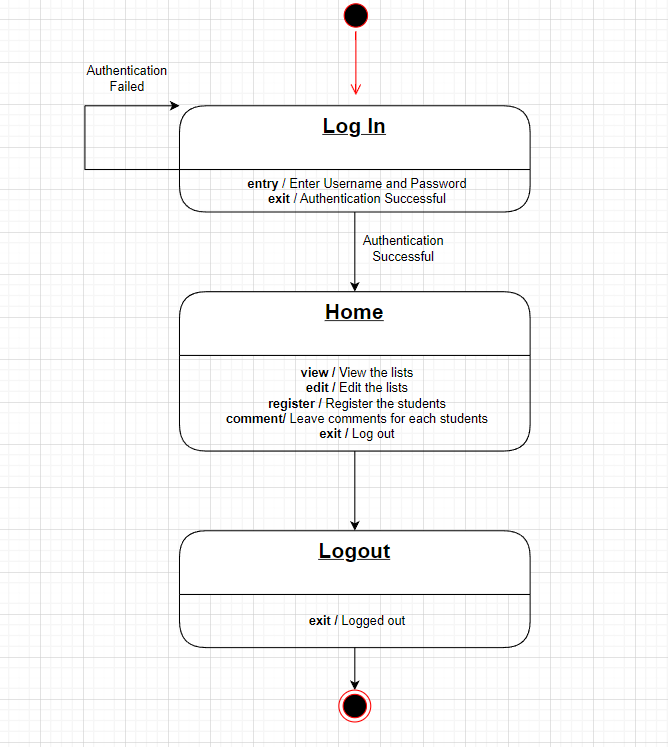
**SRO**

****

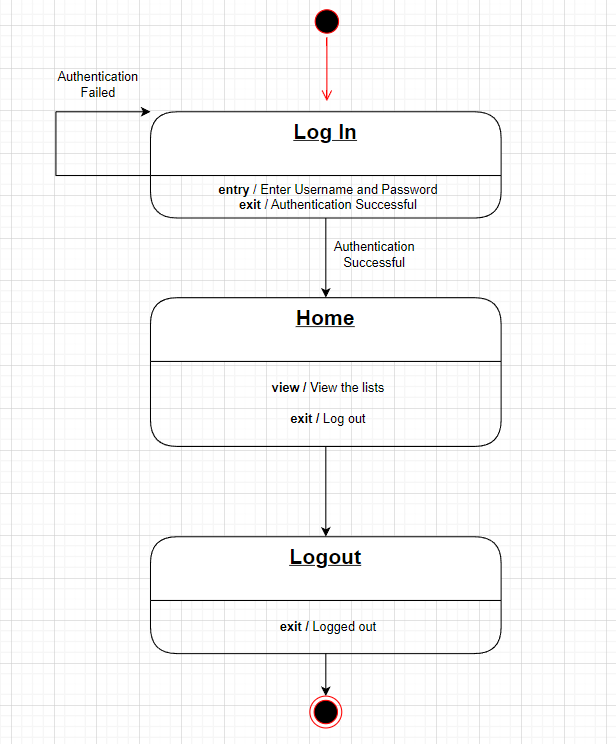
**FO**

****

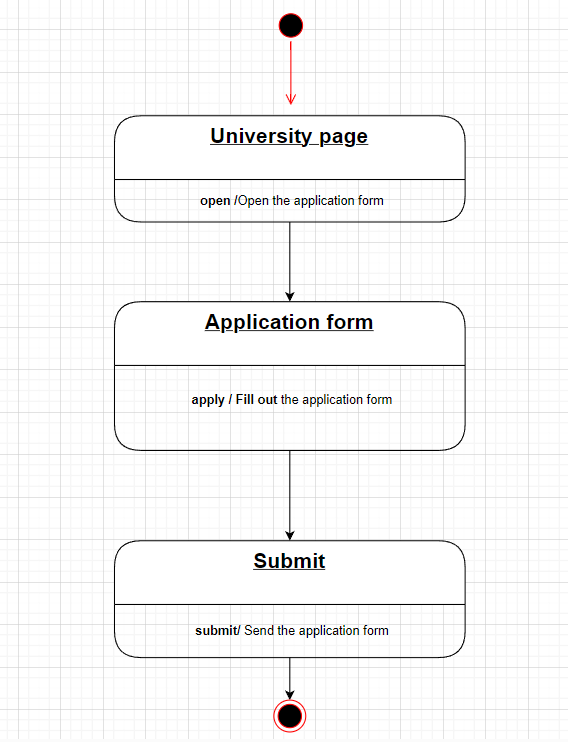
**SO**

****

**Proffesors & Staff**

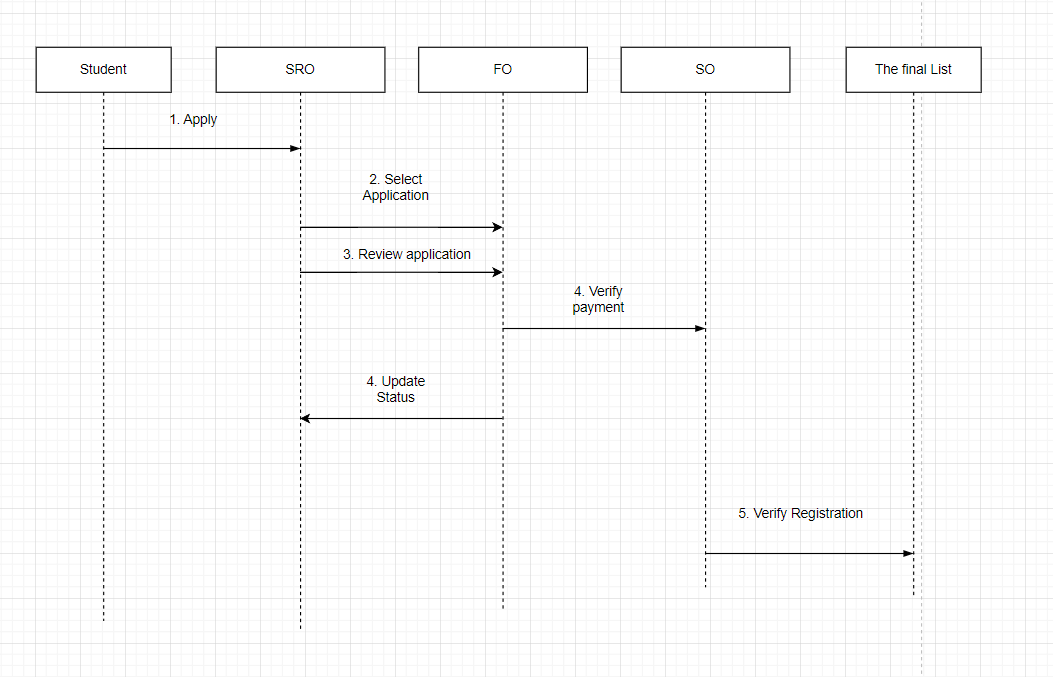
****

**Student**

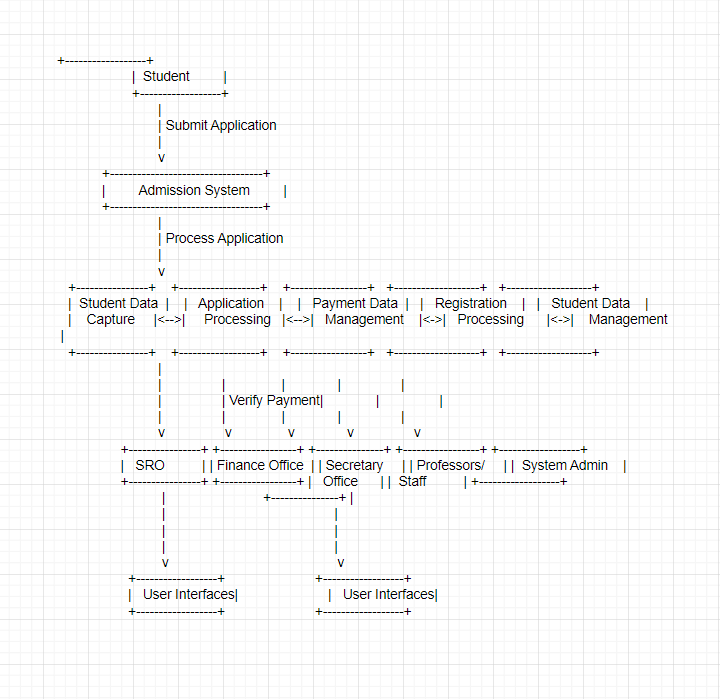
****

**4.5 Interaction Diagram**

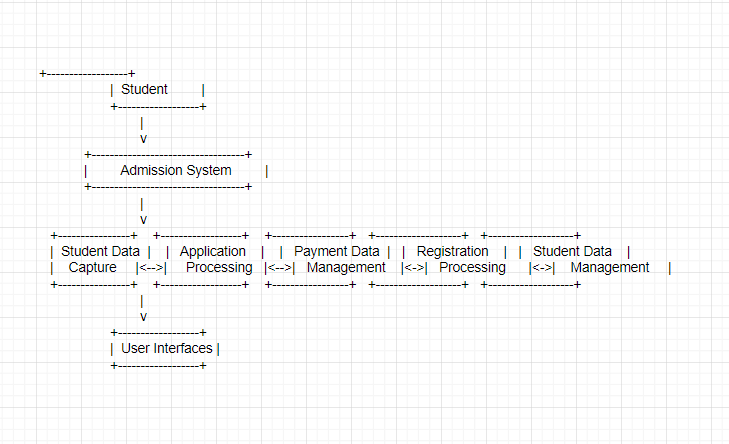
**4.5.1 Sequence Diagram**

****

**4.5.2 Collaboration Diagram**

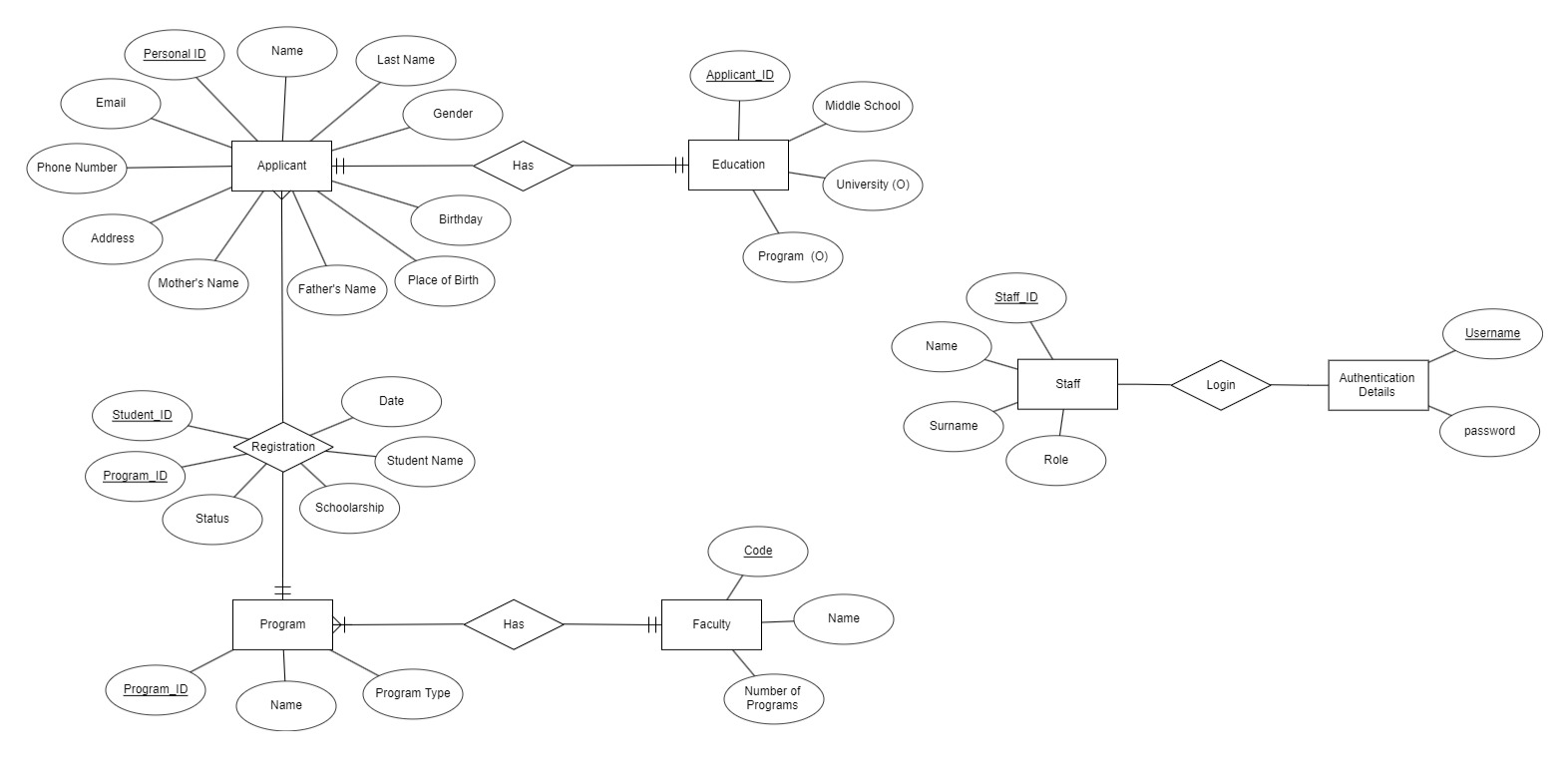
****

**4.6 Data Flow**

****

**4.7 Structural Diagram**

**4.7.1 ER Diagram**

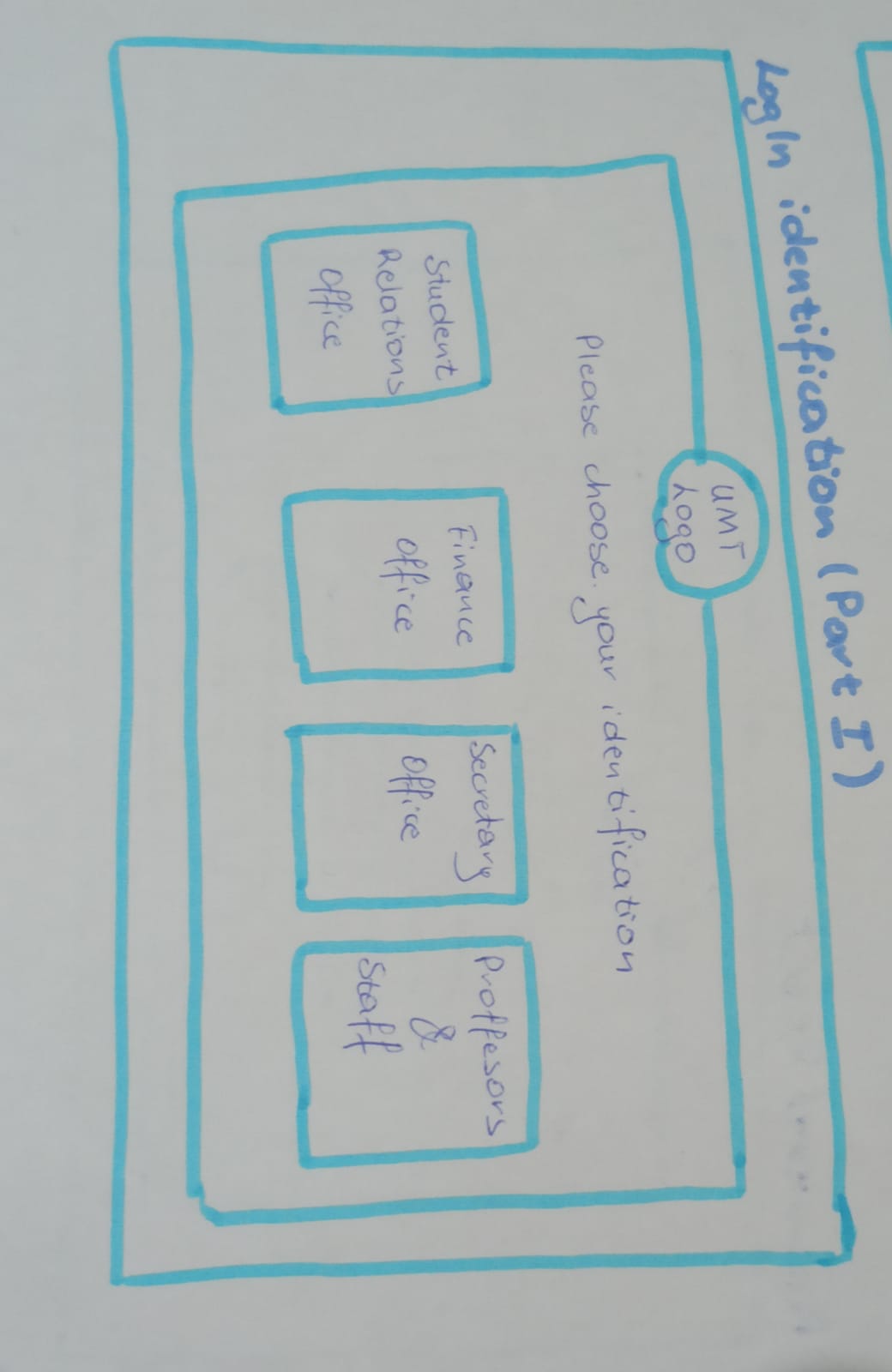
****

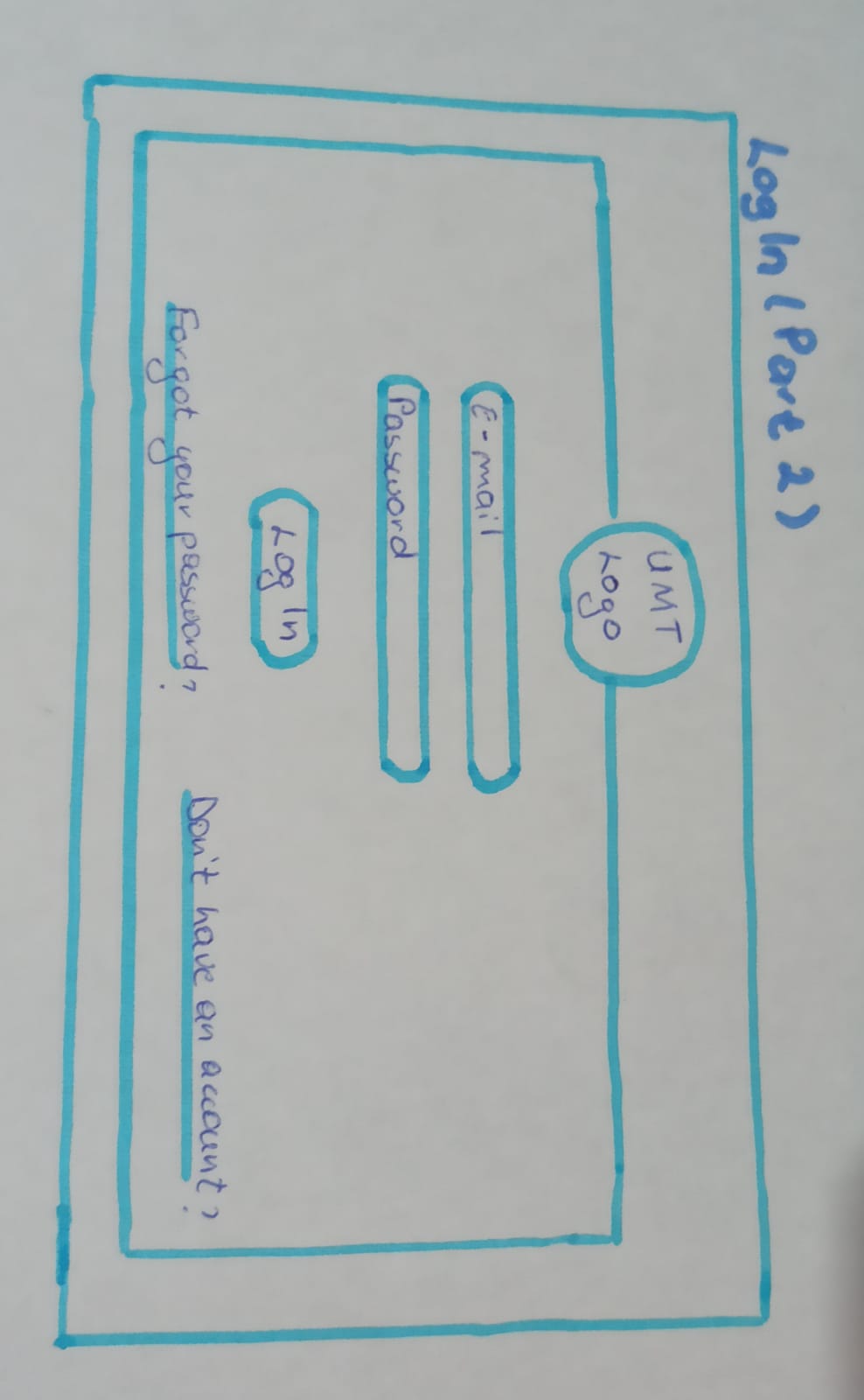
**5.Project Planning**

|  |  |
| --- | --- |
| **Team Member** | **Duty** |
| Joana | Talking with the client and taking the requirements from them  Doing the use case extended  Doing collaboration diagram  Doing data flow diagram  Editing the documentation file  Building the secretary office section(html,css,javascript)  Building the professors&staff section(html,css,javascript)  Building the back-end funcionalities (php) |
| Trejzi | Talking with the client and taking the requirements from them  Doing the state and sequence diagram  Doing the ER and and DB schema diagram  Building the login &sign up section (html,css,javascript)  Building the student relations office section(html,css,javascript)  Building the finance office section(html,css,javascript)  Building the back-end funcionalities (php) |
| Metjon | Talking with the client and taking the requirements from them  Doing the use case and activity diagram  Doing the student interface  Doing the finance office interface |
| Martin | Talking with the client and taking the requirements from them  Doing the weekly meeting reports  Doing the secretary office interface  Doing the proffesors & staff interface |
| Eduart | Talking with the client and taking the requirements from them  Making a list of requirements  Doing the sketches and first draft of designs of the application  Doing the student application form (html, css , javascript) |

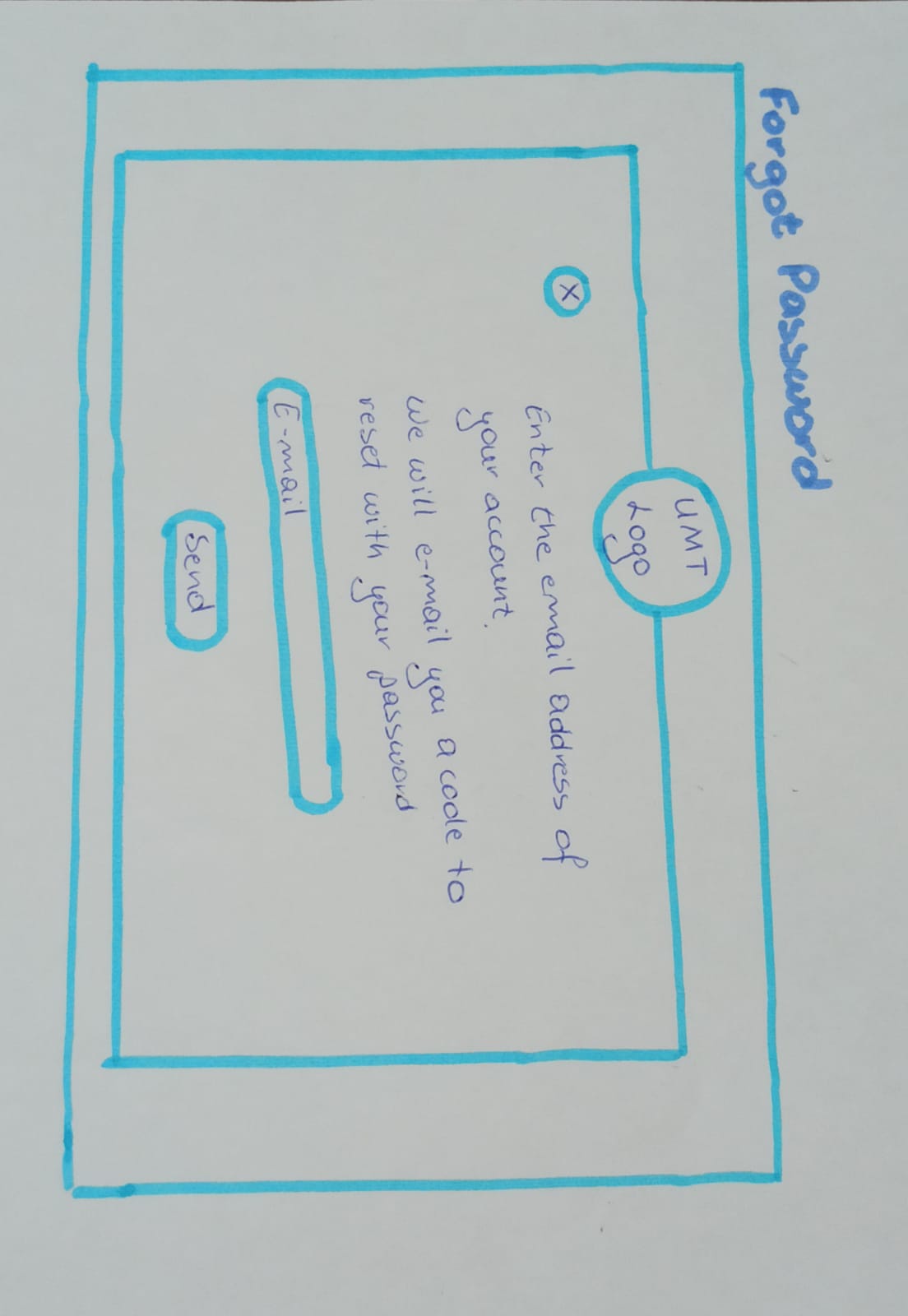
**Appendix A Sketches**

**1 .LOG IN**

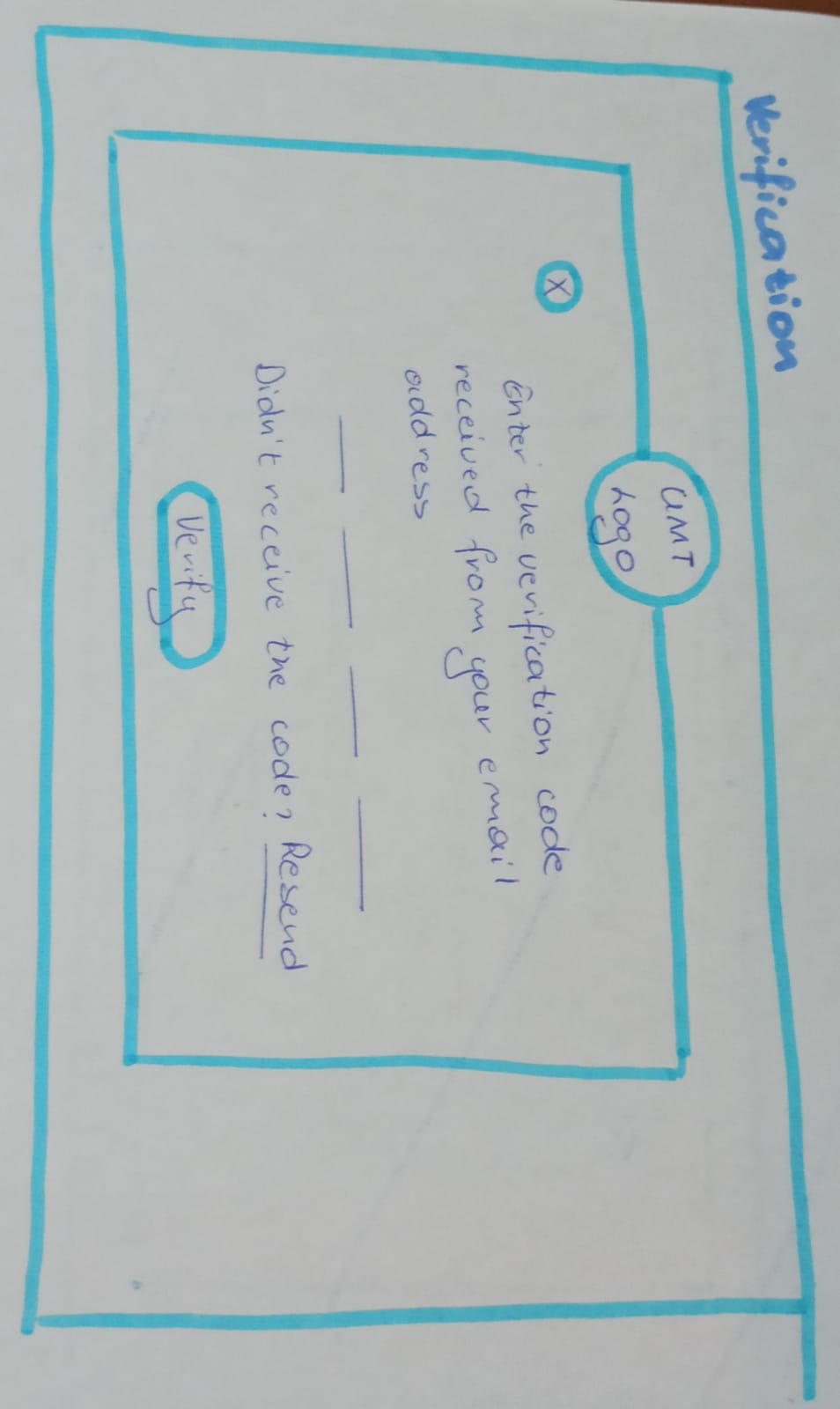


****

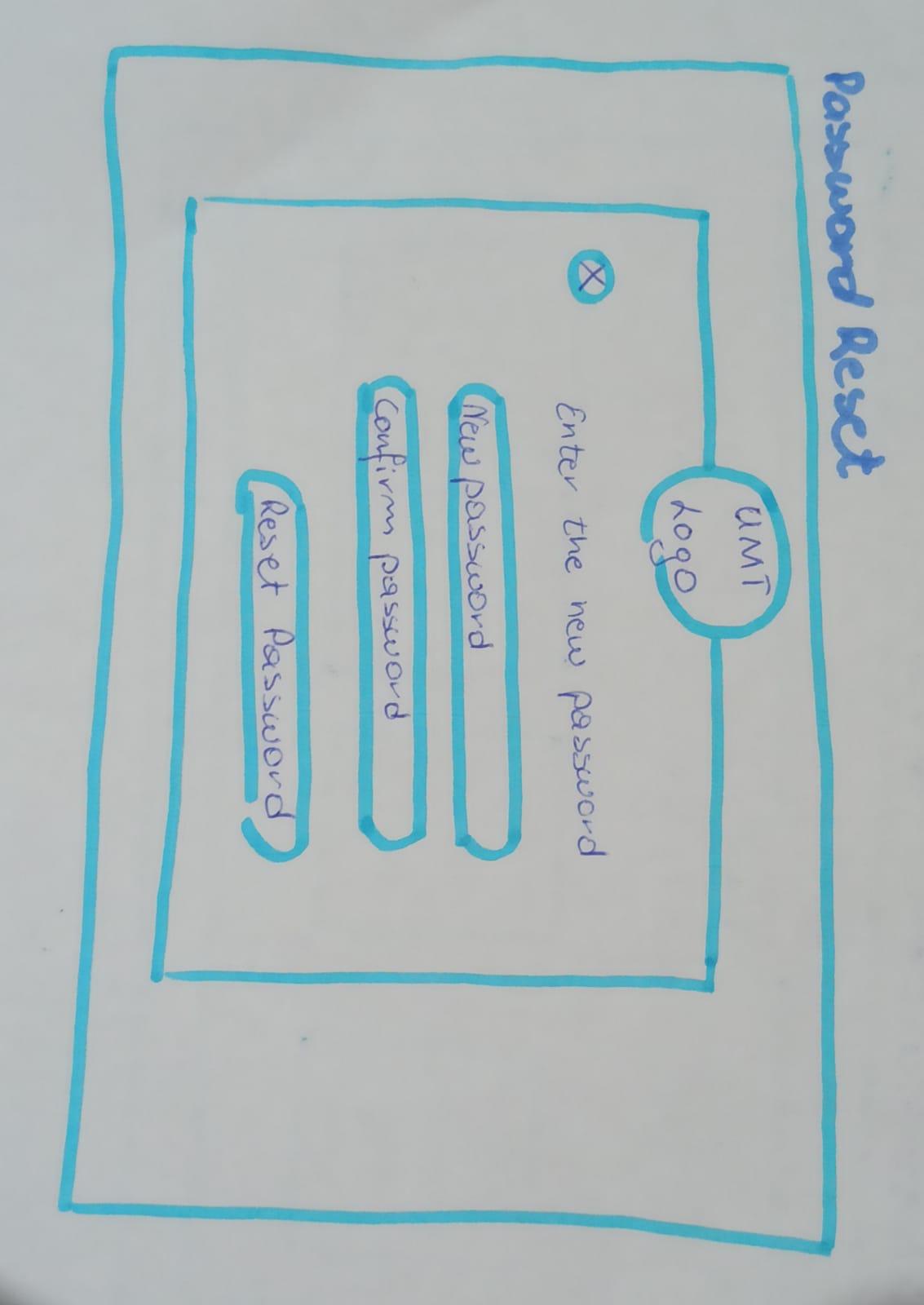
**2. FORGOT PASSWORD**

****

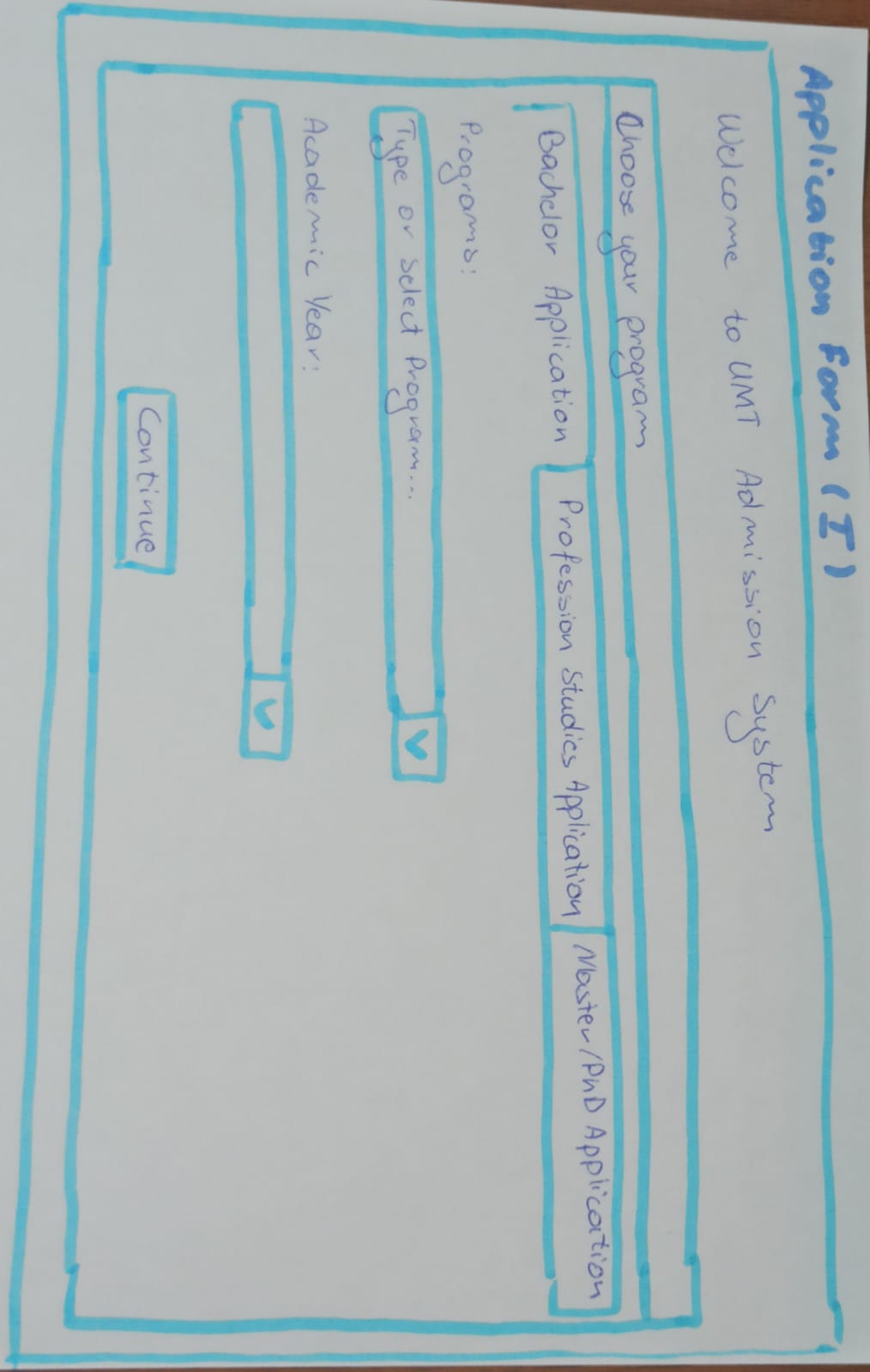
**3. VERIFICATION**

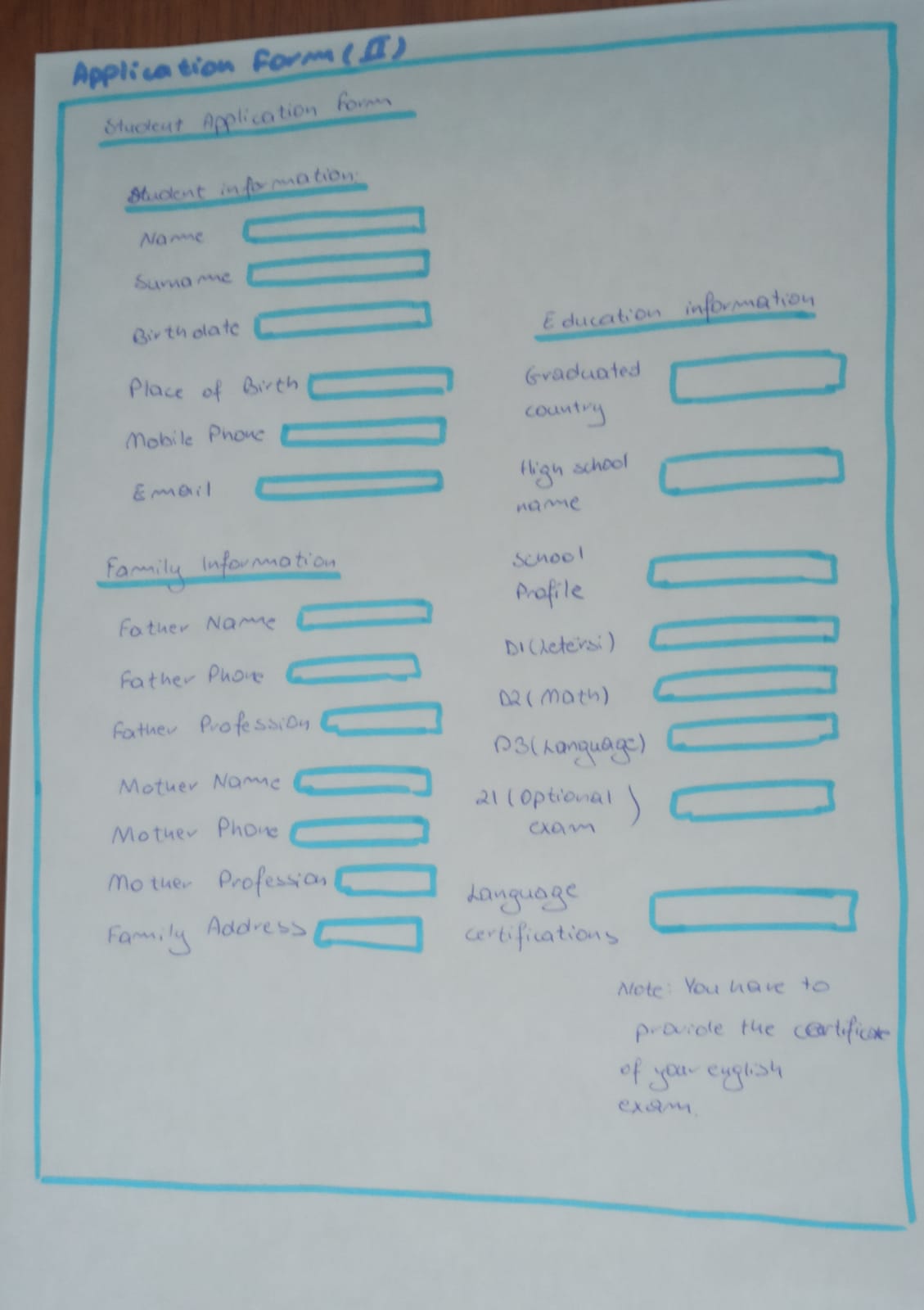
****

**4. PASSWORD RESET**

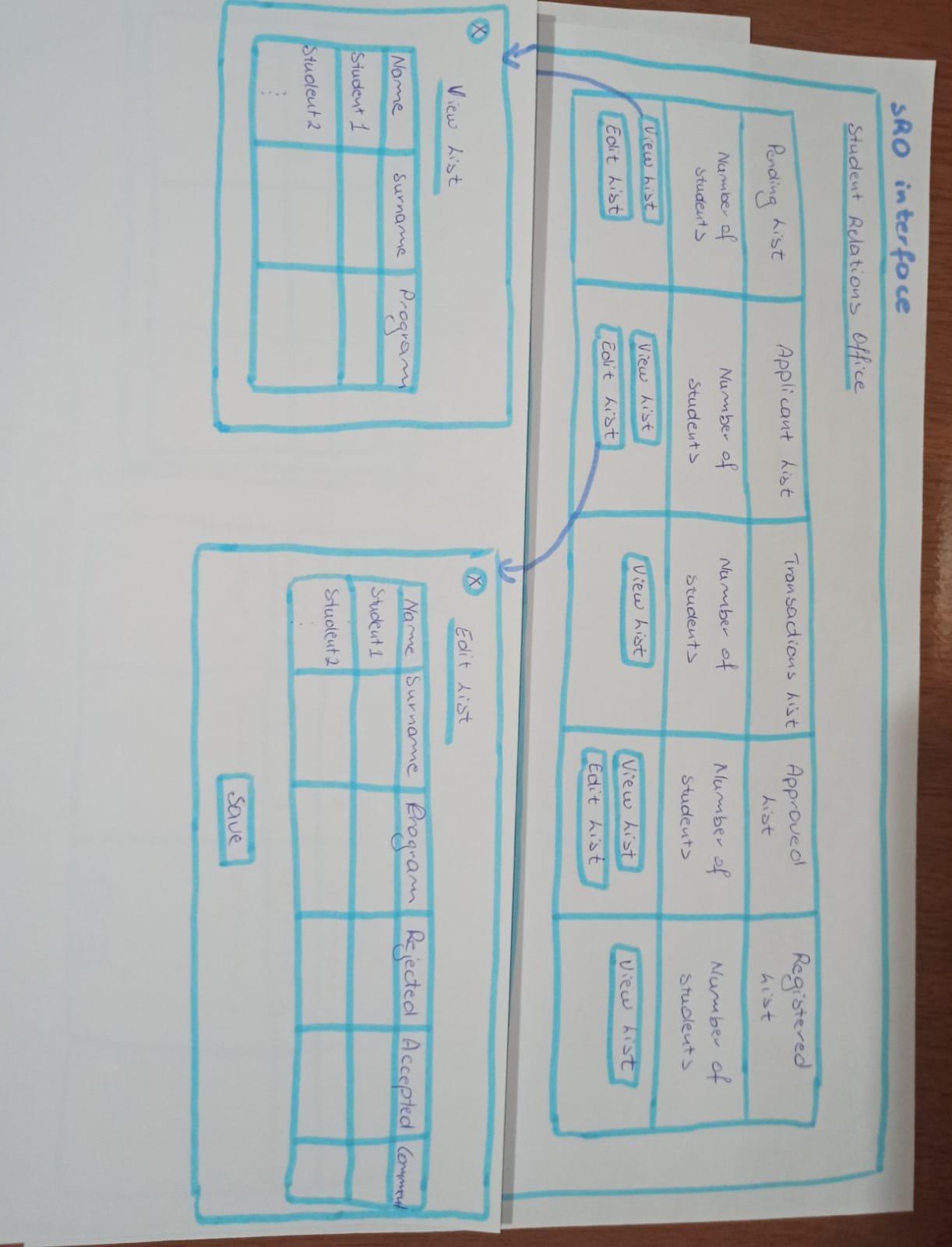
****

**5. STUDENT APPLICATION FORM**

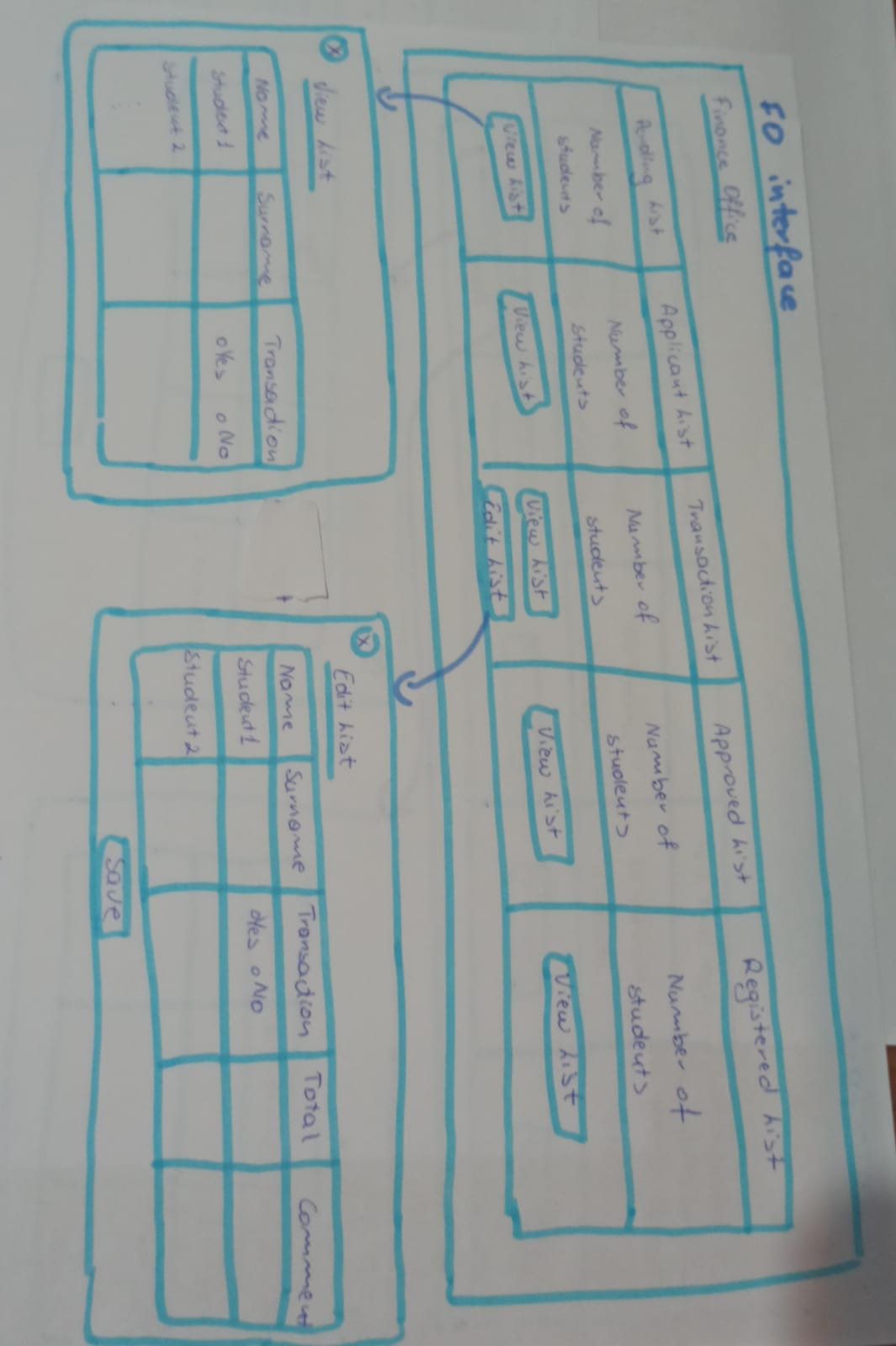
****

****

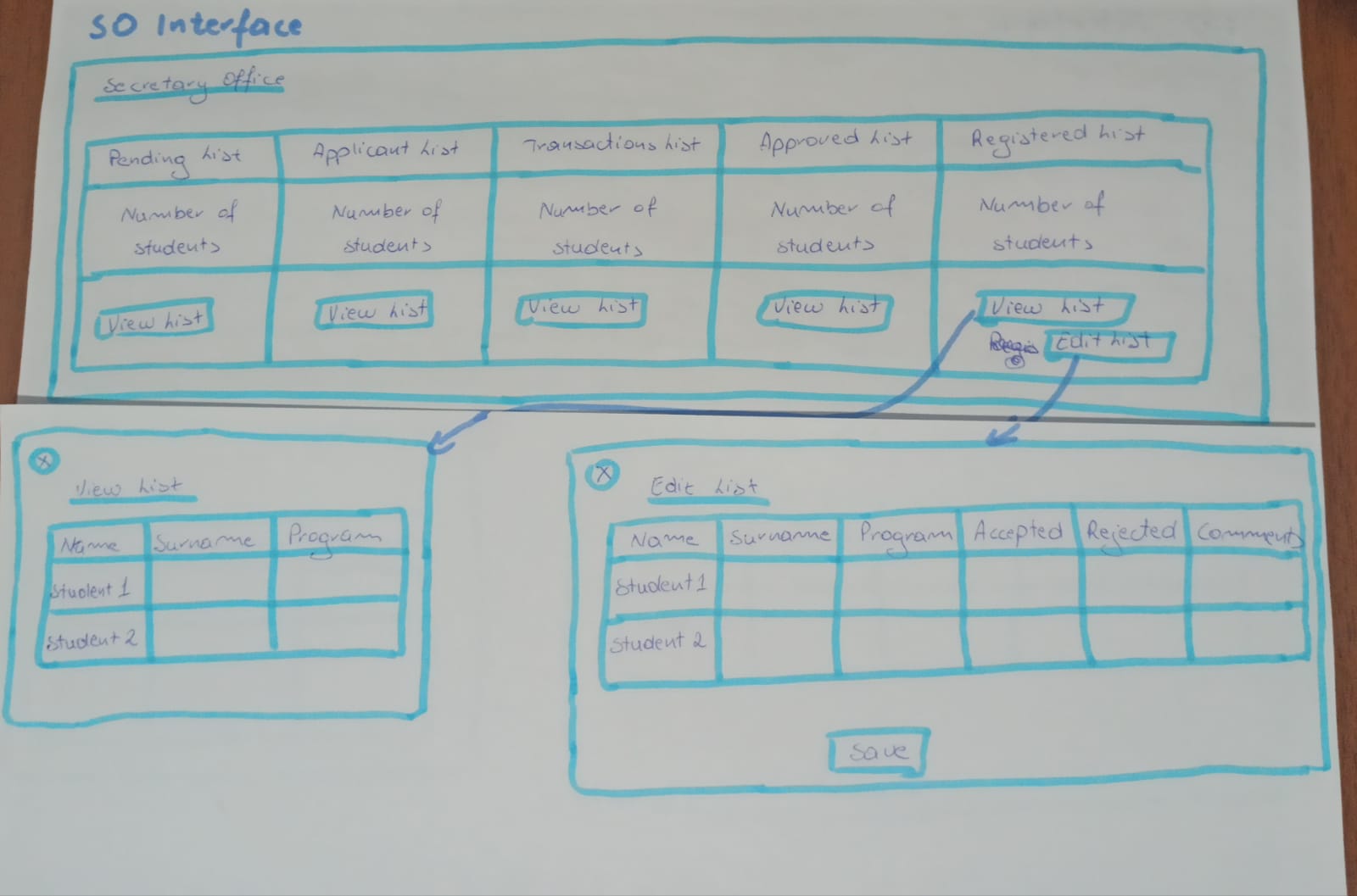
1. **STUDENT RELATIONS OFFICE INTERFACE**

****

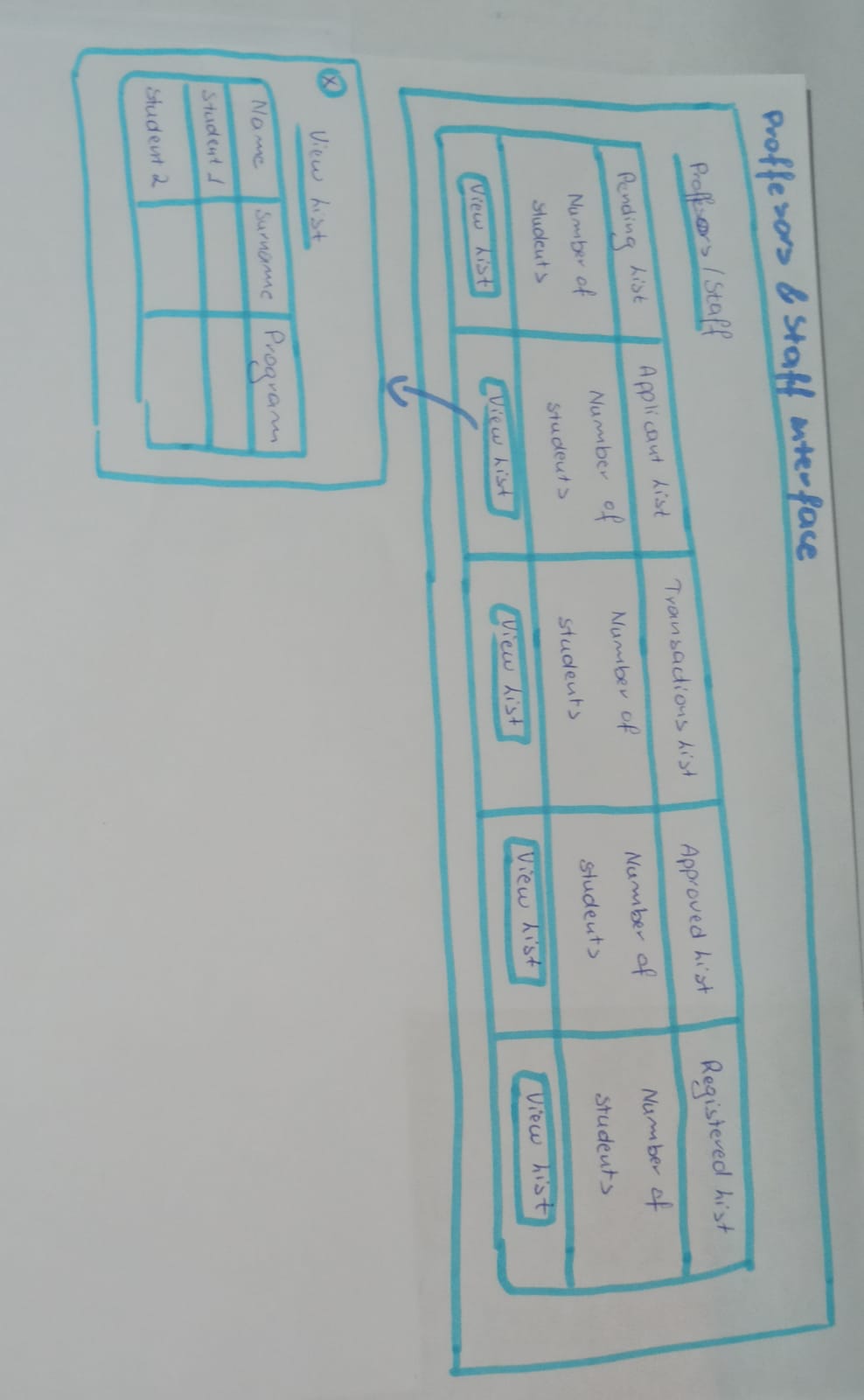
1. **FINANCE OFFICE INTERFACE**

****

1. **SECRETARY OFFICE INTERFACE**

****

1. **PROFFESORS & STAFF INTERFACE**

****