

# **Object Oriented Analysis & Design**

## **SCHOLARSHIP SELECTION SYSTEM**

**Sumaiya Fathima K.S**

**2018103607**

**S.K.Risheek Rakshit**

**2018103580**

## **Introduction:**

Scholarship selection process is currently a manual process, which makes it tedious for both the students and the officers related with this process. As details get passed through various sources to the students, a lot of people are involved in this process apart from the officers and the students like the department officials, class representatives etc. Also handing out the hard copy of the documents within the duration can get affected due to various factors and also once the students are selected all the updates are to be performed manually. All these processes can be reduced if we make the process computerized and by creating a one to one kind of atmosphere between the students and the officials involved. This project aims at making this process smooth with the help of computational solutions.

## **Problem Statement and Description:**

Scholarship selection system is a direct platform for students to apply for the scholarships based on the criteria provided by the organization. It supports everything including Viewing Scholarship details, applying for the scholarships, checking the eligibility of the students, uploading necessary documents, notifying the students regarding the status of application and the option to renew the scholarship if the organization permits. Each and every stage of verification is done by the system and the student gets updates regarding the status at each step.

The organization gets to add/edit scholarship details and criterias. He can also view the details of the student who has applied for the scholarship and he can get the final list of selected students after shortlisting. The added scholarship becomes visible to the students once it gets the approval from the admin. The admin is responsible for verifying the documents and shortlisting students. The admin also takes care of the updates and maintenance of the system.

## Actors and Goals:

### Finding Actors and goals:

#### *Primary Actors:*

- 1) Student
- 2) Organization

#### *Secondary Actors:*

- 1) Administrator

Actors	Goals
Student	View and apply for scholarships.  Renew ongoing scholarship.  Manage his/her own account .
Organization	Provide Scholarships along with some criteria and finalizes the candidates for the scholarships.
Admin	Has access to all the accounts of both students and organizations and is responsible for maintenance and updating the entire selection system.

## Use Cases

- 1) View Scholarship
- 2) Apply Scholarship
- 3) List Scholarship and criteria
- 4) Renew Scholarship
- 5) Validation
- 6) Notifying Applicants
- 7) Shortlisting Applicants
- 8) Manage and View Account
- 9) Maintenance and Updates
- 10) Sending Shortlist

### **1) View Scholarship:**

**Description:** Students can view the list of available scholarships along with its eligibility criteria.

**Actors Involved:**

- Student
- Administrator

**Preconditions:** Students should have an account and must be logged in.

**Flow:** The admin makes the list of scholarships visible to the students. The students can view the scholarship details, along with its criteria.

### **2) Apply Scholarship:**

**Description:** If a student wants to apply for a scholarship, he/she can fill the form available and attach the necessary documents.

**Actors Involved:**

- Student

**Preconditions:** Students must have an account and must have some scholarship to select from.

**Flow:** The student must fill the form provided in this section and must also upload the necessary documents.

**Success Scenario:** On success, the student is entitled for validation process.

### **3) List Scholarship and criteria**

**Description:** The organization provides the details of the scholarships along with its eligibility criteria to the administrator.

**Actors Involved:**

- Organization
- Administrator

**Preconditions:** The organization must have an account in the portal and must be verified by the administrator.

**Flow:** If the organization wants to provide a scholarship, the organization contacts the admin regarding the scholarship details along with deadline.

**Success Scenario:** On success, admin acknowledges the organization and makes the scholarship visible for students.

**Exception Scenario:** If the details provided are not sufficient, the organization is asked to submit the details again.

### **4) Renew Scholarship**

**Description:** If there is an option of renewing a scholarship, the student can apply for renewal and then the validation process continues.

**Actors Involved:**

- Student

**Precondition:** The student must have already been benefited by the scholarship and its tenure must be about to end.

**Success Scenario:** On success, the student is directed to validation process.

**Exception Scenario:** If the student reaches the deadline for renewing, then he can't proceed further and is basically directed towards applying a new scholarship.

## **5) Validation**

**Description:** When a student applies for a scholarship, the details filled by the applicant are used to check whether the details filled are authentic, applicant matches the criteria given and finally the documents uploaded are verified.

**Actors Involved:**

- Administrator(only for document verification)

**Precondition:** Student must have successfully applied for the scholarship.

**Flow:** Initially, the details of the application is checked with the student's profile. The details are checked with the company's criteria. In the final stage of this process, the document is verified by the admin.

**Scenario:** After every stage, the students must be notified about the progress, and only on passing all the three stages, the student is nominated for shortlisting process.

## **6) Notifying Applicants**

**Description:** The applicants are notified about the status of success and failure at each stage of the process.

**Flow:** If any use case triggers it the system, sends the status of the application in the ongoing process.

## **7) Shortlisting Applicants**

**Description:** The system selects few students from the eligible candidates based on some criteria given by the organization.

**Actors Involved:**

- Administrator(only for passing on the criteria)

**Preconditions:** The students must have successfully completed all the stages of verification.

**Flow:** Admin passes on the criteria provide by the organization. The system shortlists the students based on this criteria.

**Success Scenario:** On success,the applicant is notified and his details are added to the final list.

## **8) Manage and View Account**

**Description:** The student and the organization can view or edit their details. Student can check the application status.

**Actors Involved:**

- Student
- Organization
- Administrator

**Precondition:** Both the student and the organization must have an account in the portal.

## **9) Maintenance and Updates**

**Description:** The administrator has overall control over the accounts of students and organization and can update the details in each account if changes occur.

**Actors Involved:**

- Administrator

**Flow:** If a student gets selected for the scholarship, admin can modify the applicant's account like changing the attributes regarding scholarships rewarded , fee deduction etc.

## **10) Sending Shortlist**

**Description:** The list of shortlist candidates is informed to the organization.

**Actors Involved:**

- Administrator
- Organization

**Precondition:** Shortlisting procedure must be completed.

**Flow:** After the shortlisting process is over, a list of selected candidates is sent to the admin and only after the approval from the admin, the final shortlist is sent to the organization.



```

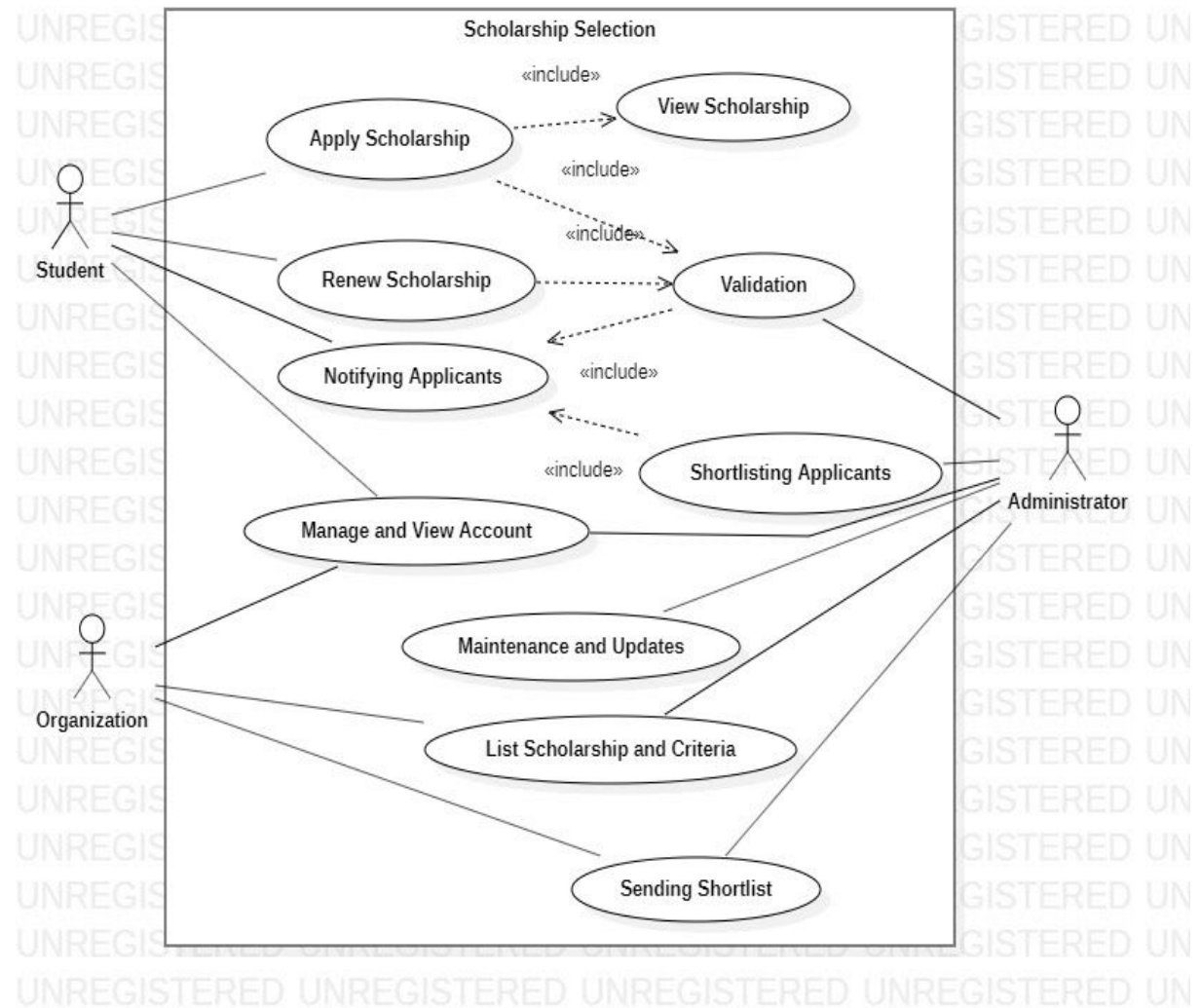
    usecaseDiagram
        actor Student
        actor Organization
        actor Administrator

        usecase UC1[Apply Scholarship]
        usecase UC2[Renew Scholarship]
        usecase UC3[Notifying Applicants]
        usecase UC4[Manage and View Account]
        usecase UC5[Maintenance and Updates]
        usecase UC6[List Scholarship and Criteria]
        usecase UC7[Sending Shortlist]
        usecase UC8[View Scholarship]
        usecase UC9[Validation]
        usecase UC10[Shortlisting Applicants]

        Student -- UC1
        Student -- UC2
        Student -- UC3
        Student -- UC4
        Organization -- UC4
        Organization -- UC5
        Organization -- UC6
        Organization -- UC7
        Administrator -- UC8
        Administrator -- UC9
        Administrator -- UC10

        UC1 -.-> UC8 : «include»
        UC1 -.-> UC9 : «include»
        UC2 -.-> UC9 : «include»
        UC3 -.-> UC10 : «include»
        UC4 -.-> UC10 : «include»
        UC9 -.-> UC10 : «include»
    
```

The diagram illustrates the functional requirements of a Scholarship Selection System. It features three actors: Student, Organization, and Administrator. The Student actor is associated with the 'Apply Scholarship', 'Renew Scholarship', 'Notifying Applicants', and 'Manage and View Account' use cases. The Organization actor is associated with 'Manage and View Account', 'Maintenance and Updates', 'List Scholarship and Criteria', and 'Sending Shortlist'. The Administrator actor is associated with 'View Scholarship', 'Validation', and 'Shortlisting Applicants'. There are several 'include' relationships: 'Apply Scholarship' includes 'View Scholarship' and 'Validation'; 'Renew Scholarship' includes 'Validation'; 'Notifying Applicants' includes 'Shortlisting Applicants'; 'Manage and View Account' includes 'Shortlisting Applicants'; and 'Validation' includes 'Shortlisting Applicants'.



# Domain Model

## **Domain Classes:**

1. Verifier
2. User
3. Organization
4. Student
5. Admin
6. Selected List
7. Scholarship
8. Student Details
9. Details&Criteria
10. Notification

### **Data Dictionary**

#### **Verifier:**

Verifier is a class used for verification purposes. In the first stage it verifies the details entered by the student with that of his details. In the second stage, it verifies whether the details given match the criteria for that particular scholarship.

#### **User:**

User is the person who has access to the portal and its contents. Every user has a unique-id with which the person can be identified.

#### **Organization:**

Organization is a user who offers scholarships to the students. The organization provides the details of the scholarship along with its criteria to the admin of the portal.

#### **Student:**

A Student uses the portal to view and apply for the scholarship. If a student wishes to apply for the scholarship, he/she has to fill the form associated with each scholarship and upload the necessary documents which are required for the scholarship.

#### **Admin:**

An admin is a user who has overall access to all the user accounts in the portal and is the one responsible for approving the scholarships, verifying the documents uploaded and to generate a list of shortlisted students. He is also responsible for updating the details and the overall maintenance of the portal.

**Selected List:**

This class contains the details of the students selected for the scholarship. The admin or organization can get the beneficiary details of the particular scholarship from this class.

**Scholarship:**

This class contains the scholarship ID and scholarship name provided by the scholarship and it also contains the acceptance status of the scholarship provided by the admin.

**Student details:**

This class consists of most of the details of the students like cgpa, DOB, annual income etc. which is used by the verifier class in the first stage of verification. The admin also updates this class if the student is selected for the scholarship.

**Details & Criteria:**

The details of the scholarship and the criteria provided by the organization to select the students is stored in this class and is used by the verifier to check whether the applicant matches the criteria.

**Notification:**

This class is used mainly for notifying the applicants regarding the progress of the application. It starts the process once it is invoked with the message to be sent by the classes during verification and shortlisting process.

**Association and Multiplicity**

Each student **has** a Student Detail(1-1)  
Each Student **applies** for any number of Scholarships(1-many)  
Verifier **verifies** Students Details(1-many)  
Verifier **verifies** with Details & Criteria(1-many)  
Verifier **sends status** to Notification(1-many)  
Every Scholarship **has** Details & Criteria(1-1)  
Admin **uses** Details and Criteria to shortlist the students(1-many)  
Admin **approves** the Scholarship(1-many)  
Admin **adds details** to Selected List(1-many)  
Admin **updates** Students Details(1-many)  
Admin **sends status** to Notification(1-many)  
Organization **adds** any number of Scholarships(1-many)  
Organization **gets** details from Selected List(1-many)

### **Attributes:**

#### **User:**

- ❖ Unique\_id

#### **Student:**

- ❖ Name
- ❖ Rollno

#### **Organization:**

- ❖ Cid
- ❖ Officer name
- ❖ Cname
- ❖ Address
- ❖ Phoneno
- ❖ Email

#### **Admin:**

- ❖ Name

#### **Notification:**

- ❖ Status
- ❖ Msg

#### **Verifier:**

- ❖ Verstat1
- ❖ Verstat2
- ❖ Verstat3

#### **Student Details:**

- ❖ CGPA
- ❖ DOB

- ❖ Gender
- ❖ Religion
- ❖ Nationality
- ❖ Caste
- ❖ Phoneno
- ❖ Email
- ❖ Address
- ❖ Annual income
- ❖ Fee
- ❖ Quota
- ❖ Backlogs
- ❖ HSC marks
- ❖ SSLC marks

**Selected List:**

- ❖ Studname
- ❖ Studroll
- ❖ Schname
- ❖ Schid
- ❖ Benefitamt

**Details & Criteria:**

- ❖ Year
- ❖ Percentage
- ❖ CGPA
- ❖ Annualincome
- ❖ Category
- ❖ Finalcriteria
- ❖ Amt
- ❖ Validity
- ❖ Backlogs
- ❖ Marks

**Classes refined using Inheritance:**

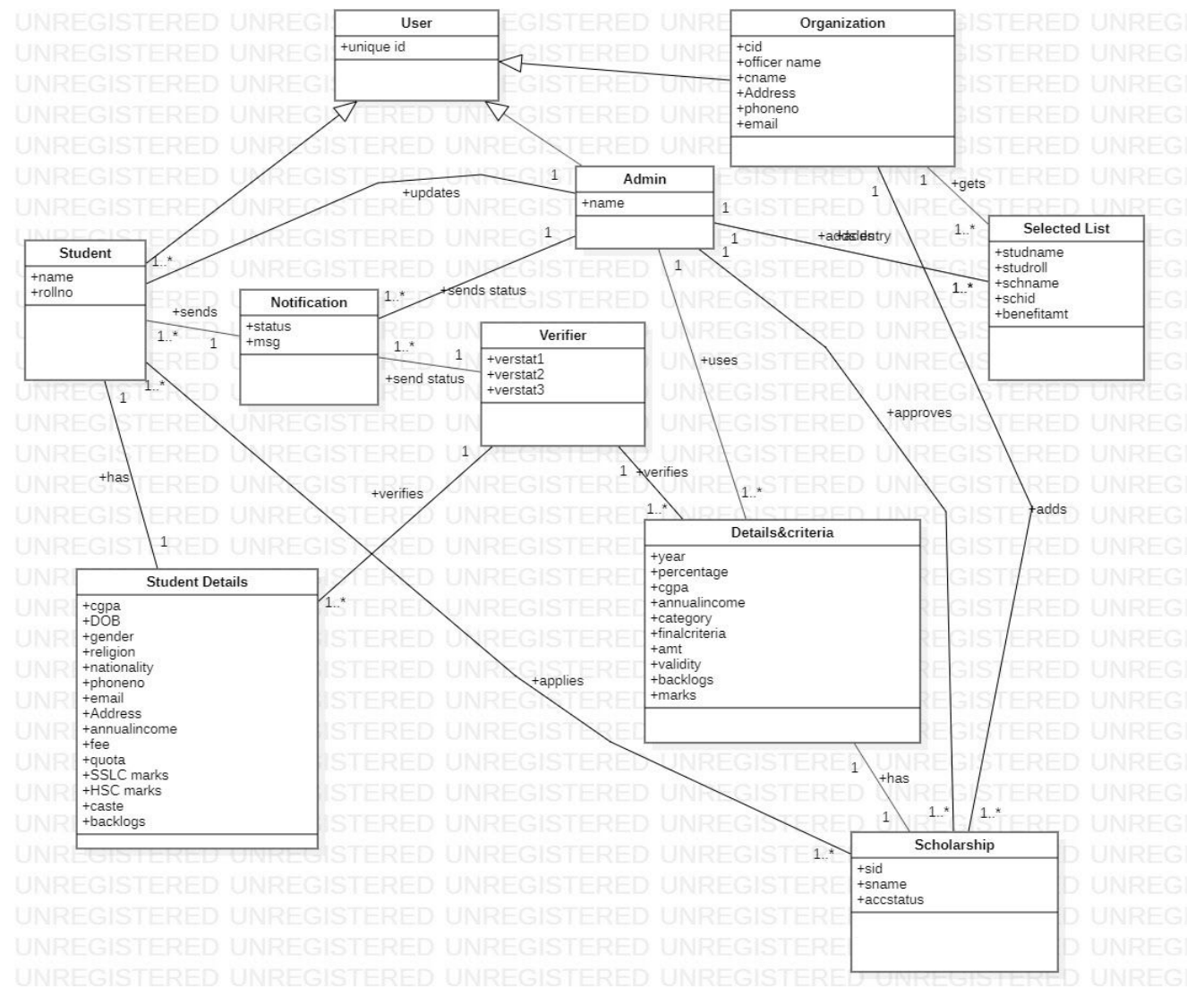
**Super Class:**

- User

- Student
- Admin
- Organization

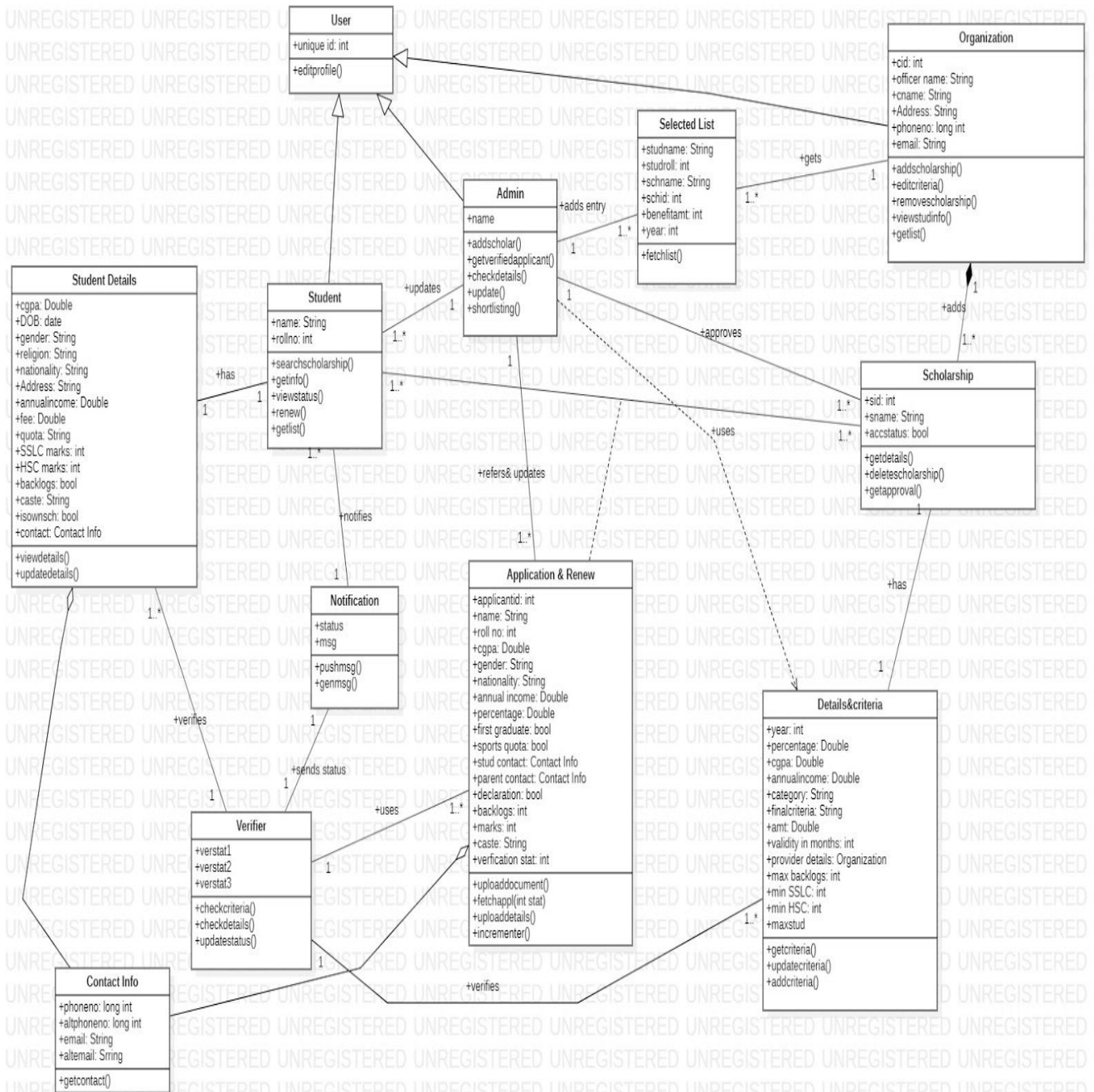
- Student
- Admin
- Organization

**Domain Model:**



# Class Model

## Version-1:



In the above model, the class contact info forms a part-whole relationship with the class **Student Details** as well as **Application&Renew**

In order to overcome this we have refined the class model so that it is appropriate.

## **Version-2:**

### **Classes**

1. Verifier
2. User
3. Organization
4. Student
5. Admin
6. Selected List
7. Scholarship
8. Student Details
9. Details&Criteria
10. Notification
11. Contact Info
12. Application & Renew

### **Classes refined using Inheritance:**

#### **1) Super Class:**

- User



### **Sub Class:**

- Student
- Admin
- Organization

### **Aggregation:**

1) *Contact info* - it forms a part whole relationship with Student Details.

### **Composition:**

*Scholarship* - a scholarship is supported by Organization. A particular scholarship must belong to a particular Organization.

### **Association Class:**

*Application&Renew*- is a Association Class between Student and Scholarship.

### **Dependency:**

Admin is dependent on Details&Criteria while executing the function shortlisting since it refers the final criteria attribute from the class Details&Criteria.

### **Description Classes:**

- 1) Student Details is a description class of Student.
- 2) Details & Criteria is a description class of Scholarship.

## Data Dictionary

### **Verifier:**

Verifier is a class used for verification purposes. In the first stage it verifies the details entered by the student with that of his details. In the second stage, it verifies whether the details given match the criteria for that particular scholarship.

### **User:**

User is the person who has access to the portal and its contents. Every user has a unique-id with which the person can be identified.

### **Organization:**

Organization is a user who offers scholarships to the students. The organization provides the details of the scholarship along with its criteria to the admin of the portal.

### **Student:**

A Student uses the portal to view and apply for the scholarship. If a student wishes to apply for the scholarship, he/she has to fill the form associated with each scholarship and upload the necessary documents which are required for the scholarship.

### **Admin:**

An admin is a user who has overall access to all the user accounts in the portal and is the one responsible for approving the scholarships, verify the documents uploaded and to generate a list of shortlisted students. He is also responsible for updating the details and the overall maintenance of the portal.

### **Selected List:**

This class contains the details of the students selected for the scholarship. The admin or organization can get the beneficiary details of the particular scholarship from this class.

### **Scholarship:**

This class contains the scholarship ID and scholarship name provided by the scholarship and it also contains the acceptance status of the scholarship provided by the admin.

### **Student details:**

This class consists of most of the details of the students like cgpa, DOB, annual income etc. which is used by the verifier class in the first stage of verification. The admin also updates this class if the student is selected for the scholarship.

### **Details & Criteria:**

The details of the scholarship and the criteria provided by the organization to select the students is stored in this class and is used by the verifier to check whether the applicant matches the criteria.

**Notification:**

This class is used mainly for notifying the applicants regarding the progress of the application. It starts the process once it is invoked with the message to be sent by the classes during verification and shortlisting process.

**Application & Renew:**

This class consists of all the details entered in the form by the candidate at the time of applying for the scholarship. It also consists of all the documents uploaded by the candidate.

**Contact Info:**

As the name suggests this class consists of all contact details for each student on the portal.

**Association and Multiplicity**

Each student **has** a Student Detail(1-1)  
Student **applies** for any number of Scholarships(many-many)  
Verifier **verifies** Students Details(1-many)  
Verifier **verifies** with Details & Criteria(1-many)  
Verifier **send** status to Notification(1-many)  
Verifier **uses** Application & Renew(1-many)  
Every Scholarship **has** Details & Criteria(1-1)  
Admin **approves** the Scholarship(1-many)  
Admin **adds** details to Selected List(1-many)  
Admin **sends** status to Notification(1-many)  
Organization **adds** any number of Scholarships(1-many)  
Organization **gets** details from Selected List(1-many)  
Admin **verifies&updates** attributes of Application & Renew(1-many)

**Class Attributes and Methods:**

Class	Attributes	Methods
User	<ul style="list-style-type: none"><li>Unique_id</li></ul>	<ul style="list-style-type: none"><li>❖ editprofile</li></ul>
Student	<ul style="list-style-type: none"><li>Name</li><li>Rollno</li></ul>	<ul style="list-style-type: none"><li>❖ Searchscholarship</li><li>❖ Uploaddetails</li></ul>

		<ul style="list-style-type: none"> <li>❖ Getinfo</li> <li>❖ Viewstatus</li> <li>❖ Renew</li> <li>❖ Viewscholarship</li> </ul>
Organization	<ul style="list-style-type: none"> <li>● Cid</li> <li>● Officer name</li> <li>● Cname</li> <li>● Address</li> <li>● Phoneno</li> <li>● Email</li> </ul>	<ul style="list-style-type: none"> <li>❖ Addscholarship</li> <li>❖ Editcriteria</li> <li>❖ Removescholarship</li> <li>❖ Viewstudinfo</li> </ul>
Admin	<ul style="list-style-type: none"> <li>● Name</li> </ul>	<ul style="list-style-type: none"> <li>❖ Addscholar</li> <li>❖ Removescholar</li> <li>❖ Getverifiedapplicant</li> <li>❖ Shortlist</li> <li>❖ Sendshort</li> <li>❖ Checkdetails</li> <li>❖ Update</li> </ul>
Notification	<ul style="list-style-type: none"> <li>● Status</li> <li>● Msg</li> </ul>	<ul style="list-style-type: none"> <li>❖ Pushmsg</li> <li>❖ Genmsg</li> </ul>
Verifier	<ul style="list-style-type: none"> <li>● Verstat1</li> <li>● Verstat2</li> <li>● Verstat3</li> </ul>	<ul style="list-style-type: none"> <li>❖ Checkcriteria</li> <li>❖ Checkdetails</li> <li>❖ Updatestatus</li> </ul>
Student Details	<ul style="list-style-type: none"> <li>● CGPA</li> <li>● DOB</li> <li>● Gender</li> <li>● Religion</li> <li>● Nationality</li> <li>● Caste</li> <li>● Phoneno</li> <li>● Email</li> <li>● Address</li> <li>● Annual income</li> <li>● Fee</li> <li>● Quota</li> <li>● Backlogs</li> <li>● HSC marks</li> <li>● SSLC marks</li> </ul>	<ul style="list-style-type: none"> <li>❖ Viewdetails</li> <li>❖ updatedetails</li> </ul>
Selected List	<ul style="list-style-type: none"> <li>● Studname</li> <li>● Studroll</li> <li>● Schname</li> <li>● Schid</li> <li>● Benefitamt</li> </ul>	<ul style="list-style-type: none"> <li>❖ Fetchlist</li> </ul>

Details & Criteria	<ul style="list-style-type: none"> <li>• Year</li> <li>• Percentage</li> <li>• CGPA</li> <li>• Annualincome</li> <li>• Category</li> <li>• Finalcriteria</li> <li>• Amt</li> <li>• Validity</li> <li>• Backlogs</li> <li>• Marks</li> </ul>	<ul style="list-style-type: none"> <li>❖ Getcriteria</li> <li>❖ UpdateCriteria</li> </ul>
Contact Info	<ul style="list-style-type: none"> <li>• Altphoneno</li> <li>• Email</li> <li>• Altemail</li> <li>• Phoneno</li> </ul>	<ul style="list-style-type: none"> <li>❖ Getcontact</li> </ul>
Application & Renew	<ul style="list-style-type: none"> <li>• Applicantid</li> <li>• Name</li> <li>• Roll no</li> <li>• Cgpa</li> <li>• Gender</li> <li>• Nationality</li> <li>• Annual income</li> <li>• Percentage</li> <li>• First graduate</li> <li>• Sports quota</li> <li>• Stud contact</li> <li>• Parent contact</li> <li>• Declaration</li> <li>• Backlog count</li> <li>• Marks</li> <li>• Caste</li> <li>• Verification Stat</li> </ul>	<ul style="list-style-type: none"> <li>❖ Uploaddocument</li> <li>❖ Fetchapp</li> </ul>

### **Class Responsibility Collaboration:**

Application & Renew	
filldetails() uploaddocs()	Contactinfo

Student	
viewsch()	Scholarship Details&Criteria

Student	
renew()	Application&Renew Scholarship Details&Criteria

Verifier	
checkcriteria() checkdetails() updatestatus()	StudentDetails Details&Criteria Admin Application&Renew Contactinfo

Notification	
pushmsg() genmsg()	contactinfo

Admin	
sendshort() shortlisting()	Selected list Organization

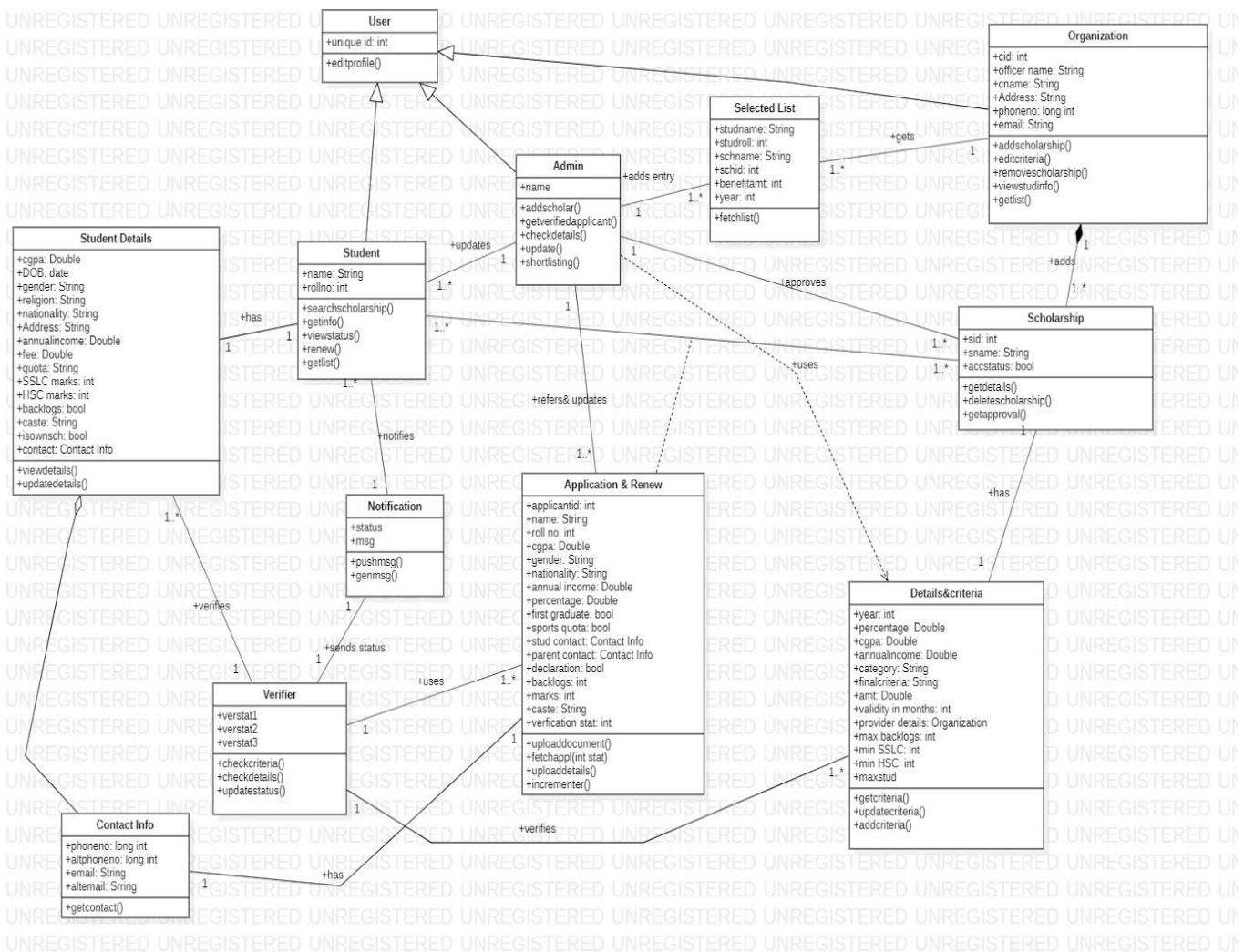
Admin	
update()	StudentDetails Organization Scholarship

## Details & Criteria

getcriteria()  
updatecriteria()

Scholarship  
Admin  
Organization

## Class Diagram:



## Sequence Modelling

Sequence Diagram	Classes Involved	Calling function	Sequence
<b>View and Update Details</b>	<ul style="list-style-type: none"> <li>• Student</li> <li>• StudentDetails</li> <li>• ContactInfo</li> </ul>	editprofile	<ol style="list-style-type: none"> <li>1. Editprofile</li> <li>2. Viewdetails</li> <li>3. Getcontact</li> <li>4. Contact</li> <li>5. Updatedetails</li> <li>6. SaveContact</li> </ol>
<b>Editing Criterias</b>	<ul style="list-style-type: none"> <li>• Organization</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	editcriteria	<ol style="list-style-type: none"> <li>1. Editcriteria</li> <li>2. Getdetails</li> <li>3. Getcriteria</li> <li>4. (return) Criterias</li> <li>5. (return)Scholarship</li> <li>6. Update criteria</li> </ol>
<b>Verification - 1&amp;2</b>	<ul style="list-style-type: none"> <li>• Verifier</li> <li>• Student</li> <li>• StudentDetails</li> <li>• ContactInfo</li> <li>• Application&amp;Renew</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	checkdetails	<ol style="list-style-type: none"> <li>1. Checkdetails</li> <li>2. Getinfo</li> <li>3. Viewdetails</li> <li>4. Getcontact</li> <li>5. (return)Contact</li> <li>6. (return)Details</li> <li>7. Student</li> <li>8. UpdateStatus</li> <li>9. Incrementer(value)</li> <li>opt[verstat = 1]</li> <li>10. Checkcriteria</li> <li>11. Getdetails</li> <li>12. Getcriteria</li> <li>13. (return)criteria</li> <li>14. (return)scholarship</li> <li>15. Updatestatus</li> <li>16. Incrementer</li> </ol>
<b>AddingScholarships</b>	<ul style="list-style-type: none"> <li>• Organization</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	Addscholarship	<ol style="list-style-type: none"> <li>1. Addscholarship</li> <li>2. Create(name,id)</li> <li>3. (create)create</li> <li>4. Addcriteria</li> </ol>

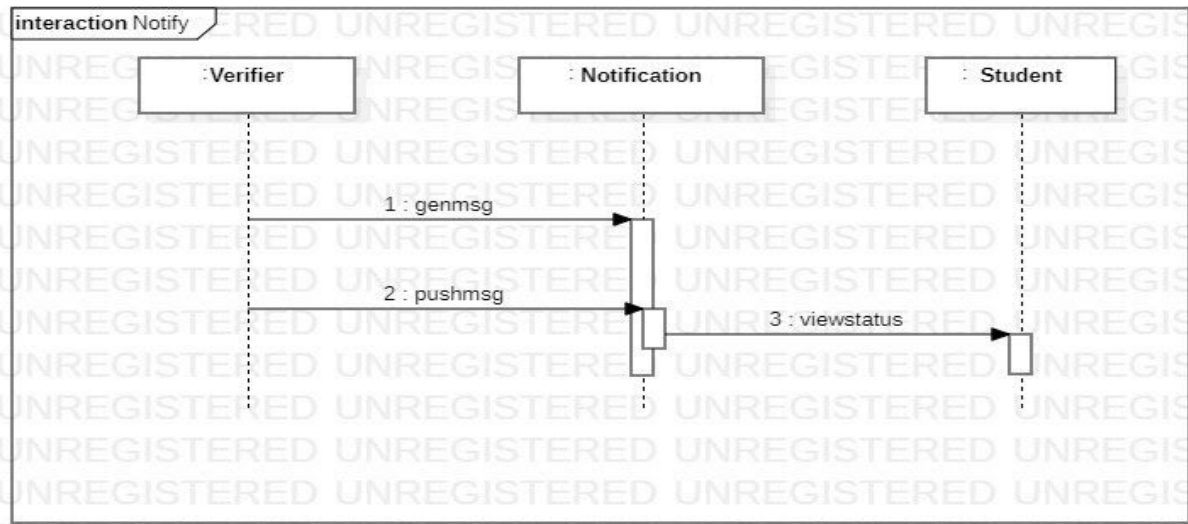


<b>UpdateDetails</b>	<ul style="list-style-type: none"> <li>• StudentDetails</li> <li>• Admin</li> </ul>	Updatedetails	1.Updatedetails 2.Admin
<b>Renewing Scholarship</b>	<ul style="list-style-type: none"> <li>• Student</li> <li>• Application&amp;Renew</li> </ul>	Renew	1.Renew 2.Fetchappl 3.(return)application 4.Upload details
<b>Shortlisting</b>	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Application</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	Getverifiedapplicants	1.Getverifiedapplications 2.Fetchappl(stat) 3.(return)application 4.Getdetails 5.Getcriteria(final) 6.(return)Final 7.(return)Final 8.Shortlisting 9.Create(name,rollno,schid,schname,year, amt)
<b>Verifying</b>	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Application&amp;Renew</li> <li>• Verifier</li> </ul>	Checkdetails	1.Checkdetails Application&Renew Verifier 2.Fetchappl(status) 3.(return)application 4.Updatestat 5.Updatestat
<b>Applying Scholarship</b>	<ul style="list-style-type: none"> <li>• Student</li> <li>• Application&amp;Renew</li> </ul>	Apply	1.Apply 2.(create)application 3.Uploaddetails 4.Upload documents
<b>Fetching List</b>	<ul style="list-style-type: none"> <li>• User</li> <li>• Organization</li> <li>• Student</li> <li>• Selected List</li> </ul>	User invokes getlist() of Student or Organization	[if user is Organization] 1.Getlist 2.Fetchlist 3.(return)list [if user is Student] 1.Getlist 2.Fetchlist 3.List
<b>Viewing Scholarship</b>	<ul style="list-style-type: none"> <li>• Student</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	Search scholarship	1.Search scholarship Loop(schid 1-n): 2.Getdetails Opt(accstatus==1) 3.Getcriteria 4.(return)criteria 5.(return)details
<b>Accept Scholarship</b>	<ul style="list-style-type: none"> <li>• Admin</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	Checkdetails	1.Checkdetails 2.Getdetails 3.Getcriteria 4.(return)details&Criteria 5.(return)Scholarship

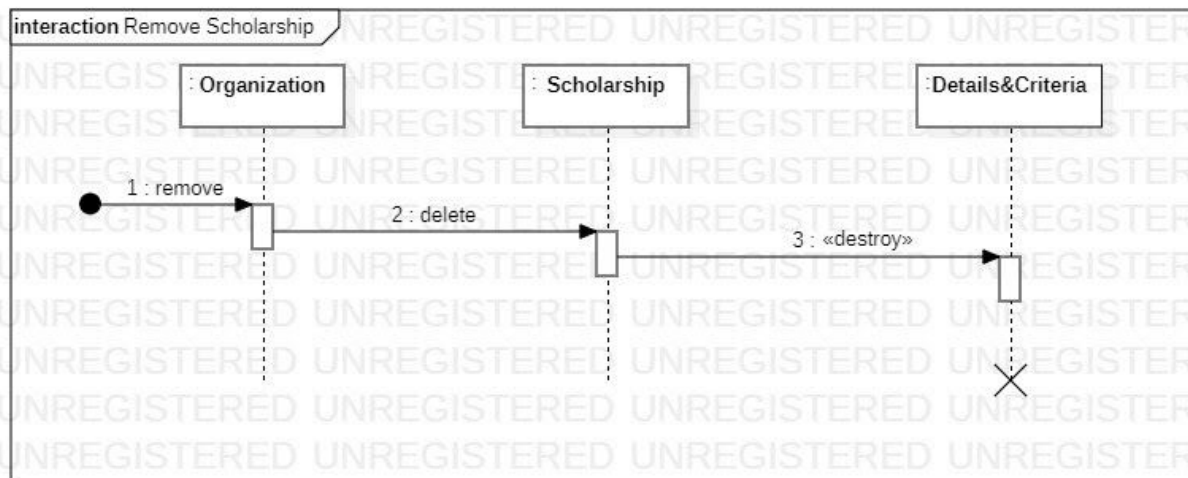
			6.Getapproval opt[accstatus = False] 7.Destroy
<b>View Student Info</b>	<ul style="list-style-type: none"> <li>• Organization</li> <li>• Selected List</li> <li>• Student</li> <li>• StudentDetails</li> <li>• ContactInfo</li> </ul>	Viewstud info	1.Viewstudinfo 2.Fetchlist 3.(return)list Loop[roll no in list]: 4.Getinfo 5.Viewdetails 6.Getcontact 7.(return)contact 8.(return)details 9.(return)info
<b>Remove Scholarship</b>	<ul style="list-style-type: none"> <li>• Organization</li> <li>• Scholarship</li> <li>• Details&amp;Criteria</li> </ul>	Remove	1.Remove 2.Delete 3.<<destroy>>
<b>Notify</b>	<ul style="list-style-type: none"> <li>• Verifier</li> <li>• Notification</li> <li>• Student</li> </ul>	Genmsg	1.Genmsg 2.Pushmsg 3.Viewstatus

## Sequence Diagram:

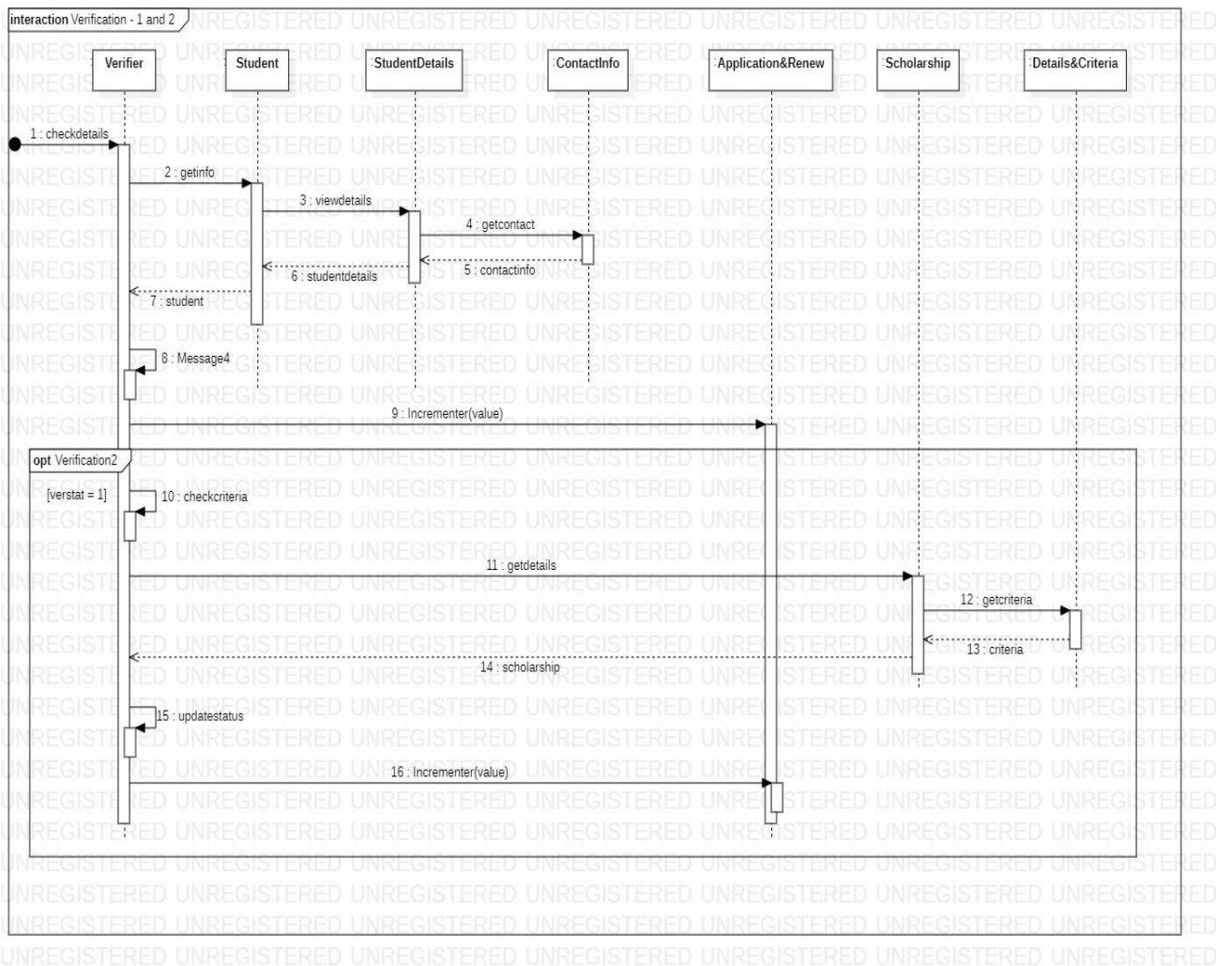
### Notify:



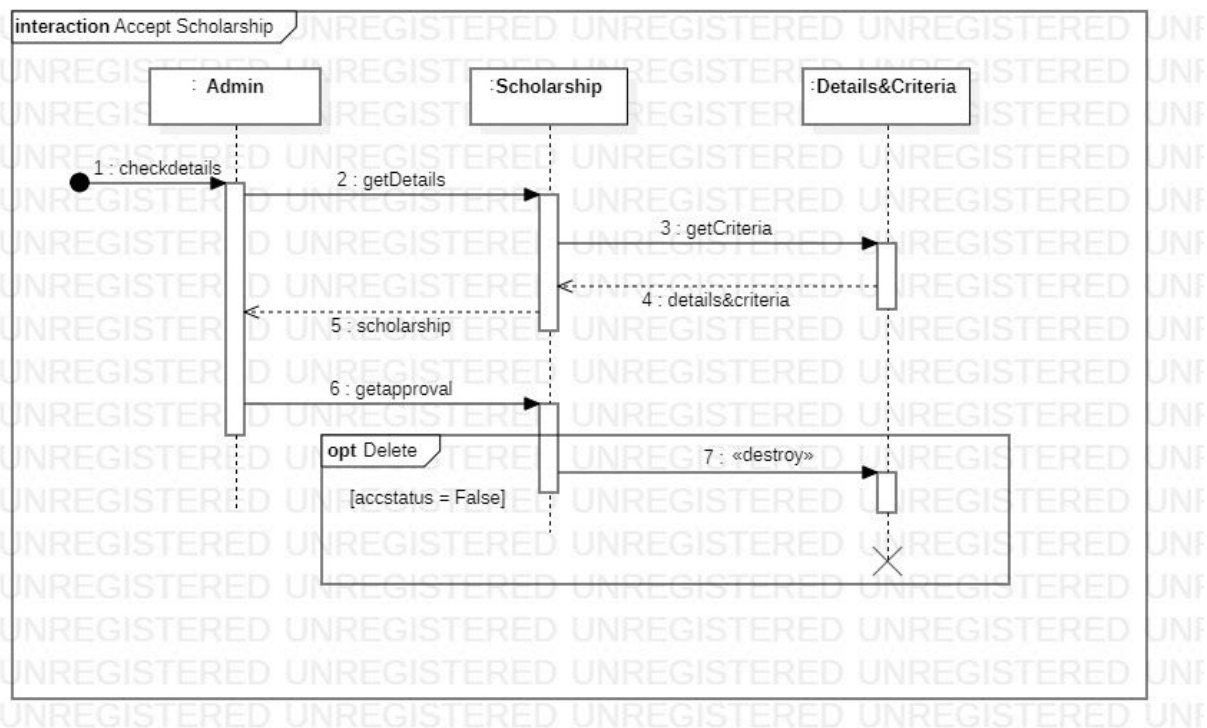
### Remove Scholarship:



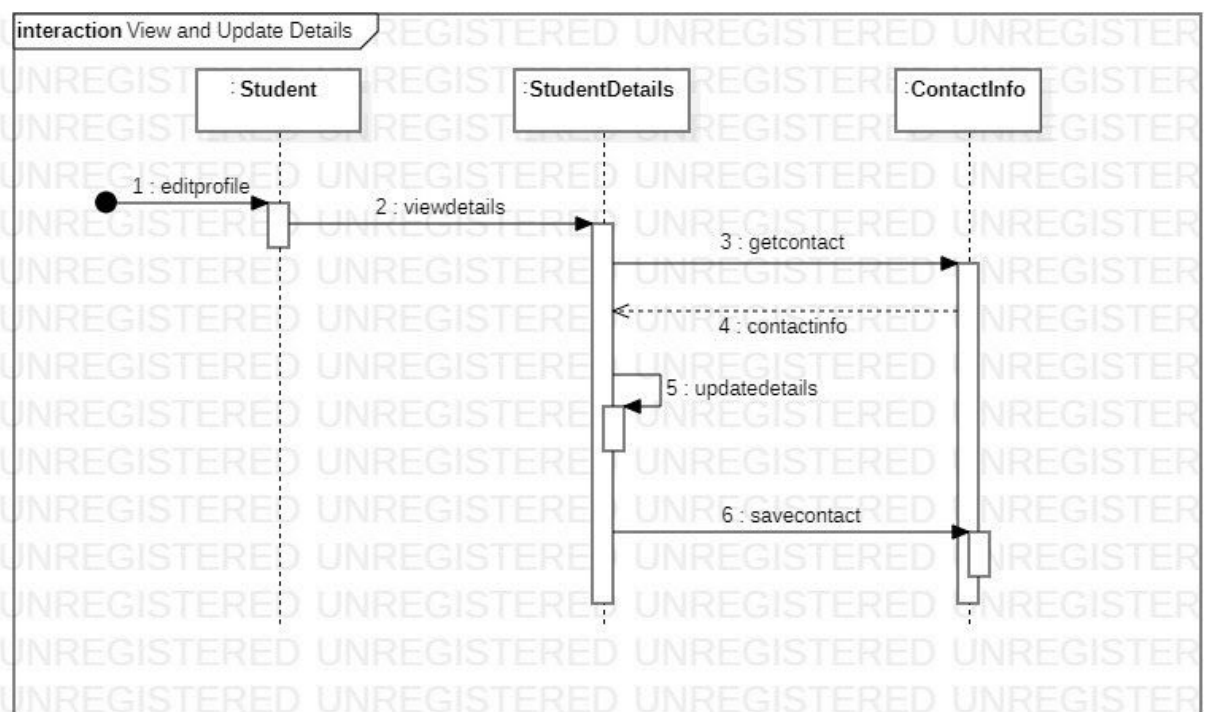
## Verification - 1&2:



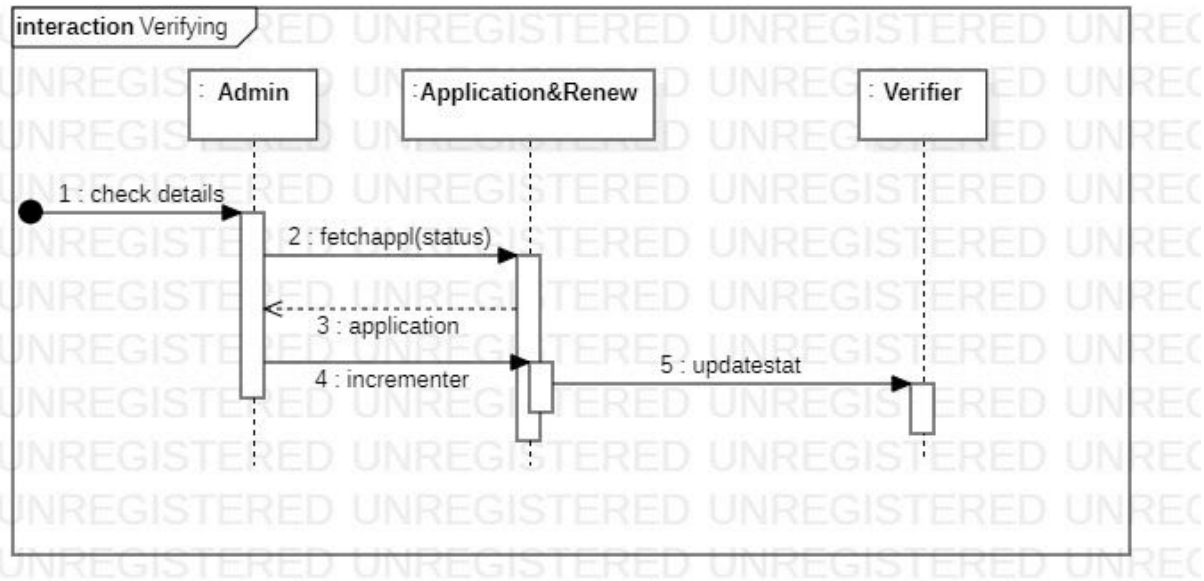
## Accept Scholarship:



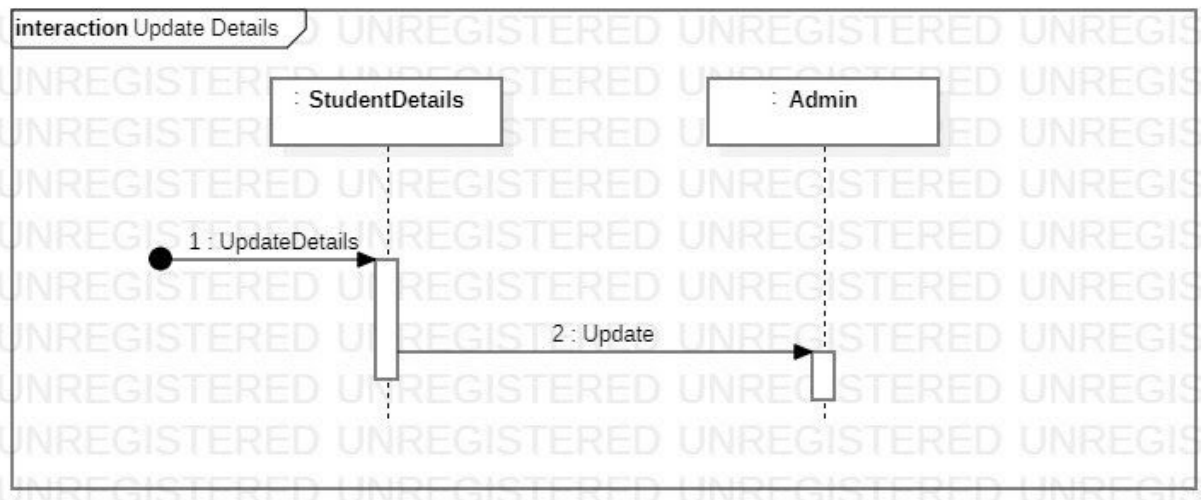
## View and Update Details:



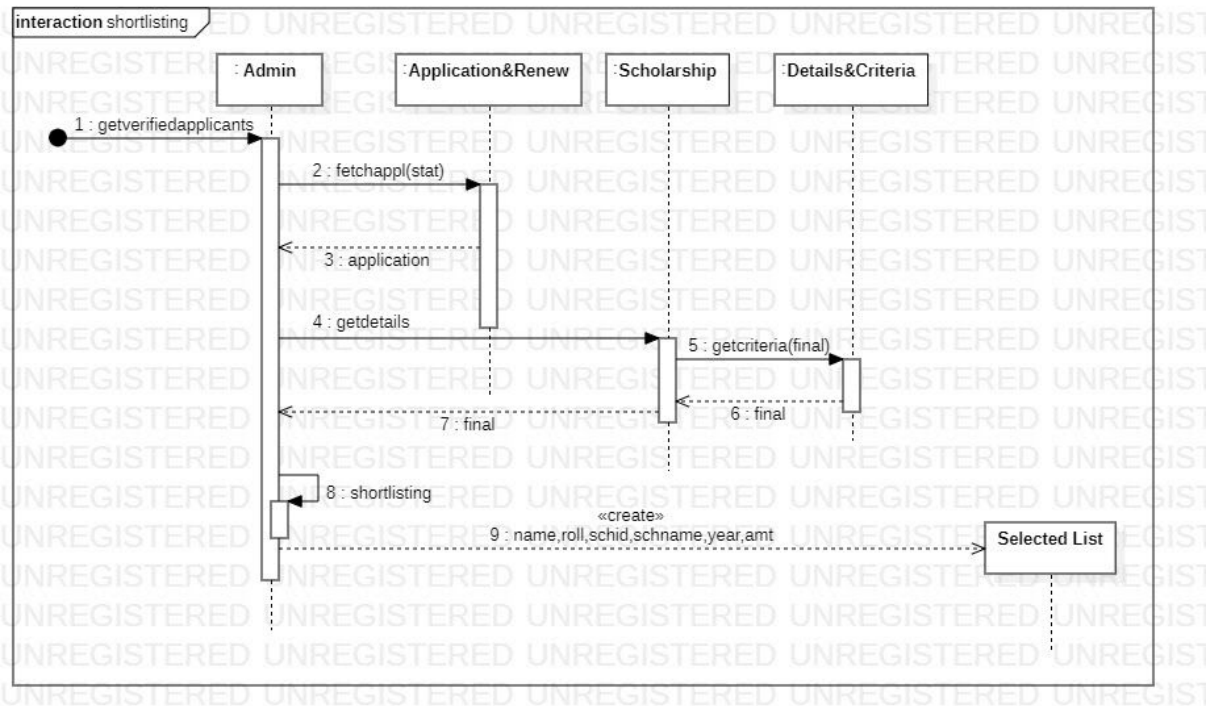
## Verifying



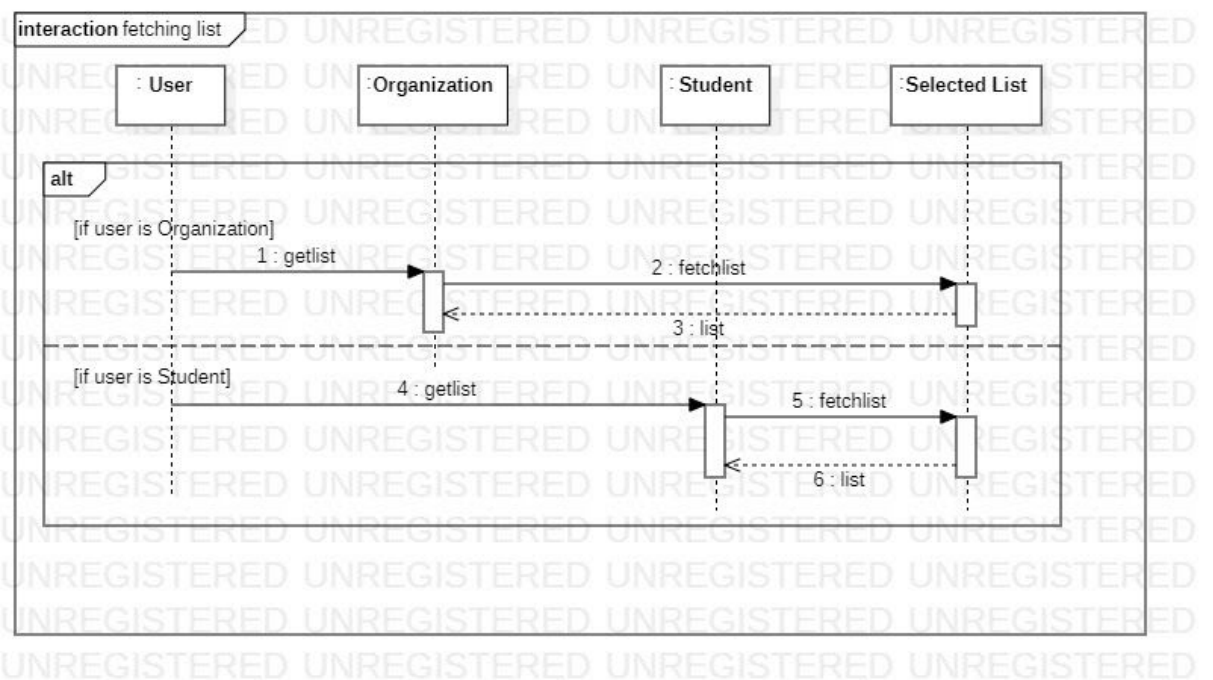
## UpdateDetails



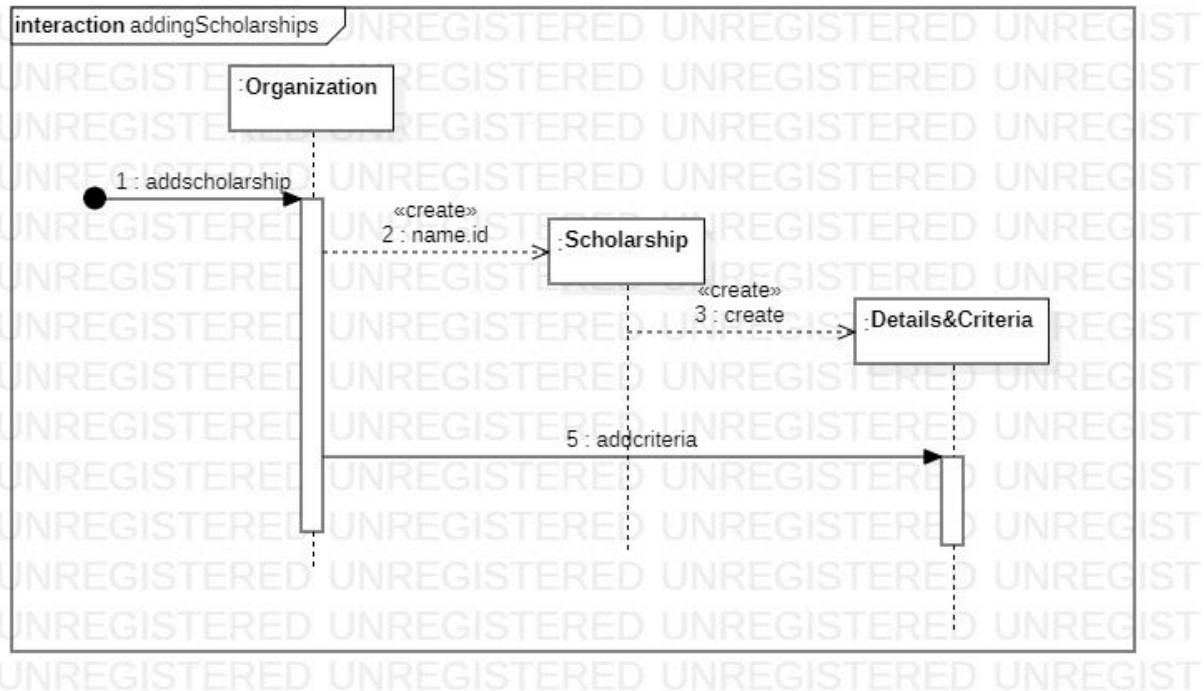
## Shortlisting:



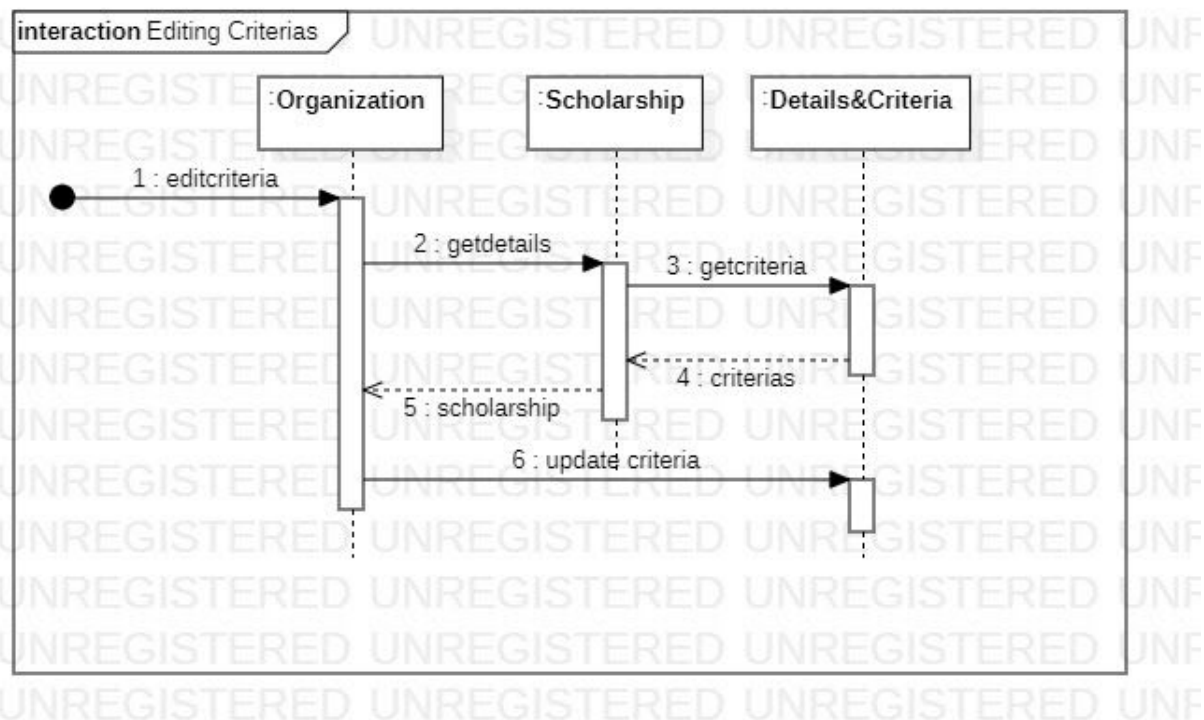
## Fetching list



## Adding Scholarships

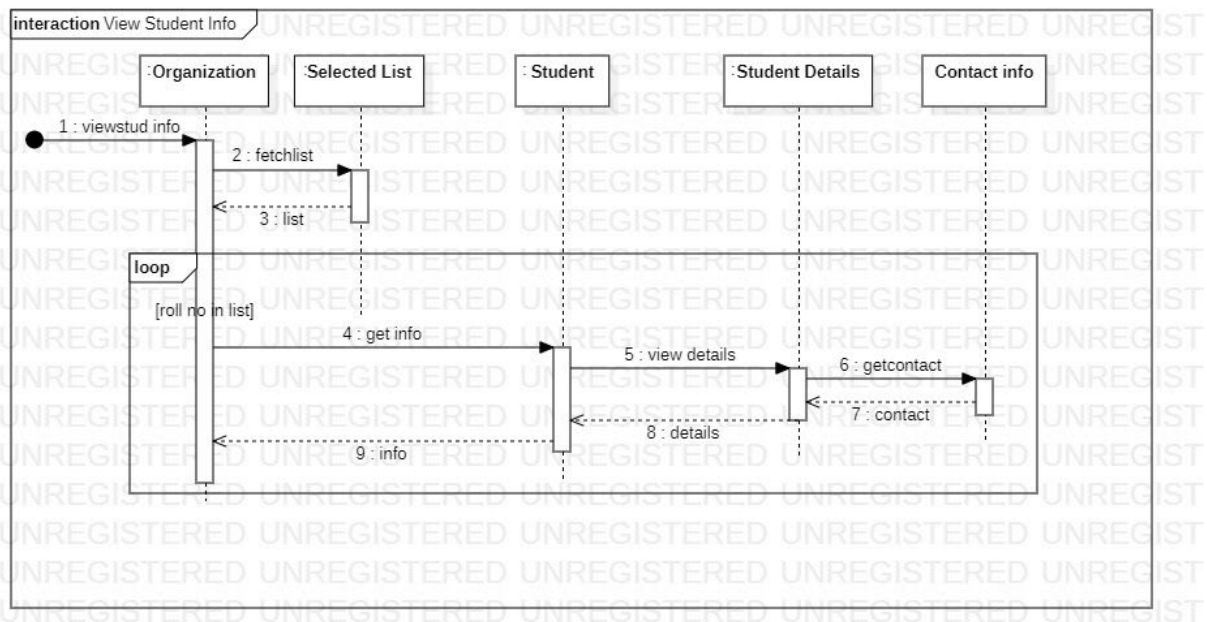


## Editing Criterias

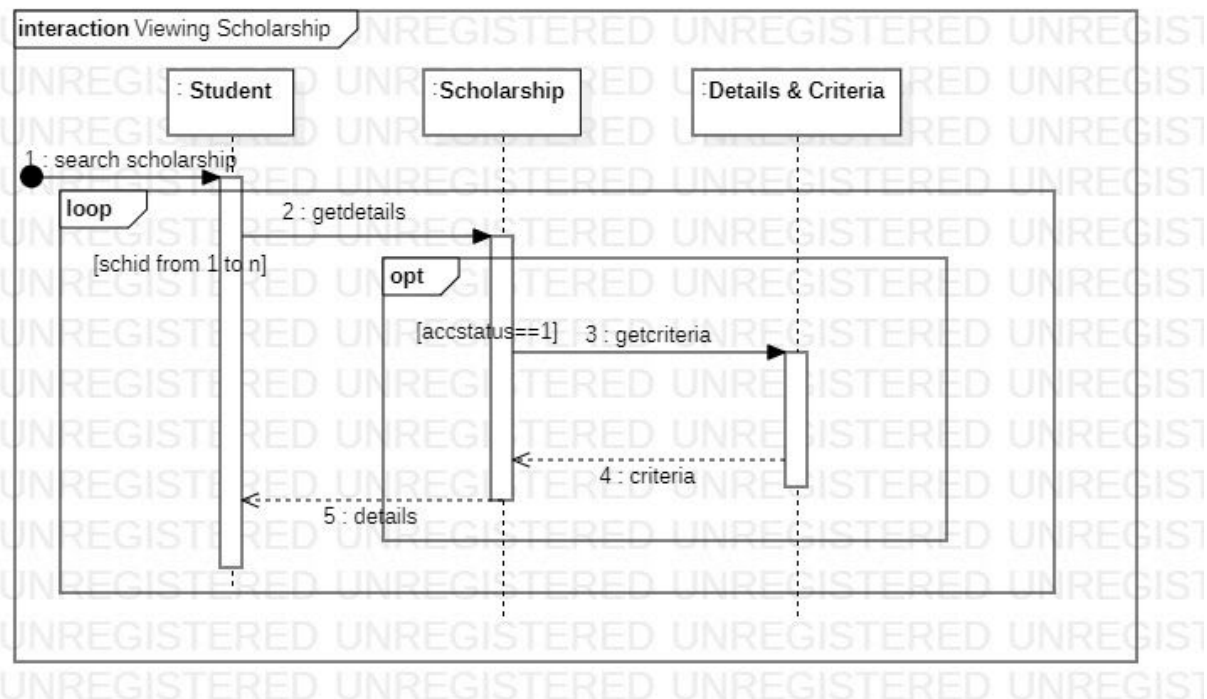




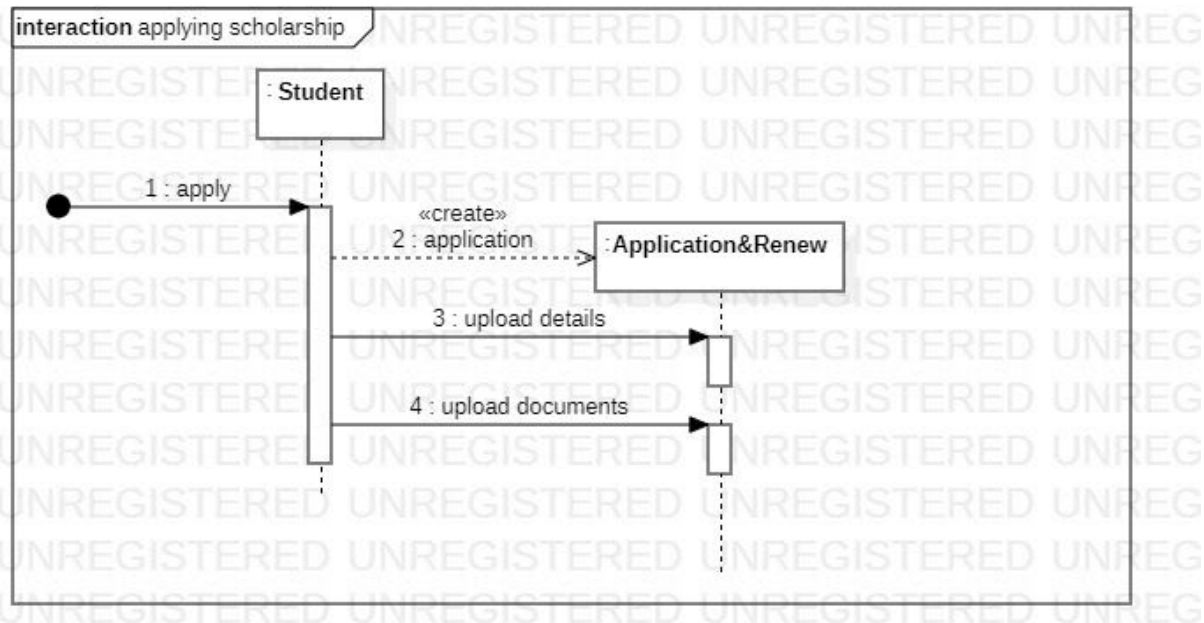
## View Student Info



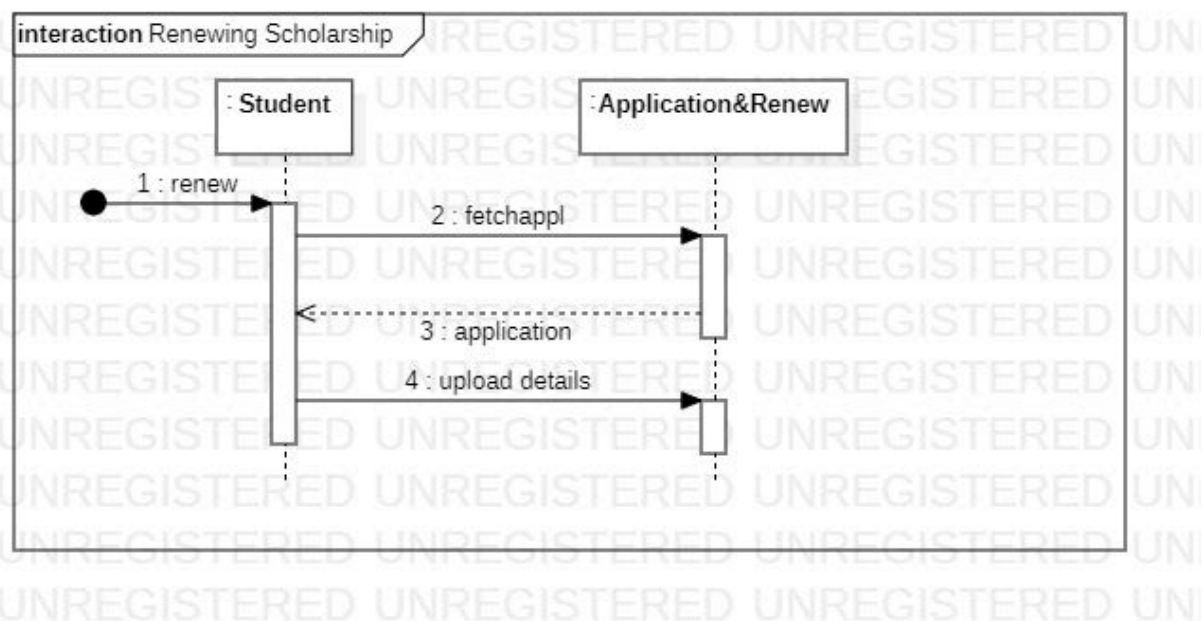
## Viewing Scholarship



## Apply Scholarship



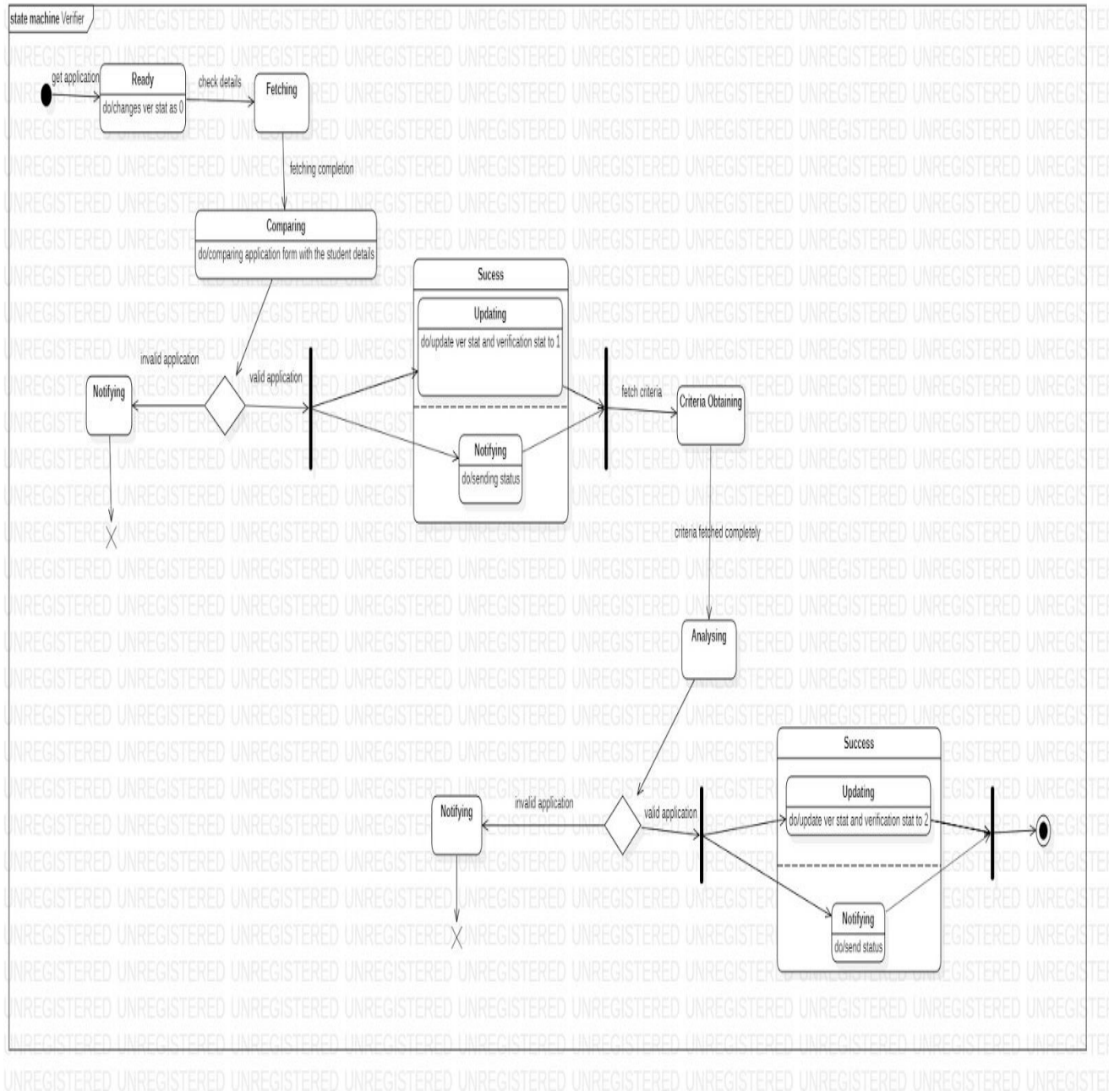
## Renewing Scholarship



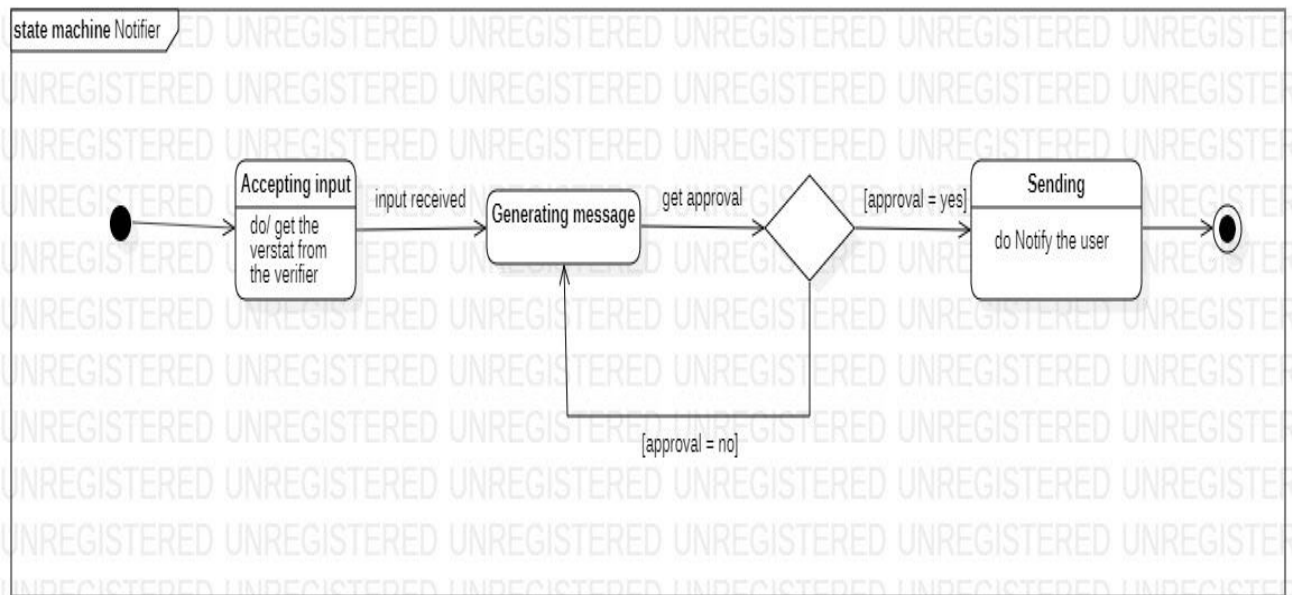
# State Models

A state model gives the complete picture of the states an object goes through during its lifetime. Only those objects with temporal behaviour possess a state diagram.

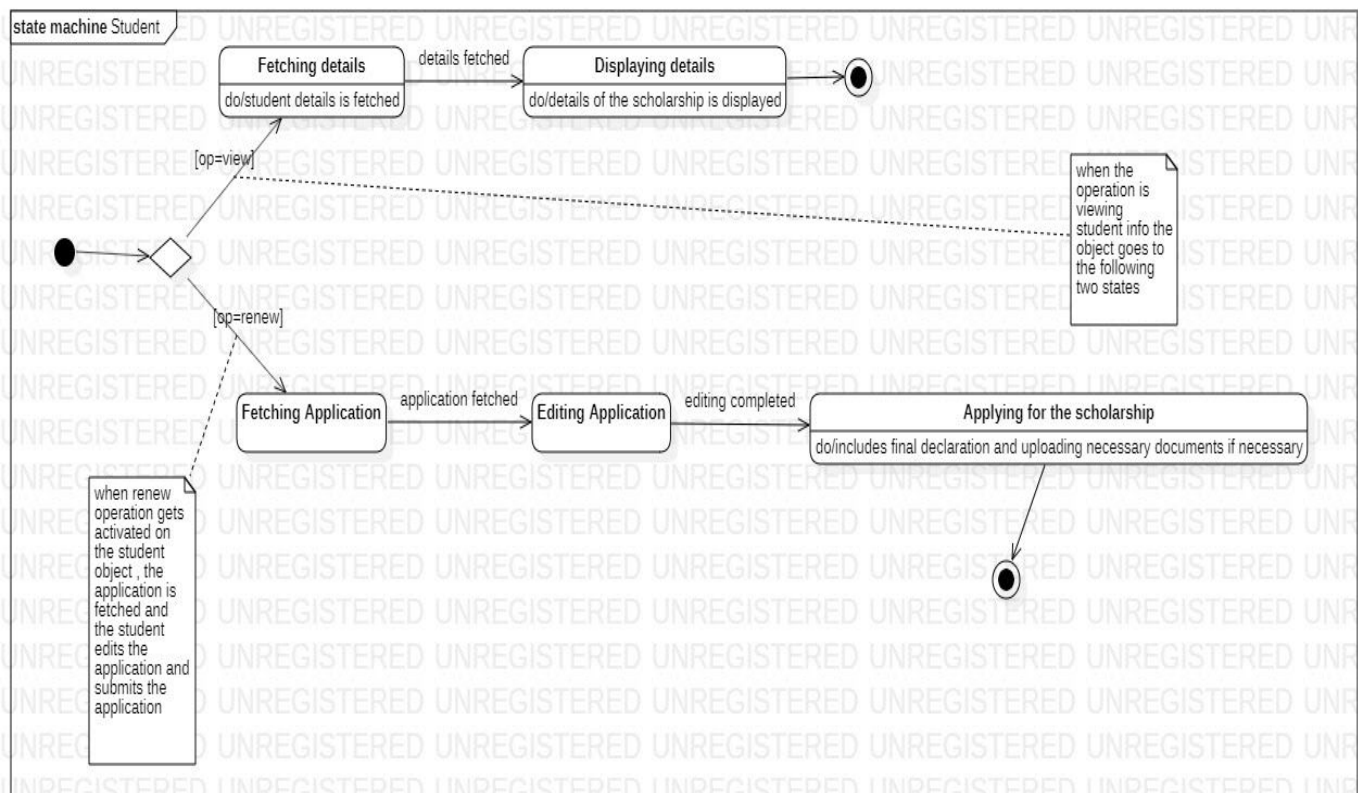
## Verifier



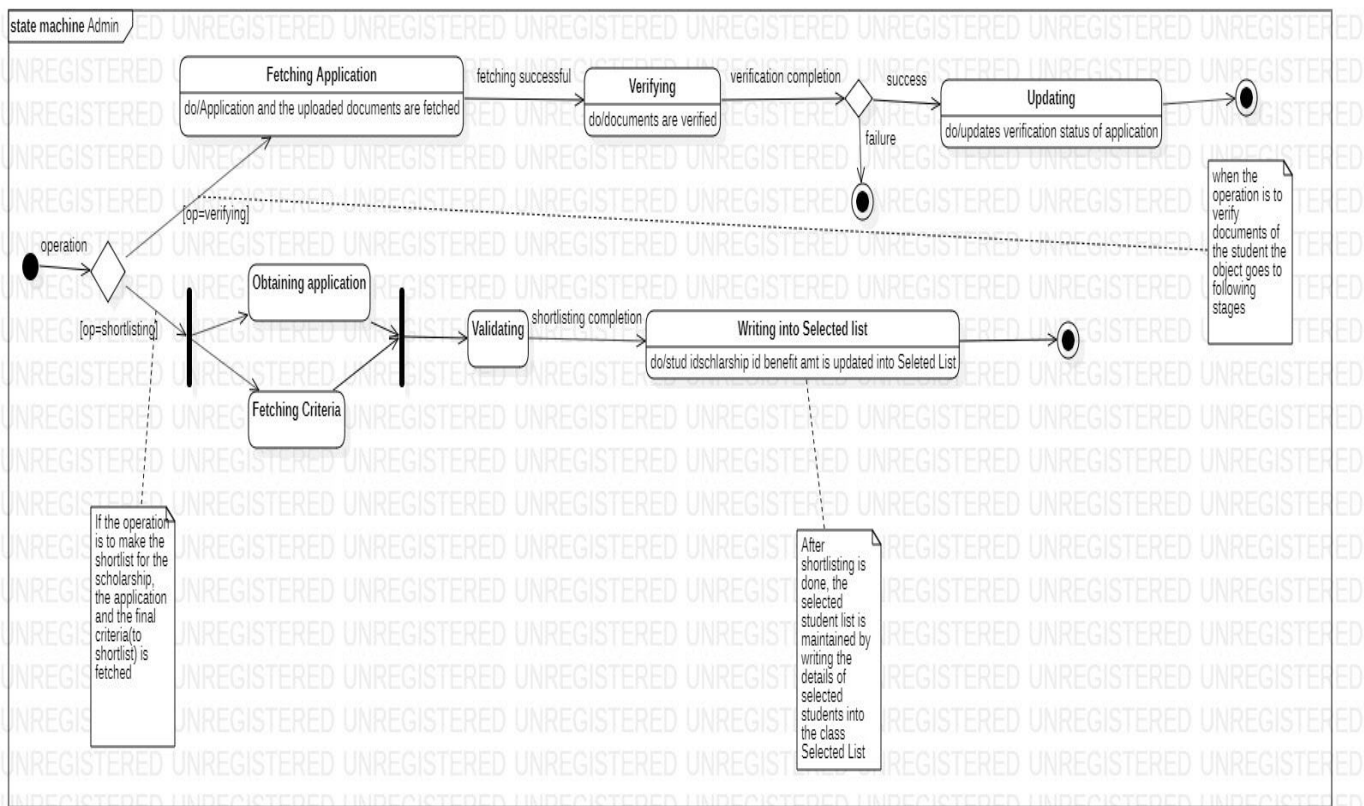
## Notifier:



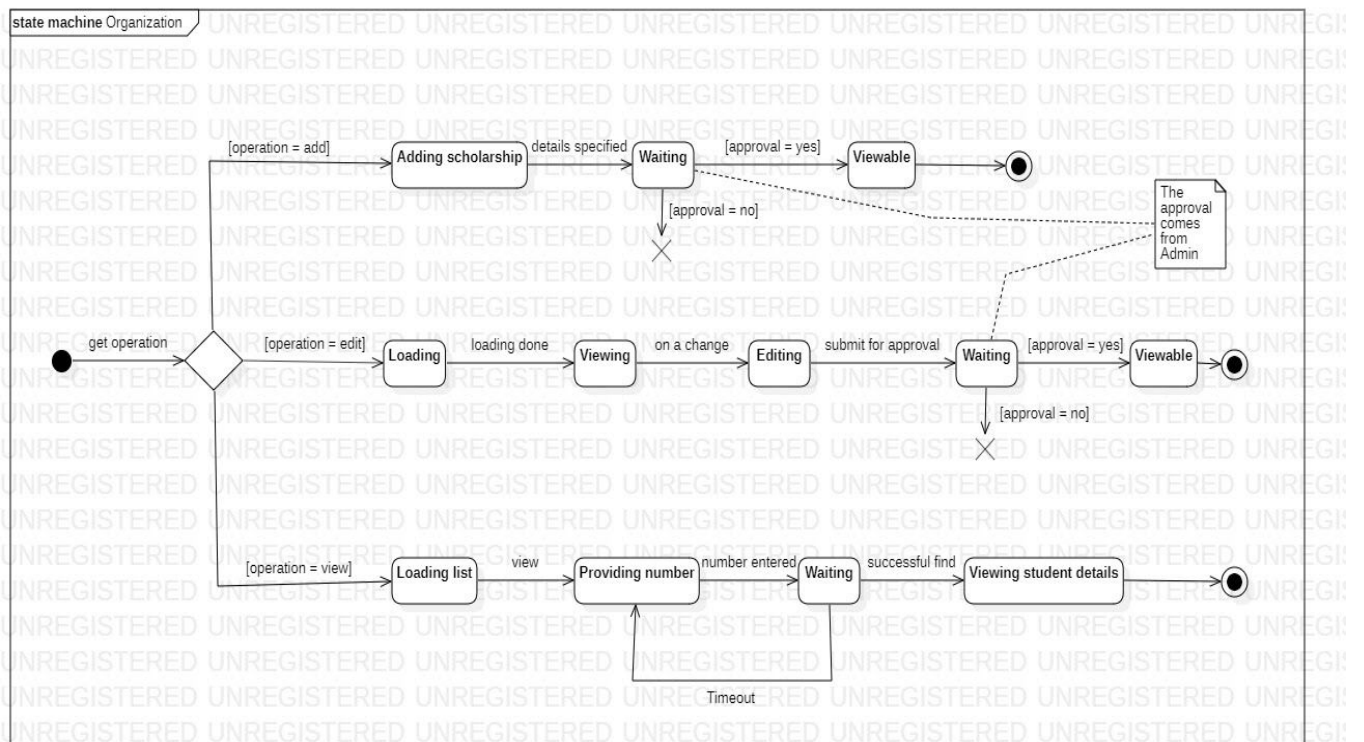
## Student:



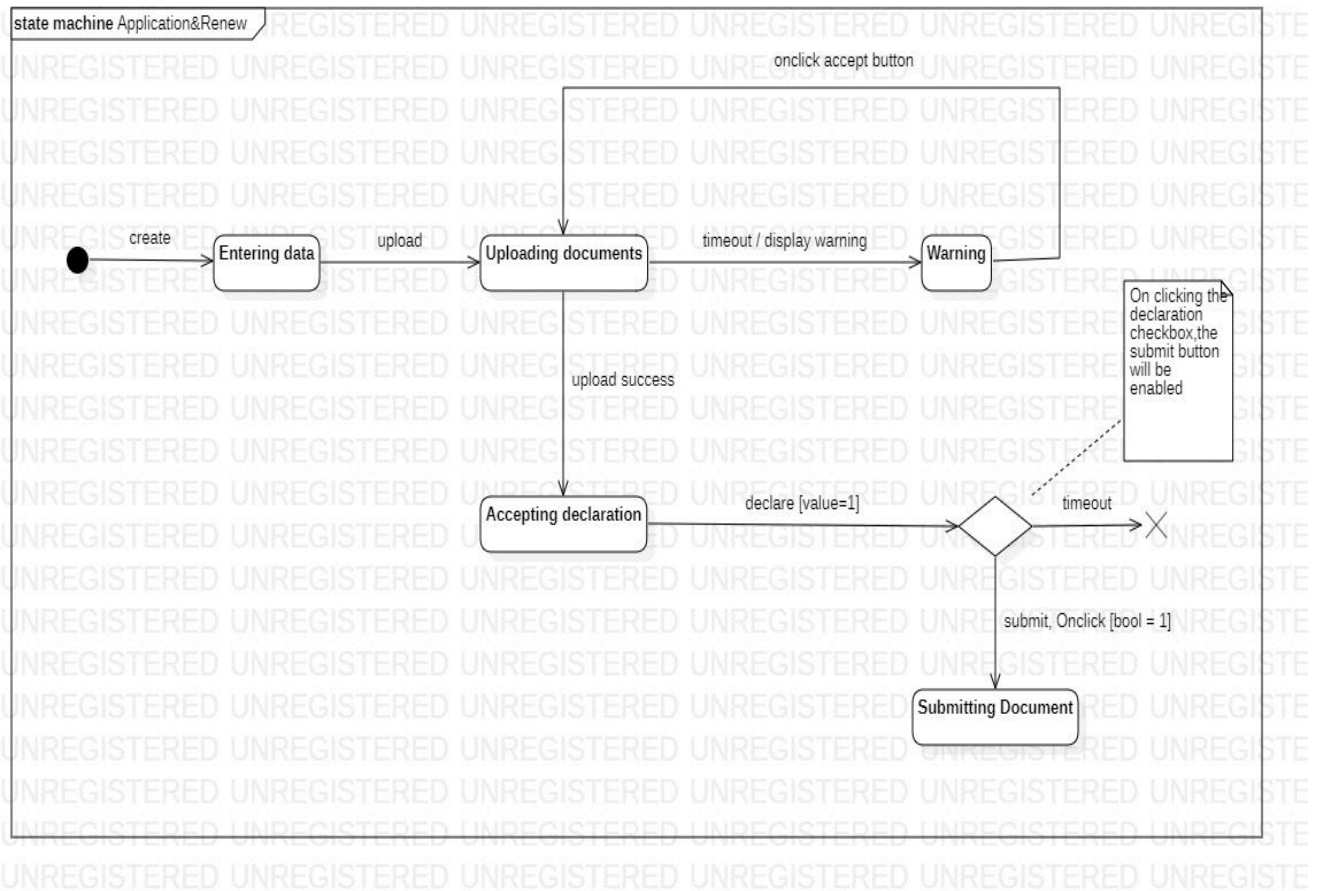
## Admin



## Organization



## Application&Renew



# Activity Modelling

## Activity Diagram:

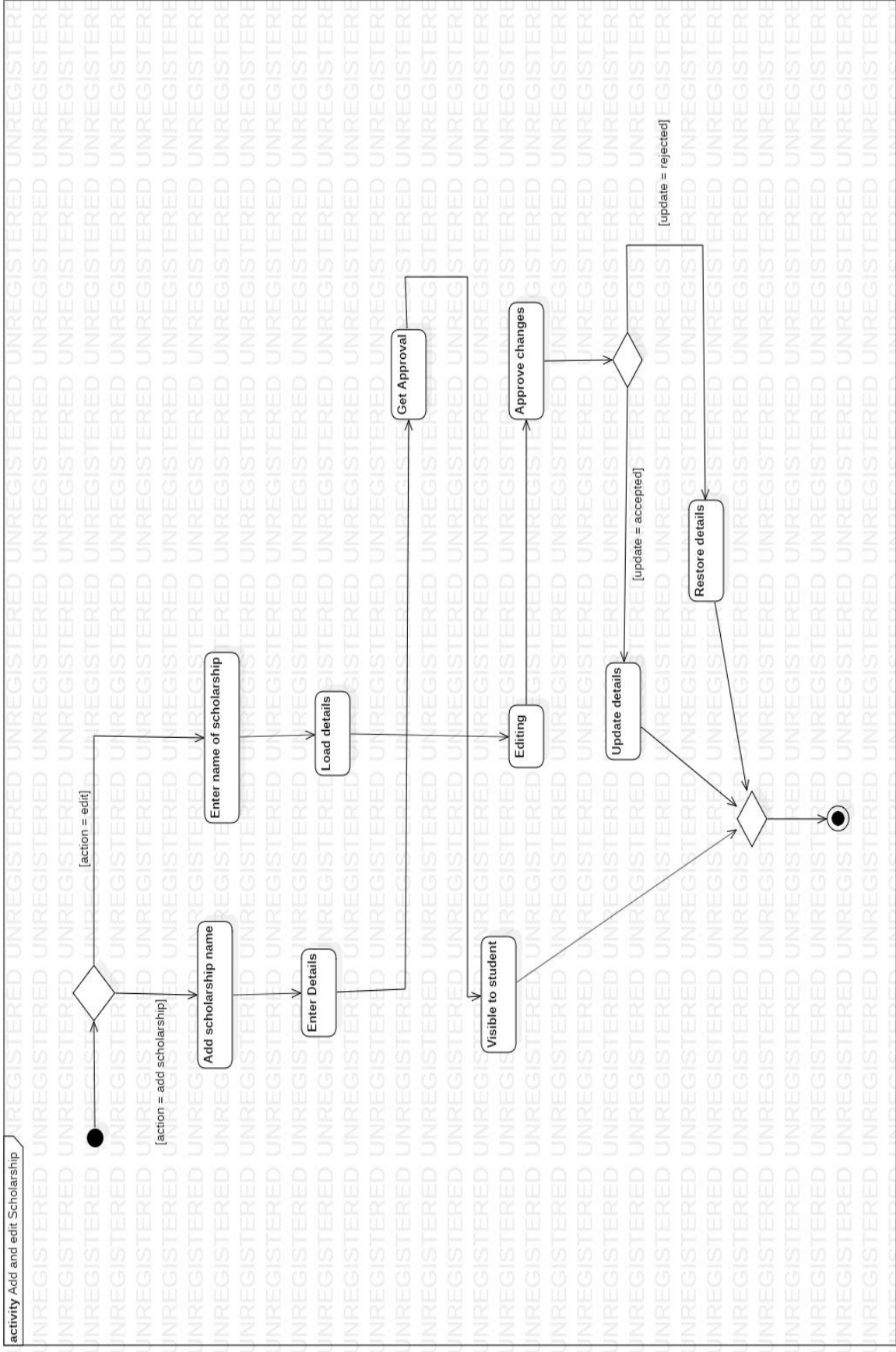
### **Add or update Scholarship:**

#### **Objects Involved:**

- Organization
- Admin
- Scholarship
- Details&Criteria

**Description:** The Organization can opt to either add a scholarship or edit the details and criteria of the scholarship it provides.

- When it opts to add a scholarship, it has to first enter the name of the scholarship followed by its details. Once it has been added, the scholarship has to get approval from the admin. Once it is approved it is made visible to the students.
- If the option is to edit the scholarship details, the organization has to first enter the name of the scholarship and the corresponding details are loaded. Then the organization can edit the details. Once it is edited it has to get approval from the admin, if it is approved the details are updated, and if not the old details are restored.





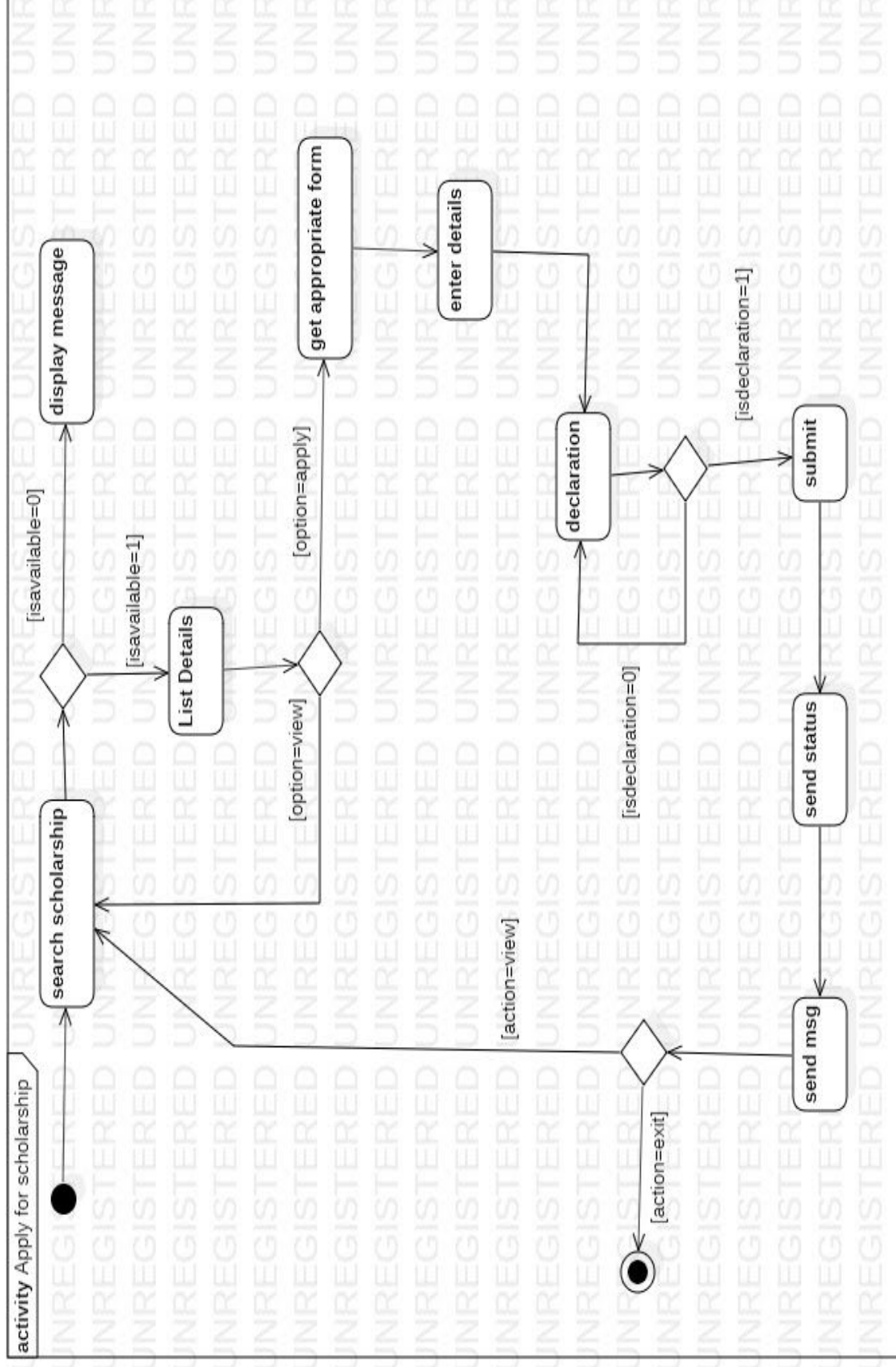
## **Apply for Scholarship:**

### **Objects Involved:**

- Student
- Scholarship
- Details&Criteria
- Application&Renew
- Notifier

**Description:** The Student can view for the scholarships and its details.

- If the scholarship is closed a message is shown to inform the student.
- Else the student can apply for the scholarship or continue browsing the scholarships available.
- If the student wants to apply ,then he/she has to fill the appropriate form and upload the necessary documents.On successful submission the student will be notified.



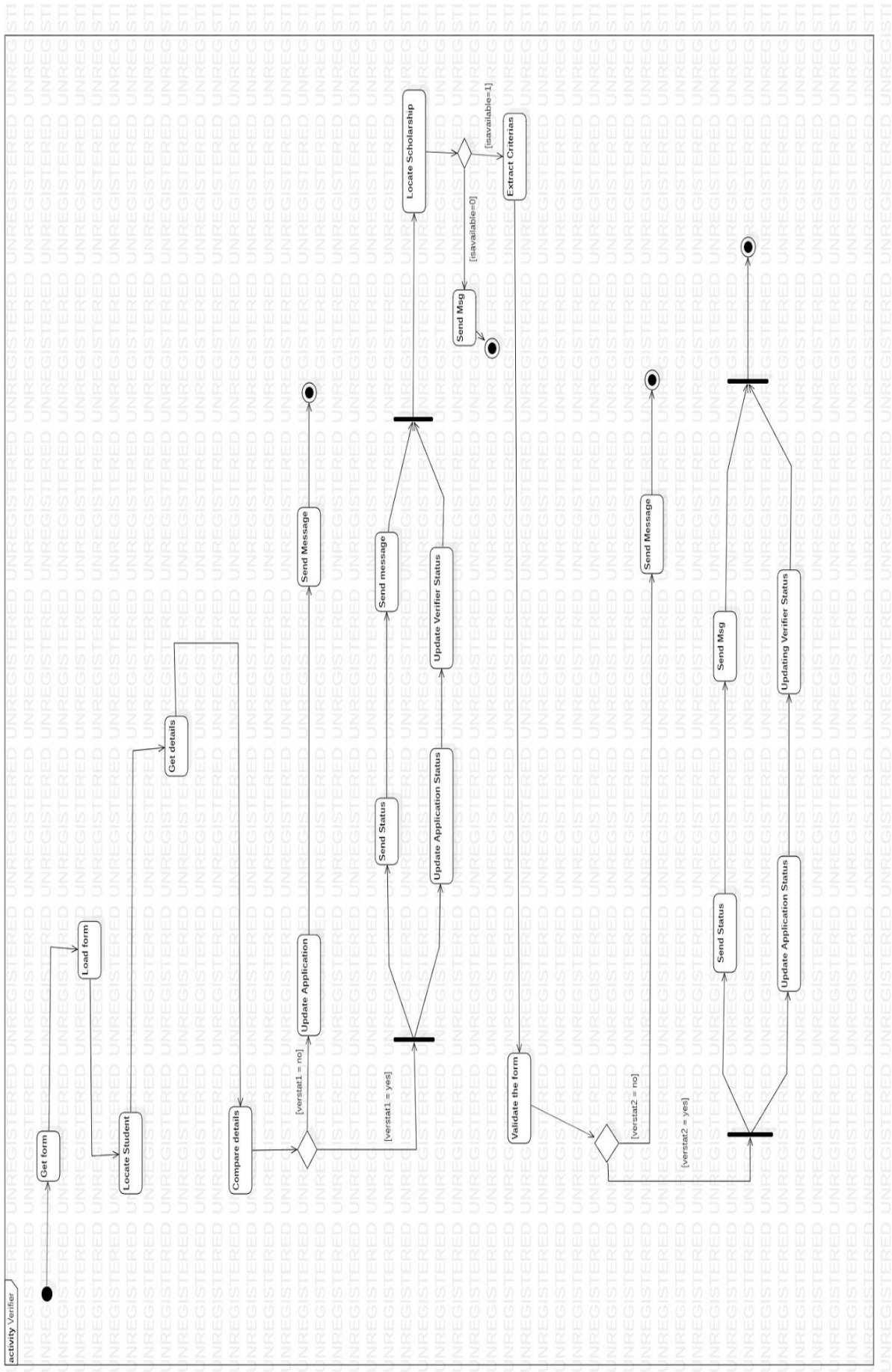
## **Validating the application:**

### **Objects Involved:**

- Student
- Student Details
- Scholarship
- Details&Criteria
- Notifier
- Application&Renew
- Verifier

**Description:** The submitted applications are verified in 2 stages by the verifier.

- The details of the student are fetched and are compared with the details filled by the candidate in the form. On success it goes for stage 2 of verification, else the student will be notified about the termination.
- Now, the application is checked against the criteria specified for the scholarship. On success, the student is notified and the appropriate updates take place. On failure, the student will be notified regarding the termination.



## **Verifying and Shortlisting:**

### **Objects Involved:**

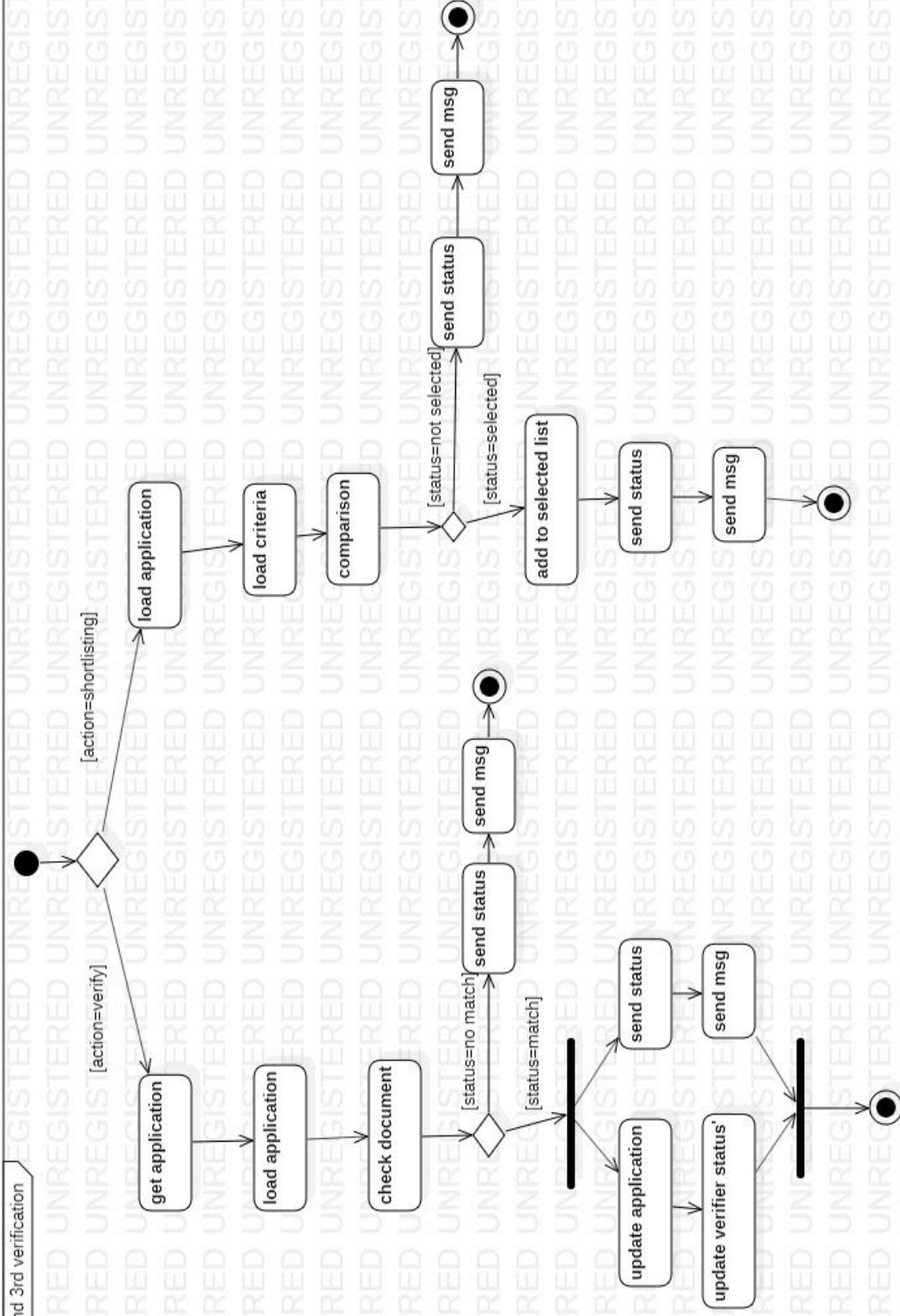
- Admin
- Scholarship
- Details&Criteria
- Notifier
- Application&Renew
- Verifier

**Description:** The admin has two main functionalities:

- 1) Verifying documents
- 2) Shortlisting

- If the admin opts to verify documents, then the corresponding applications and documents are fetched. If they are valid, appropriate updates are made and the student is notified.
- If the admin opts for shortlisting, then the final criteria for the scholarship is obtained and the students are shortlisted. The shortlisted students are added to the selected list and are also notified. If they are not selected, they receive a notification regarding the same.














activity Shortlisting and 3rd verification



# Code Generation

This PC > Desktop > OOAD > Lab > code > Model

Receipt\_files

Name	Date modified	Type	Size
 Admin.java	04-11-2020 19:27	JAVA File	1 KB
 Application & Renew.java	04-11-2020 19:27	JAVA File	2 KB
 Application.java	04-11-2020 19:27	JAVA File	1 KB
 Contact Info.java	04-11-2020 19:27	JAVA File	1 KB
 Details&criteria.java	04-11-2020 19:27	JAVA File	2 KB
 Notification.java	04-11-2020 19:27	JAVA File	1 KB
 Organization.java	04-11-2020 19:27	JAVA File	1 KB
 Scholarship.java	04-11-2020 19:27	JAVA File	1 KB
 Selected List.java	04-11-2020 19:27	JAVA File	1 KB
 Student Details.java	04-11-2020 19:27	JAVA File	2 KB
 Student.java	04-11-2020 19:27	JAVA File	1 KB
 User.java	04-11-2020 19:27	JAVA File	1 KB
 Verifier.java	04-11-2020 19:27	JAVA File	1 KB

## Admin.java

```
*Admin.java - Notepad
File Edit Format View Help

import java.util.*;
|
public class Admin extends User {

    public Admin() {
    }

    public void name;

    public void addscholar() {
        // TODO implement here
    }

    public void getverifiedapplicant() {
        // TODO implement here
    }

    public void checkdetails() {
        // TODO implement here
    }

    public void update() {
        // TODO implement here
    }

    public void shortlisting() {
        // TODO implement here
    }

}
```

---



## Application&Renew

 \*Application & Renew.java - Notepad

File Edit Format View Help

```
import java.util.*;

public class Application & Renew {

    public Application & Renew() {
    }
    public int applicantid;

    public String name;

    public int roll no;

    public Double cgpa;

    public String gender;

    public String nationality;

    public Double annual income;

    public Double percentage;

    public bool first graduate;

    public bool sports quota;

    public Contact Info stud contact;

    public Contact Info parent contact;

    public bool declaration;


    public int backlogs;
```

```
public bool declaration;  
  
public int backlogs;  
  
public int marks;  
  
public String caste;  
  
public int verification stat;  
  
public void uploaddocument() {  
    // TODO implement here  
}  
public void fetchappl(void int stat) {  
    // TODO implement here  
}  
public void uploaddetails() {  
    // TODO implement here  
}  
  
public void incrementer() {  
    // TODO implement here  
}
```

}

---

## Contactinfo:

 \*Contact Info.java - Notepad  
File Edit Format View Help

---

```
import java.util.*;

public class Contact Info {

    public Contact Info() {
    }
    public long int phoneno;

    public long int altphoneno;


    public String email;

    public Srring altemail;
|
    public void getcontact() {
        // TODO implement here
    }

}
```

---

## Details&criteria:

 \*Details&criteria.java - Notepad  
File Edit Format View Help

---

```
import java.util.*;

public class Details&criteria {

    public Details&criteria() {
    }

    public int year;

    public Double percentage;

    public Double cgpa;

    public Double annualincome;

    public String category;

    public String finalcriteria;

    public Double amt;

    public int validity in months;

    public Organization provider details;

    public int max backlogs;

    public int min SSLC;

    public int min HSC;
```

```
public int max backlogs;

public int min SSLC;

public int min HSC;

public void maxstud;

public void getcriteria() {
    // TODO implement here
}


public void updatecriteria() {
    // TODO implement here
}

public void addcriteria() {
    // TODO implement here
}
```

}

---

## Notification:

 \*Notification.java - Notepad  
File Edit Format View Help

---

```
import java.util.*;

public class Notification {

    public Notification() {
    }

    public void status;

    public void msg;

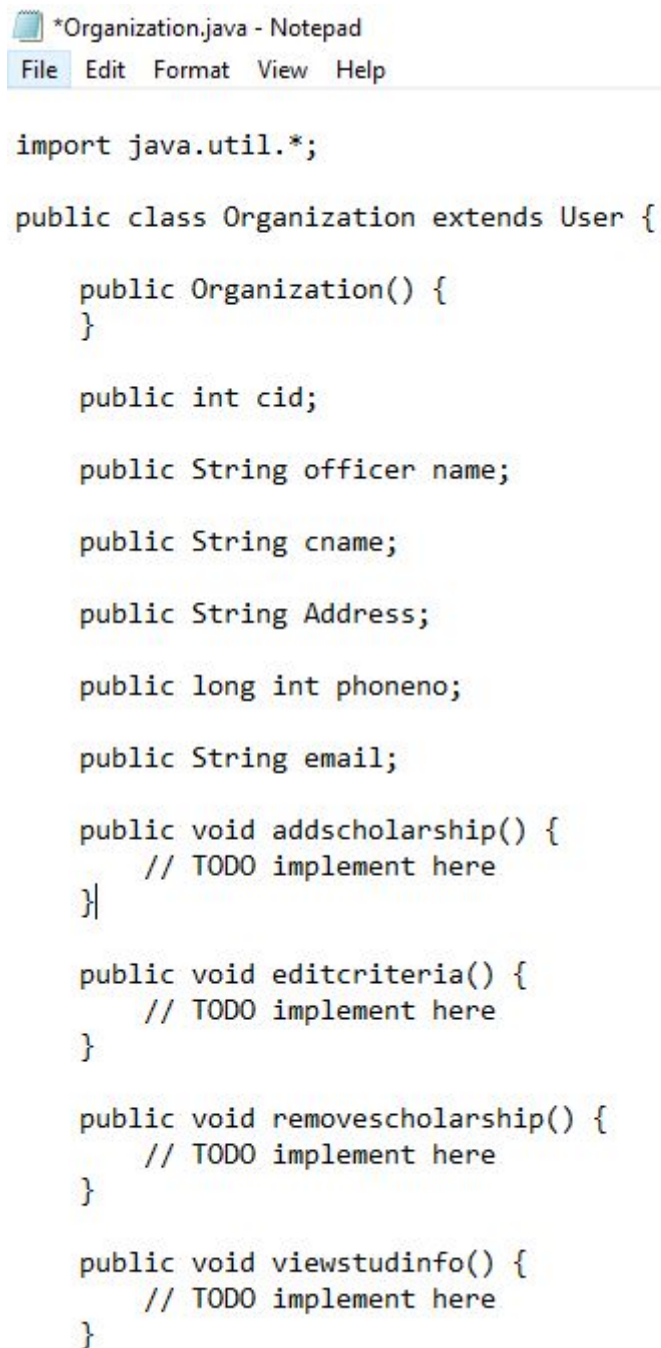
    public Verifier sends status;

    public void pushmsg() {
        // TODO implement here
    }
|
    public void genmsg() {
        // TODO implement here
    }

}
```

---

## Organization:



```
*Organization.java - Notepad
File Edit Format View Help

import java.util.*;

public class Organization extends User {

    public Organization() {
    }

    public int cid;

    public String officer name;

    public String cname;

    public String Address;

    public long int phoneno;

    public String email;

    public void addscholarship() {
        // TODO implement here
    }

    public void editcriteria() {
        // TODO implement here
    }

    public void removescholarship() {
        // TODO implement here
    }

    public void viewstudinfo() {
        // TODO implement here
    }
}
```

```
    public void editcriteria() {  
        // TODO implement here  
    }  
  
    public void removescholarship() {  
        // TODO implement here  
    }  
  
    public void viewstudinfo() {  
        // TODO implement here  
    }  
  
    public void getlist() {  
        // TODO implement here  
    }  
}
```

---



## Scholarship:

---

```
Scholarship - Notepad
File Edit Format View Help

import java.util.*;

public class Scholarship {

    public Scholarship() {
    }

    public int sid;

    public String sname;

    public bool accstatus;

    public Details&criteria has;

    public void getdetails() {
        // TODO implement here
    }

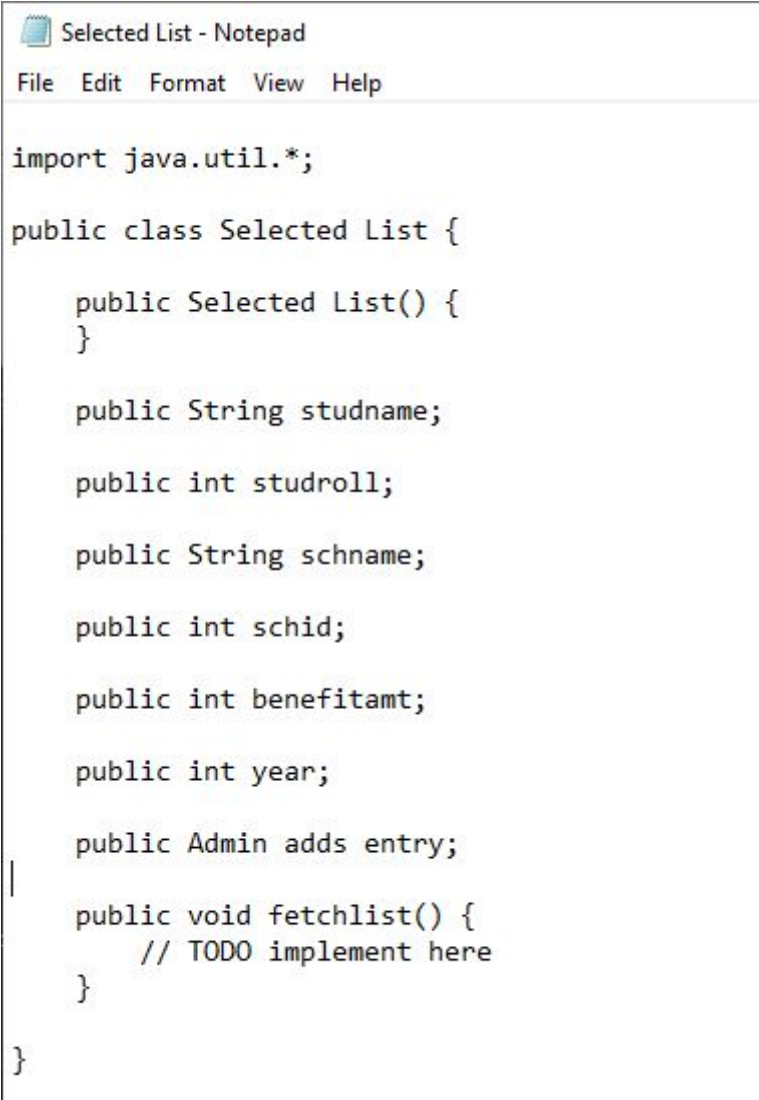
    public void deletescholarship() {
        // TODO implement here
    }

    public void getapproval() {
        // TODO implement here
    }

}
```

---

## Selected List:



```
Selected List - Notepad
File Edit Format View Help

import java.util.*;

public class Selected List {

    public Selected List() {
    }

    public String studname;

    public int studroll;

    public String schname;

    public int schid;

    public int benefitamt;

    public int year;

    public Admin adds entry;

    public void fetchlist() {
        // TODO implement here
    }

}
```

## Student Details:

```
Student Details - Notepad
File Edit Format View Help

import java.util.*;

public class Student Details {

    public Student Details() {
    }

    public Double cgpa;
    public date DOB;
    public String gender;
    public String religion;
    public String nationality;
    public long int phoneno;
    public String email;
    public String Address;
    public Double annualincome;
    public Double fee;
    public String quota;
    public int SSLC marks;
    public int HSC marks;
    public bool backlogs;
    public String caste;
    public bool isownsch;
    public Contact Info contact;

    public void viewdetails() {
        // TODO implement here
    }
    |
    public void updatedetails() {
        // TODO implement here
    }
}
```

---

## Student:

```
Student - Notepad
File Edit Format View Help

import java.util.*;

public class Student extends User {

    public Student() {
    }

    public String name;

    public int rollno;

    public Student Details has;

    public void searchscholarship() {
        // TODO implement here
    }

    public void getinfo() {
        // TODO implement here
    }

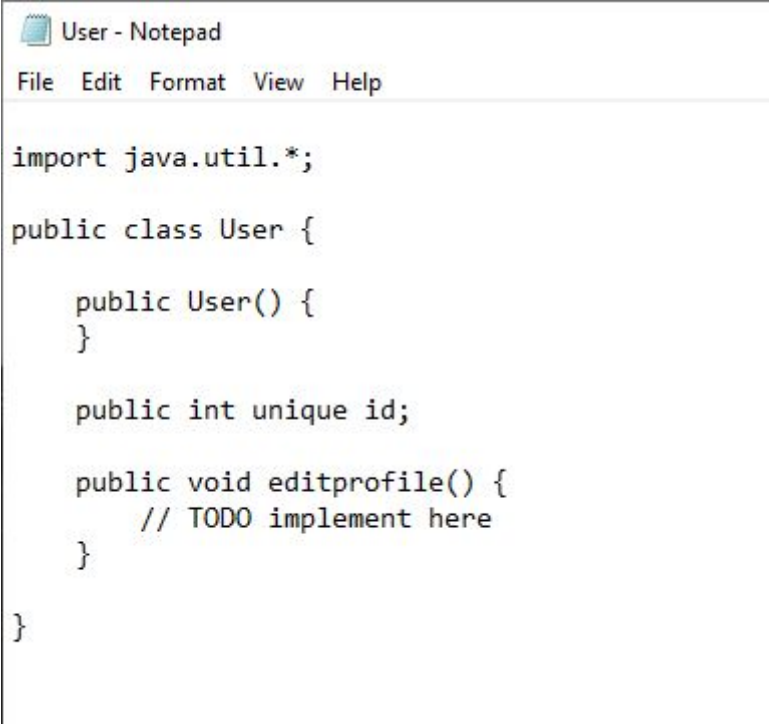
    public void viewstatus() {
        // TODO implement here
    }

    public void renew() {
        // TODO implement here
    }

    public void getlist() {
        // TODO implement here
    }

}
```

**User:**



```
import java.util.*;

public class User {

    public User() {
    }

    public int unique id;

    public void editprofile() {
        // TODO implement here
    }

}
```

## Verifier:

---

```
Verifier - Notepad
File Edit Format View Help

import java.util.*;

public class Verifier {

    public Verifier() {
    }

    public void verstat1;

    public void verstat2;

    public void verstat3;

    public void checkcriteria() {
        // TODO implement here
    }

    public void checkdetails() {
        // TODO implement here
    }

    public void updatestatus() {
        // TODO implement here
    }

}
```

---

## **Conclusion:**

We can implement the above model in programming languages such as Java and Python which supports Object Oriented Programming. We also need a proper database to store and retrieve the data when necessary.

Proper implementation of the above model with user friendly GUI and ambient size of database which stores all the documents helps many to access this system without any trouble.

When the implemented code of the model is complete, it can be linked with the college database so that student details can be fetched from that database and used for validation purposes. Doing this also helps the management to get all the details of the students in a single task instead of fetching student details separately from the database and the scholarship details separately and it also saves loads of data existing in two different databases.