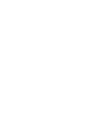


**

***Software Requirement and Design Specifications***

***[Society Club Portal]***

|  |  |
| --- | --- |
| *Course Code* | CS3004 |
| *Instructor* | *Mrs. Javeria Farooq* |
| *Project Team* | *19K-0289 IBRAHIM ALI*  *19K-1443 IBRAHIM ALI ASGHAR* |
| *Submission Date* | *22-Dec-2021* |



*Table of Contents*

[CS3004 1](#_Toc91023662)

[1. Introduction 4](#_Toc91023663)

[1.1. Purpose of Document 4](#_Toc91023664)

[1.2. Document Convention 4](#_Toc91023665)

[2. Overall System Description 4](#_Toc91023666)

[2.1. Project Background 4](#_Toc91023667)

[2.2. Project Scope 4](#_Toc91023668)

[2.3. Not in Scope 4](#_Toc91023669)

[2.4. Stakeholders 4](#_Toc91023670)

[2.5. Operating Environment 4](#_Toc91023671)

[2.6. System Constraints 4](#_Toc91023672)

[2.7. Assumptions & Dependencies 5](#_Toc91023673)

[3. External Interface Requirements 5](#_Toc91023674)

[3.1. Hardware Interfaces 5](#_Toc91023675)

[3.2. Software Interfaces 5](#_Toc91023676)

[3.3. Communications Interfaces 5](#_Toc91023677)

[4. Functional Requirements 5](#_Toc91023678)

[4.1. Functional Hierarchy 5](#_Toc91023679)

[4.2. Use Case Diagram 6](#_Toc91023680)

[5. Non-functional Requirements 19](#_Toc91023681)

[5.1. Performance Requirements 19](#_Toc91023682)

[5.2. Safety Requirements 19](#_Toc91023683)

[5.3. Security Requirements 19](#_Toc91023684)

[SDS 20](#_Toc91023685)

[6. System Architecture 21](#_Toc91023686)

[6.1. System Level Architecture 21](#_Toc91023687)

[6.1.1. Component Diagram 21](#_Toc91023688)

[6.1.2. Deployment Diagram 21](#_Toc91023689)

[6.2. Software Architecture 22](#_Toc91023690)

[7. Detailed System Design 22](#_Toc91023691)

[7.1. Database Design 22](#_Toc91023692)

[7.1.1. ER Diagram 23](#_Toc91023693)

[8. Application Design 24](#_Toc91023694)

[8.1. Sequence Diagram 24](#_Toc91023695)

[8.1.1. ADMIN 24](#_Toc91023696)

[8.1.2. SOCIETY 25](#_Toc91023697)

[8.1.3. STUDENT 26](#_Toc91023698)

[8.2. Communication Diagram 27](#_Toc91023699)

[8.2.1. ADMIN 27](#_Toc91023700)

[8.2.2. SOCIETY 27](#_Toc91023701)

[8.2.3. STUDENT 28](#_Toc91023702)

[8.3. Activity Diagram 29](#_Toc91023703)

[8.3.1. ADMIN 29](#_Toc91023704)

[8.3.2. SOCIETY 30](#_Toc91023705)

[8.3.3. STUDENT 31](#_Toc91023706)

[8.3.4. LOGIN 31](#_Toc91023707)

[8.4. State Diagram 32](#_Toc91023708)

# Introduction

#### Purpose of Document

*This document is to fulfil the software requirements and Design Specifications for project “Society Club Portal “.*

#### Document Convention

*Font used in this document is Times New Roman and font size is 12.*

# Overall System Description

#### Project Background

*Majority of the Students find it difficult to keep the track of upcoming events and society inductions through emails. We have developed a “Society Club Portal” which keeps all the track regarding events and inductions in one place.*

#### Project Scope

* + *Student will be notified about the events and inductions being conducted by societies.*
  + *Society will manage their events and inductions for the students.*
  + *Admin can add/update/delete students and societies from the portal.*

#### Not in Scope

*It will not manage the overall venues in the university. There are fixed venues for event scheduling in our project.*

#### Stakeholders

* + *Admin*
  + *Students*
  + *Societies*
  + *Website Developer*
  + *Database Developer*

#### Operating Environment

*The operating system used in our project is Windows 10. The software is Microsoft Visual Studio Code for frontend and backend connectivity. SSMS for Database. ASP.NET MVC is used for backend and frontend.*

#### System Constraints

* ***Software constraints:*** *Browser (Google Chrome, Internet Explorer, Firefox etc)*
* ***Hardware constraints:*** *Must have Running System*
* ***Cultural constraints:*** *English*
* ***User constraints****: Website is developed for University Societies and Students.*

#### Assumptions & Dependencies

* *For now, it is for single user because it is launched on local host.*

# External Interface Requirements

#### Hardware Interfaces

*Laptop/PC, Keyboard, and Mouse.*

#### Software Interfaces

*The SSMS is based for the Sequential Query Language no external libraries are required to run this software.*

*For frontend and backend, we have use ASP.NET MVC.*

#### Communications Interfaces

*The user will need Web Browser. For development purposes the web browser used is Google Chrome and is accessible through local host.*

# Functional Requirements

#### Functional Hierarchy

*We have three major functional requirements*

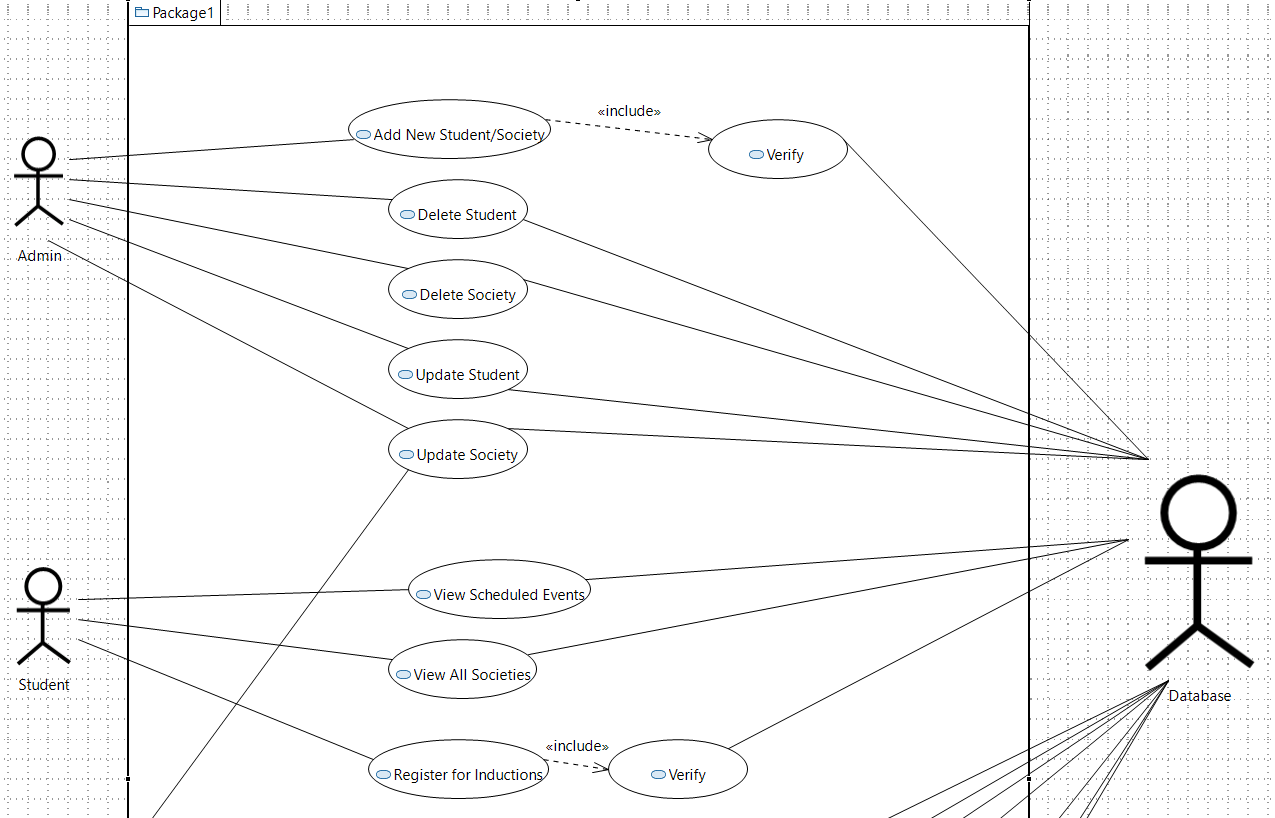
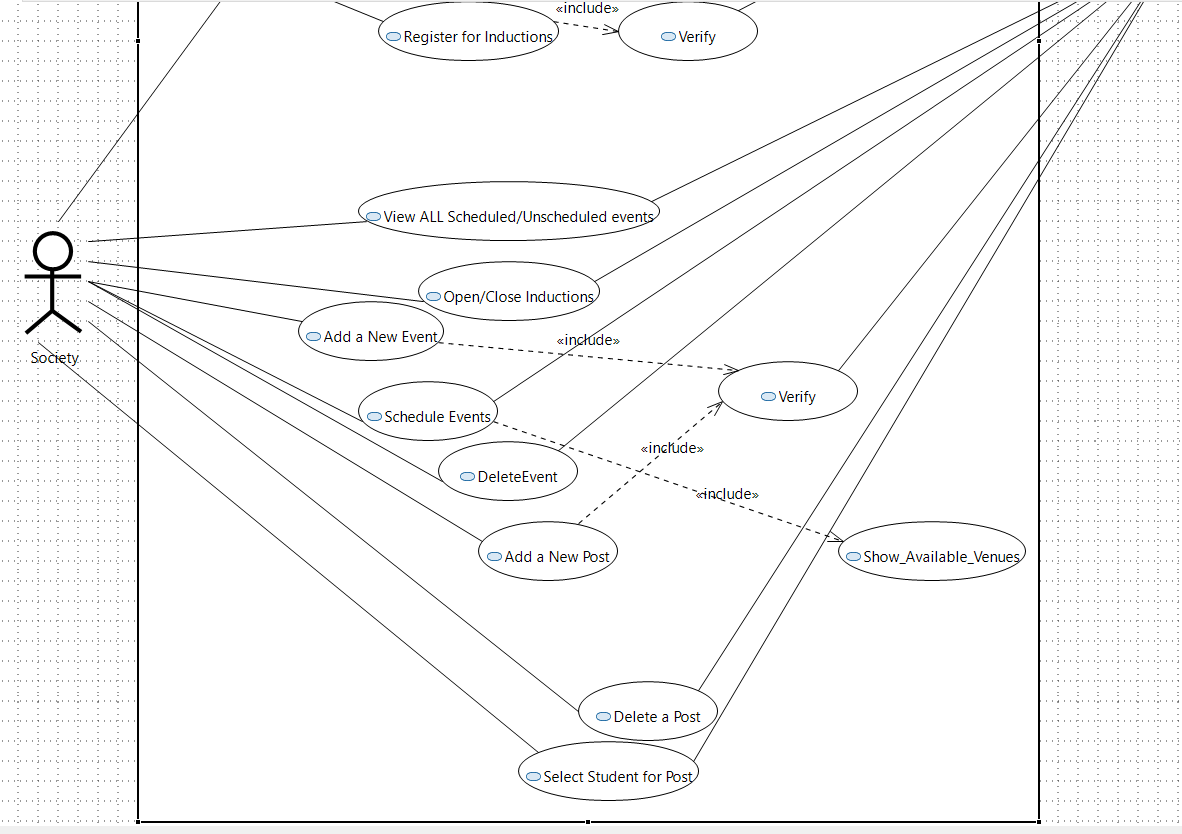
* *Admin*
* *Student*
* *Society*

*Admin can add/update/delete students and societies in Portal.*

*The students can view all the upcoming events, all the societies, register for inductions.*

*The societies can add/delete events, schedule events, open/close inductions, add/delete posts, select registered students on posts.*

#### Use Case Diagram



|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | ADD NEW STUDENT |
| **Use Case Description:** To add student to the society club portal | |
| **Primary actor:** Admin | **Other actors:** Database |
| **Stakeholders:** Student |  |
| **Relationships**  **Includes:** Verify   **Extends:** | |
| **Flow of Events:**  **1. Actor will click on add student**  **2.**  He will add the details of the student  **3. After then the system will verify** | |
| **Post-conditions:**  The student will be able to join the society club portal and follow up to it. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | ADD NEW SOCIETY |
| **Use Case Description:** To add society to the society club portal | |
| **Primary actor:** Admin | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes:**  Verify   **Extends:** | |
| **Flow of Events:**  **1. Actor will click on add society**  **2.**  He will add the details of the society  **3. After then the system will verify** | |
| **Post-conditions:**  The society will be able to put their activities on this portal so that student can follow up.. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | DELETE STUDENT |
| **Use Case Description:** To delete student from the society club portal | |
| **Primary actor:** Admin | **Other actors:** Database |
| **Stakeholders:** Student |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  **1. Actor will click on delete student**  **2.**  He will delete the details of the student.  3. Students will be delete from every table of database. | |
| **Post-conditions:**  Student will not be able to login to this portal afterwards. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | DELETE SOCIETY |
| **Use Case Description:** To delete society from the society club portal | |
| **Primary actor:** Admin | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  **1. Actor will click on delete society**  **2.**  He will delete the details of the society.  3. Society will be deleted from every table of database. | |
| **Post-conditions:**  Society will not be able to put their activities on to this portal afterwards. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Update Student |
| **Use Case Description:** To update students’ details in society club portal | |
| **Primary actor:** Admin, Student | **Other actors:** Database |
| **Stakeholders:** Student |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  **1. Actor will click on update student**  **2.**  He will update the details of the student. | |
| **Post-conditions:**  Student will be updated and his/her updated details will be visible to this portal afterwards. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Update Society |
| **Use Case Description:** To update society’ details in society club portal | |
| **Primary actor:** Admin, Society | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  **1. Actor will click on update society.**  **2.**  He will update the details of the society. | |
| **Post-conditions:**  Society will be updated and their updated details will be visible to this portal afterwards. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | View Scheduled Events |
| **Use Case Description:** Actor will look up to the scheduled events of the societies. | |
| **Primary actor:** Student | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  **1. Actor will click on view events.**  **2.**  He will be able to see the upcoming events of details. | |
| **Post-conditions:** --- | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | View all Societies |
| **Use Case Description:** Actor will look up to the details of every society. | |
| **Primary actor:** Student | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes: -**  **Extends:**  - | |
| **Flow of Events:**  1. Actor will click on Societies.  **2.**  He will be sent to the new page where he will be able to look details of society in table form. | |
| **Post-conditions:** He can register for the society. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Register for Inductions |
| **Use Case Description:** Actor can register in the society inductions. | |
| **Primary actor:** Student | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes: -**Verify.  **Extends:**  - | |
| **Flow of Events:**   1. If the inductions will be open then he will be able to apply 2. But if he is already in the hierarchy then he will be not eligible to do so. | |
| **Post-conditions:** If society likes to include him in their hierarchy, then they will select them using another use-case. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | View All Scheduled/Unscheduled Events |
| **Use Case Description:** Actor can see the events of their own, whether they are scheduled or not. | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes:**  **Extends:**  - | |
| **Flow of Events:**   1. Actor will click on Events menu and then a new window will appear. 2. This window will be having all the scheduled events first and then all the non-scheduled ones. | |
| **Post-conditions:** If society want, they can schedule their non-scheduled events. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Open/Close Inductions |
| **Use Case Description:** Actor can open or close their inductions and look at their posts. | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes:**  **Extends:**  - | |
| **Flow of Events:**   1. If the inductions are open then actor can click on open induction and allow students to register for their inductions. 2. If the inductions are close then actor can click on close induction and remove all the registration of the students for their society. | |
| **Post-conditions:** This will notify student that inductions are open you can apply now. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Description** | | | |
| **Use Case name:** | | Add a New Event | |
| **Use Case Description:** Actor can add new event to their society | | | |
| **Primary actor:** Society | | **Other actors:** Database | |
| **Stakeholders:** Society | |  | |
| **Relationships**  **Includes:**  Verify  **Extends:**  - | | | |
| **Flow of Events:**   1. Actor will click on ADD NEW EVENT. 2. Then he will fill up the details. 3. Click on Submit | | | |
| **Post-conditions:** After adding the event actor can schedule that event. | | | |
|  | | | |
| **Use Case Description** | | |
|  | | |
| **Use Case name:** | Schedule Event | |
| **Use Case Description:** Actor can schedule their new/existing event of their society. | | |
| **Primary actor:** Society | **Other actors:** Database | |
| **Stakeholders:** Society |  | |
| **Relationships**  **Includes:** Show\_available\_venues()  **Extends:**  - | | |
| **Flow of Events:**   1. Actor will click on Schedule EVENT. 2. Then he will fill up the details. 3. Then system will help out to show the available venues at that the particular time. 4. Then Actor will select the best possible venue according to him/her. 5. Then actor will click on Schedule. 6. Hence the event will be scheduled. | | |
| **Post-conditions:** After scheduling this event will be visible to student and also to the actor as scheduled. | | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Delete EVENT |
| **Use Case Description:** Actor can delete event from their society | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society |  |
| **Relationships**  **Includes:**  **Extends:**  - | |
| **Flow of Events:**  1.Actor will click on delete EVENT.  2.Then the specific EVENT will immediately be deleted and disappear from the screen. | |
| **Post-conditions:** Society will lose that event and all the scheduling will also be deleted with the deletion of that event. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Add a new POST |
| **Use Case Description:** Actor can add new post to their society. | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society, Student |  |
| **Relationships**  **Includes:** Verify  **Extends:**  - | |
| **Flow of Events:**  1.Actor will click on ADD NEW POST.  2.Then he will fill up the details.  3.Click on Submit | |
| **Post-conditions:** After adding POST this post can be populated by adding student on post.  Can also delete the post. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Delete POST |
| **Use Case Description:** Actor can delete post from their society | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society, Student |  |
| **Relationships**  **Includes:**  **Extends:**  - | |
| **Flow of Events:**  1.Actor will click on delete POST.  2.Then it will immediately be deleted and disappear from the screen. | |
| **Post-conditions:** Student will lose its position from that post and that post will be no more for this in this portal. | |

|  |  |
| --- | --- |
| **Use Case Description** | |
| **Use Case name:** | Select Student |
| **Use Case Description:** Actor can select the registered student for their posts. | |
| **Primary actor:** Society | **Other actors:** Database |
| **Stakeholders:** Society, Student |  |
| **Relationships**  **Includes:**  **Extends:**  - | |
| **Flow of Events:**  1.Actor will click on SELECT STUDENT  2. Then the student will be selected for the following POST. | |
| **Post-conditions:** After selection the student will be in the hierarchy of that society. | |

# Non-functional Requirements

#### Performance Requirements

*The Website Interface is user Friendly and can be improved further. Moreover, the data retrieval is fast but the speed of server depends on the system we are using. As the project is on Local host that’s why until now there is single user and hence no use of concurrency but can be implemented once it the website is published.*

#### Safety Requirements

*The system should be fast else the website will be down and the server files will not work properly, but as such it will not damage, lose or cause harm to the system or data.*

#### Security Requirements

*Every Society/Student can login into their account via UserID/Password, this provides the privacy.*

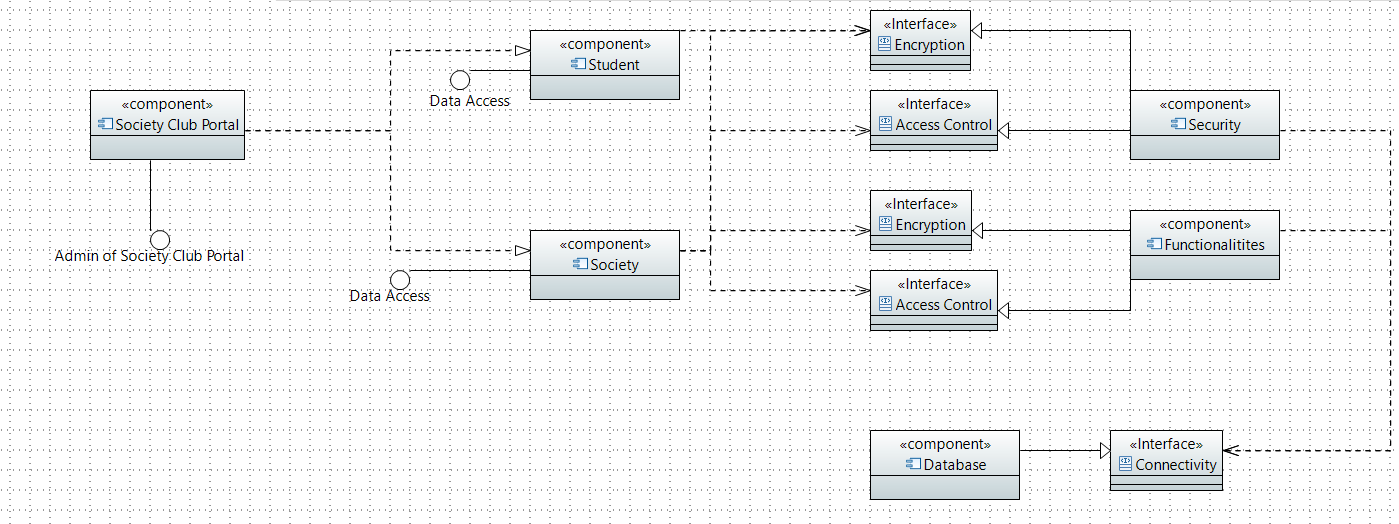
## SDS

# System Architecture

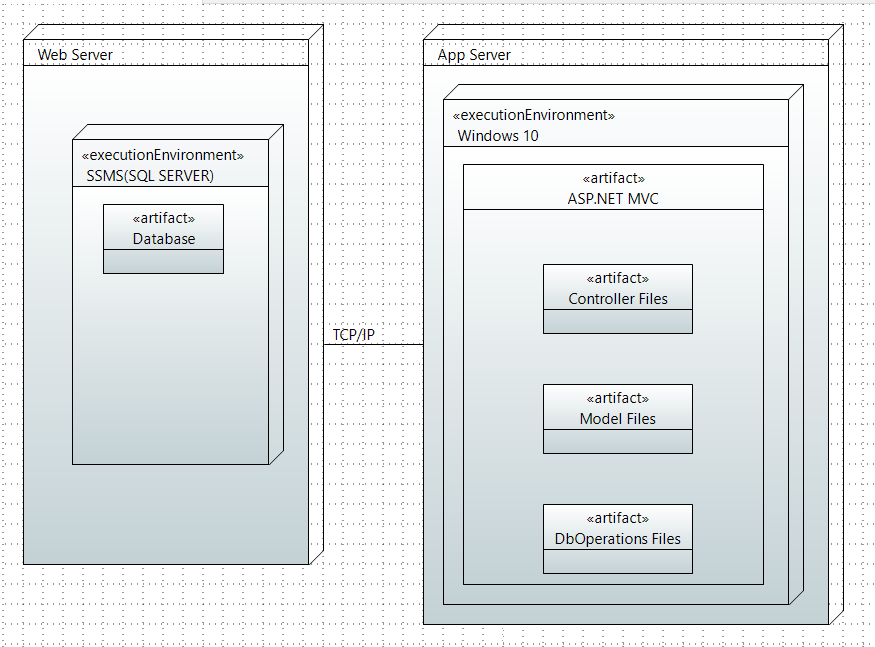
#### System Level Architecture

##### Component Diagram

*Our System consists of three modules admin, student, society. As System level architecture show the top level that is why no internal description is given.*

**

##### Deployment Diagram

**

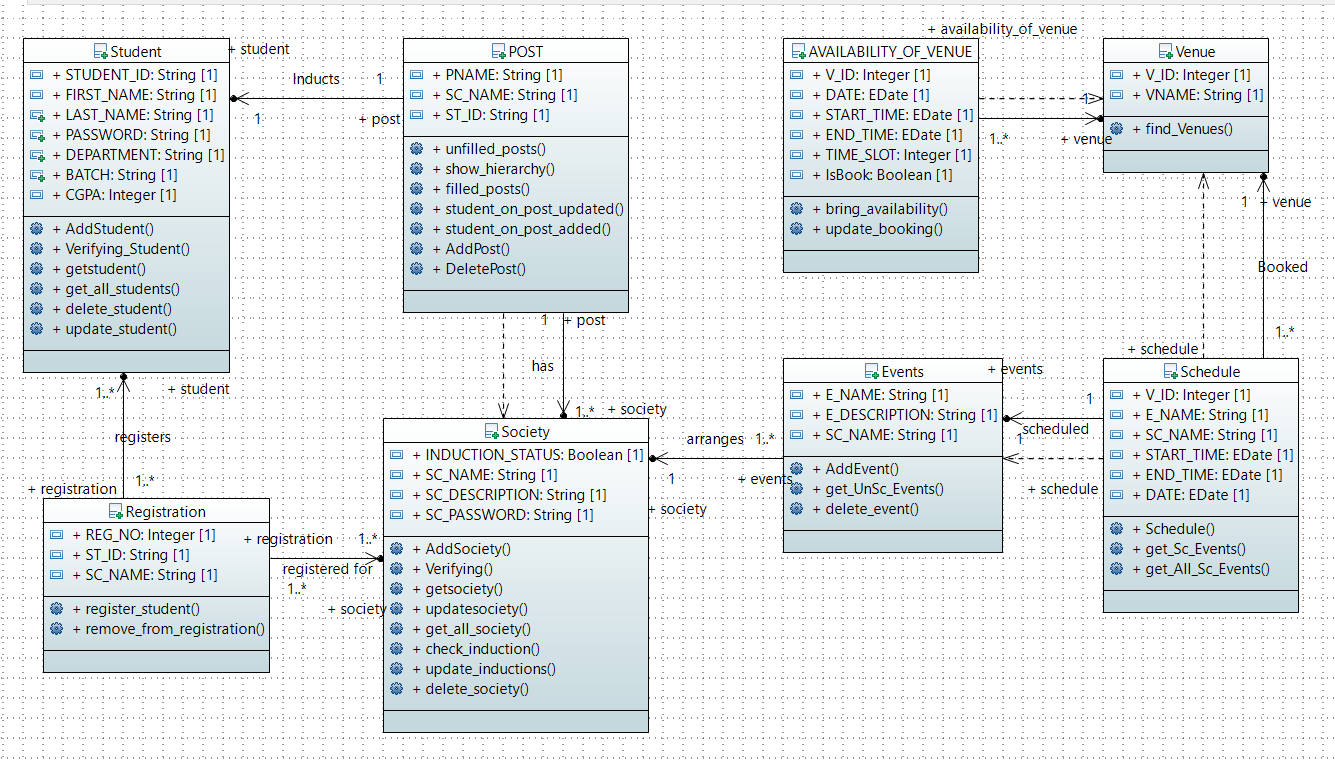
#### Software Architecture

View

Controllers

Model (Database)

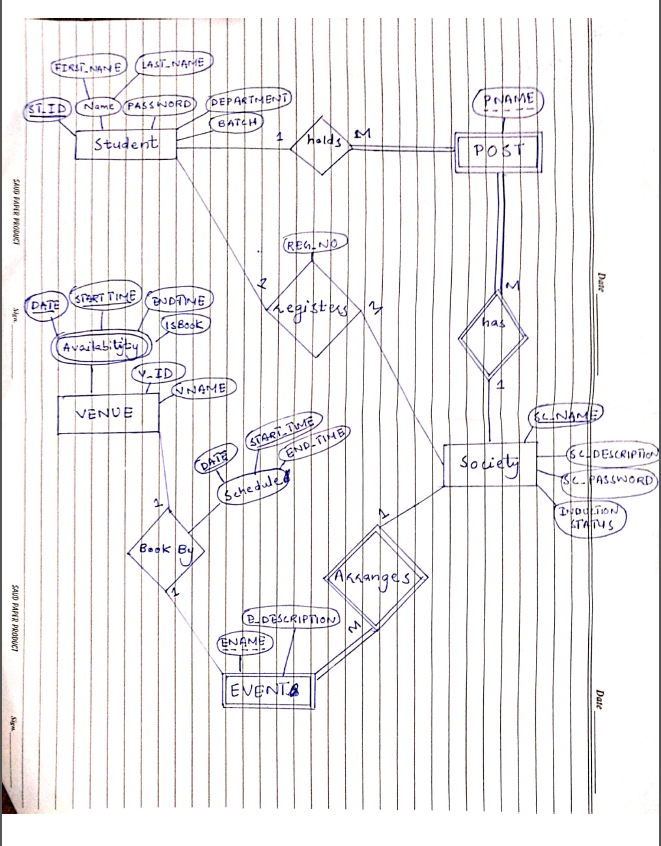
# Detailed System Design



#### Database Design

*Database designed is represented by the following ERD:*

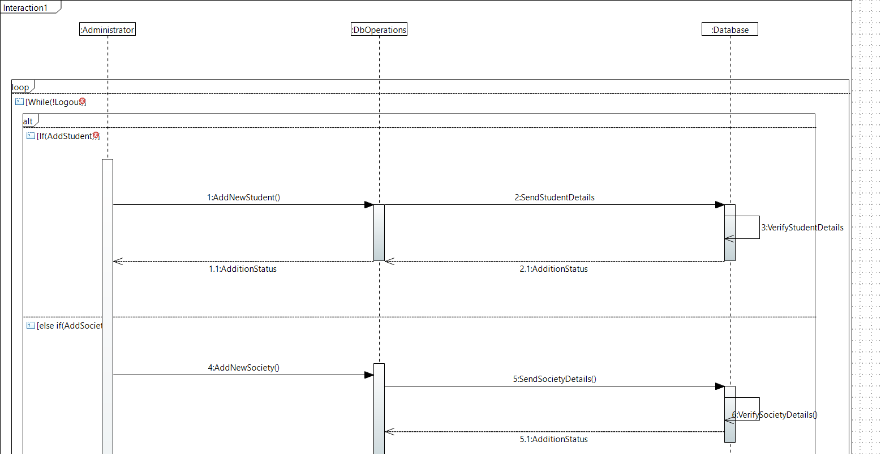
##### ER Diagram

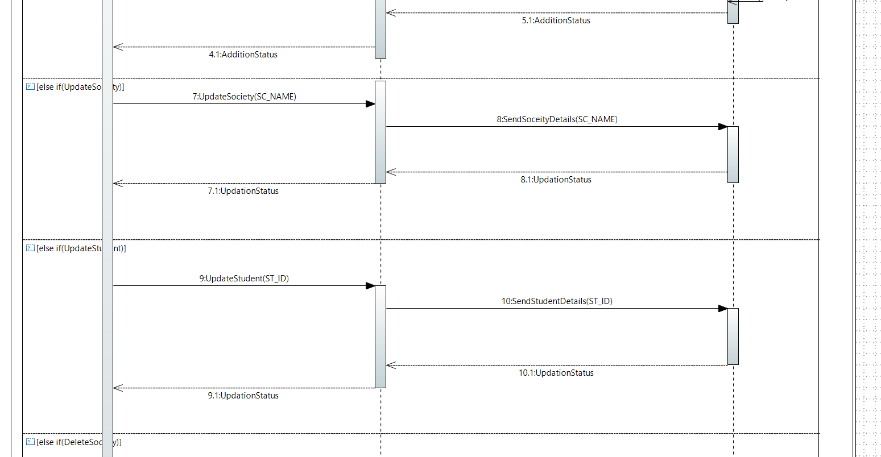
**

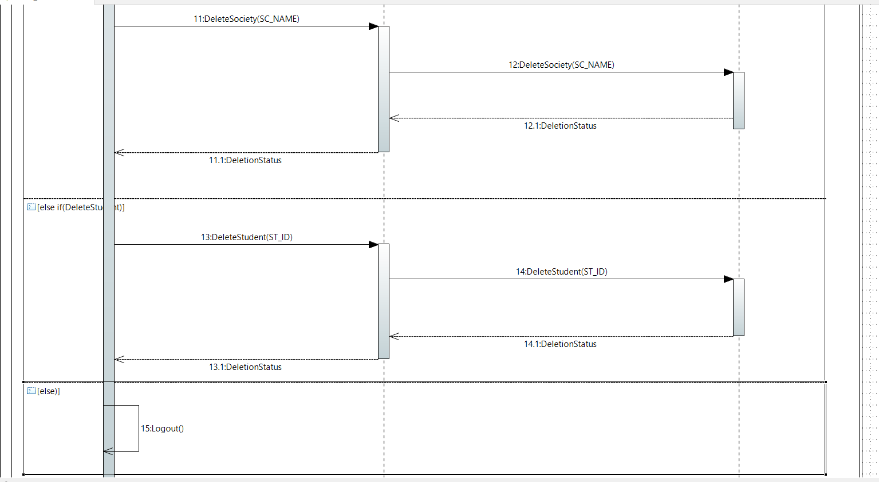
# Application Design

#### Sequence Diagram

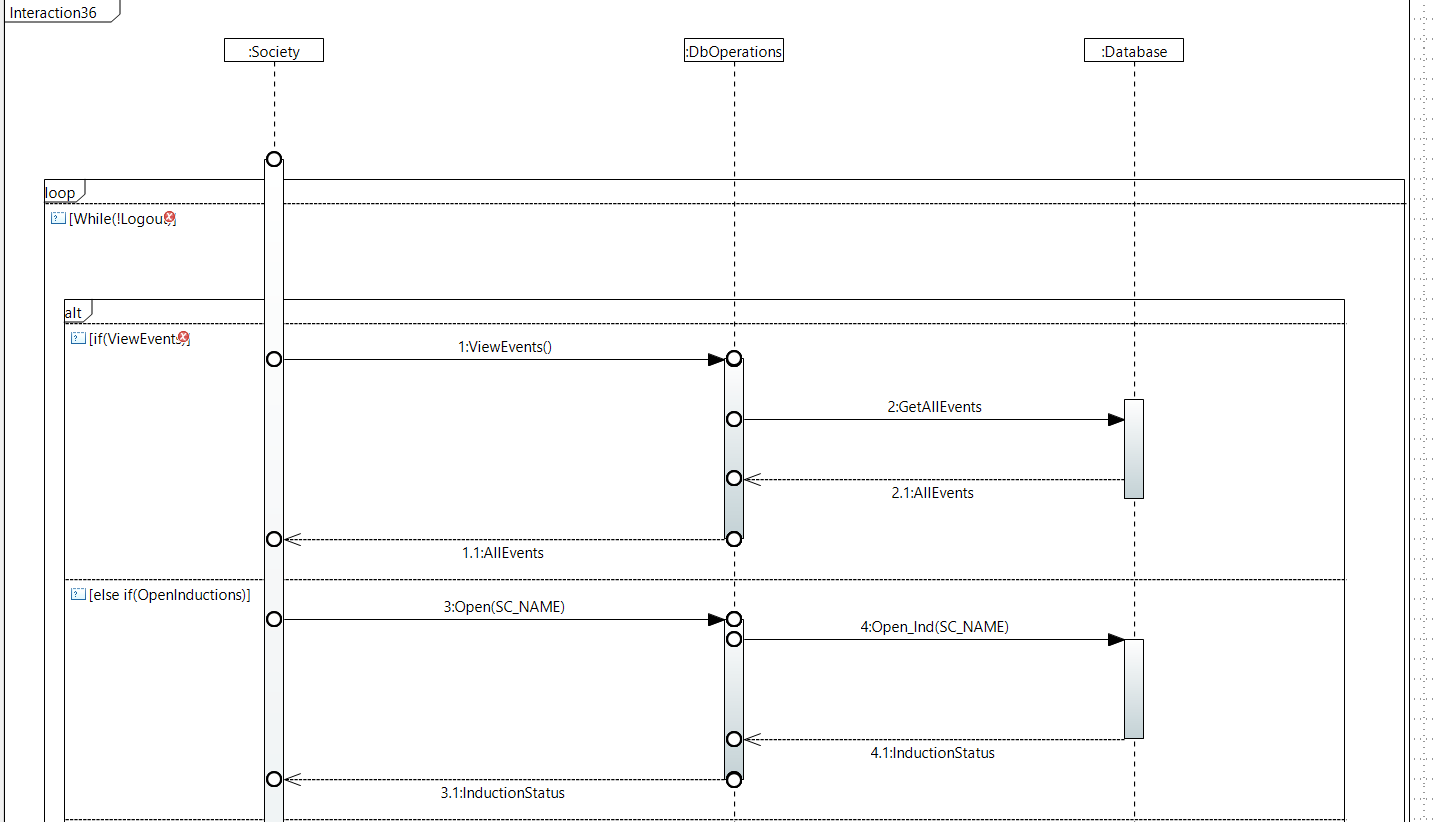
##### ADMIN

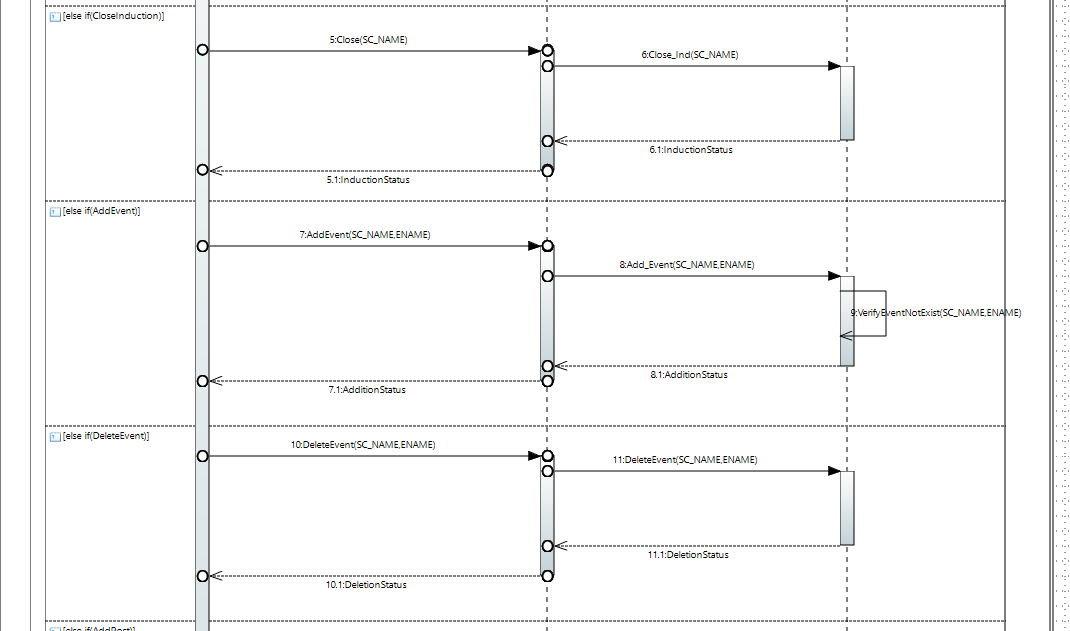
******

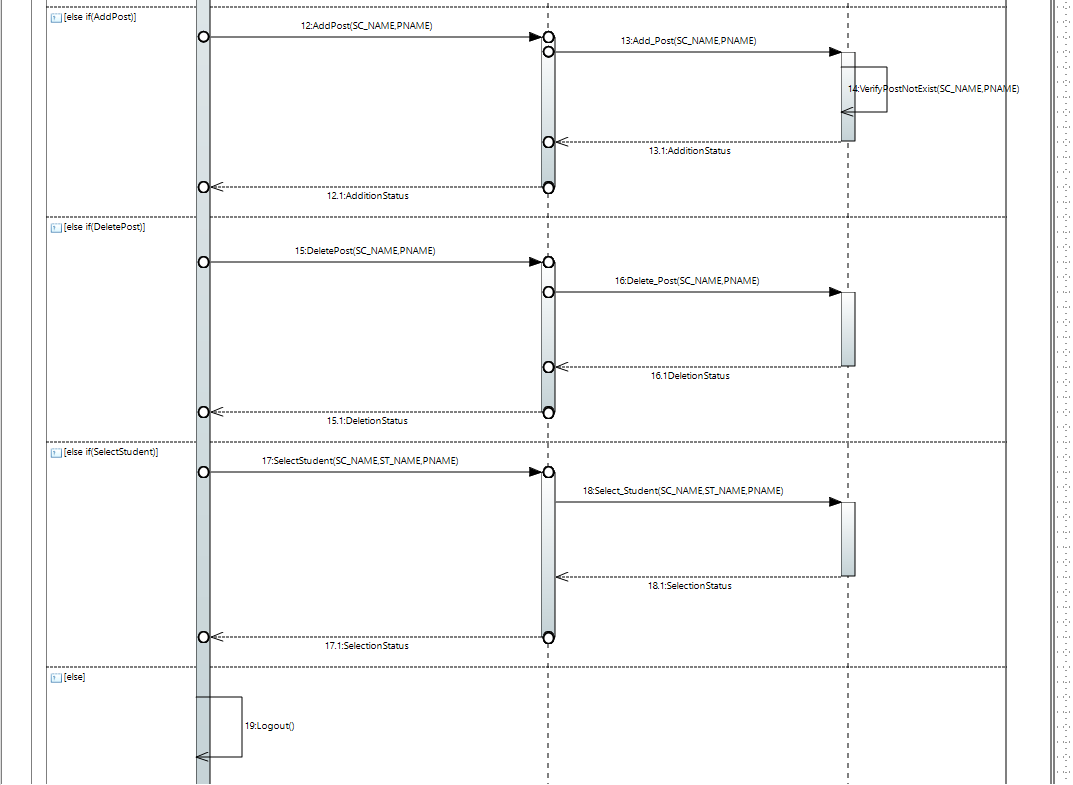
******

******

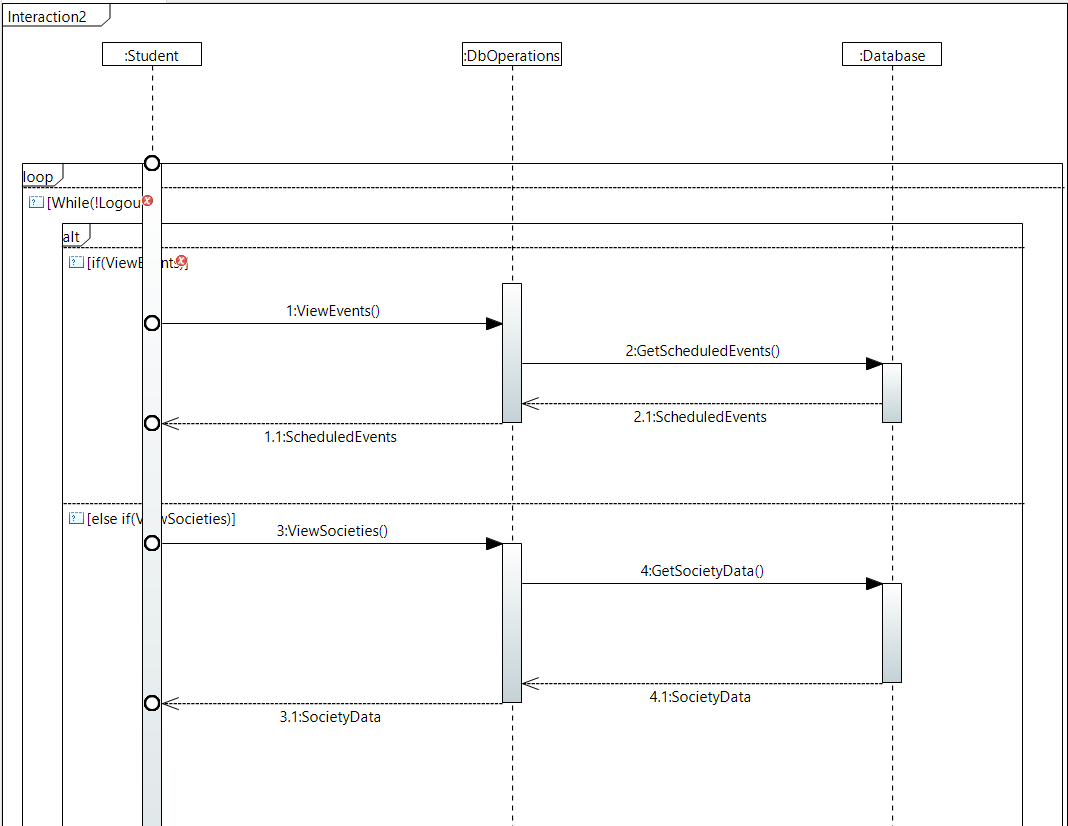
##### SOCIETY

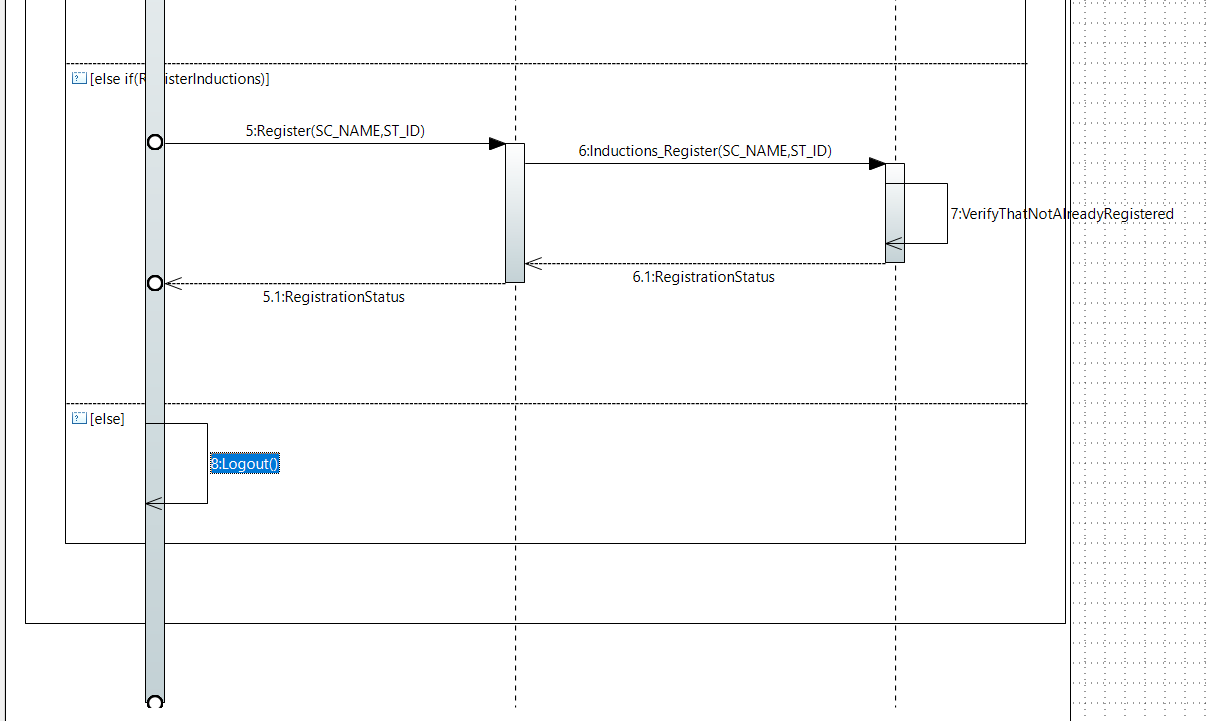
******

******

******

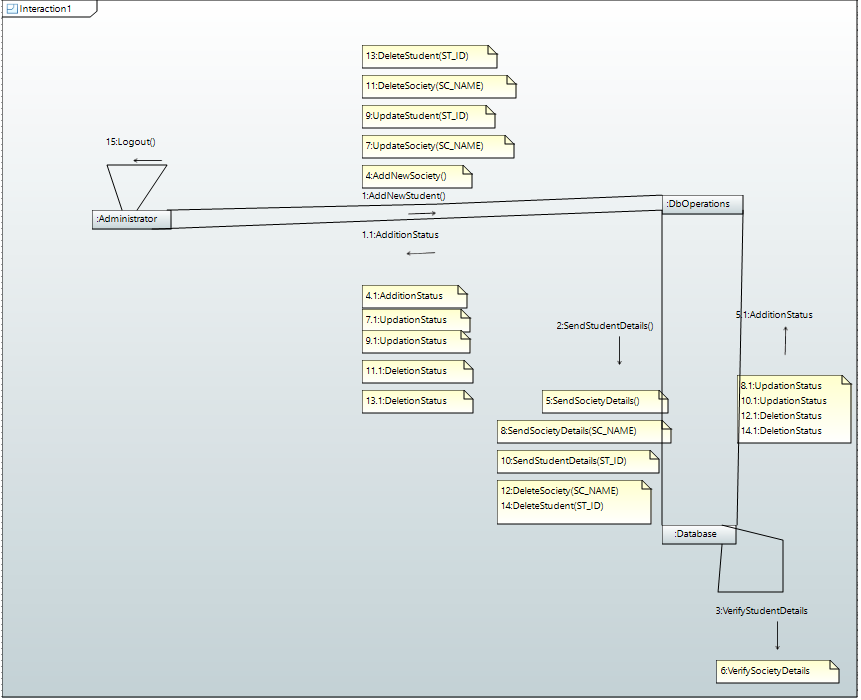
##### STUDENT

******

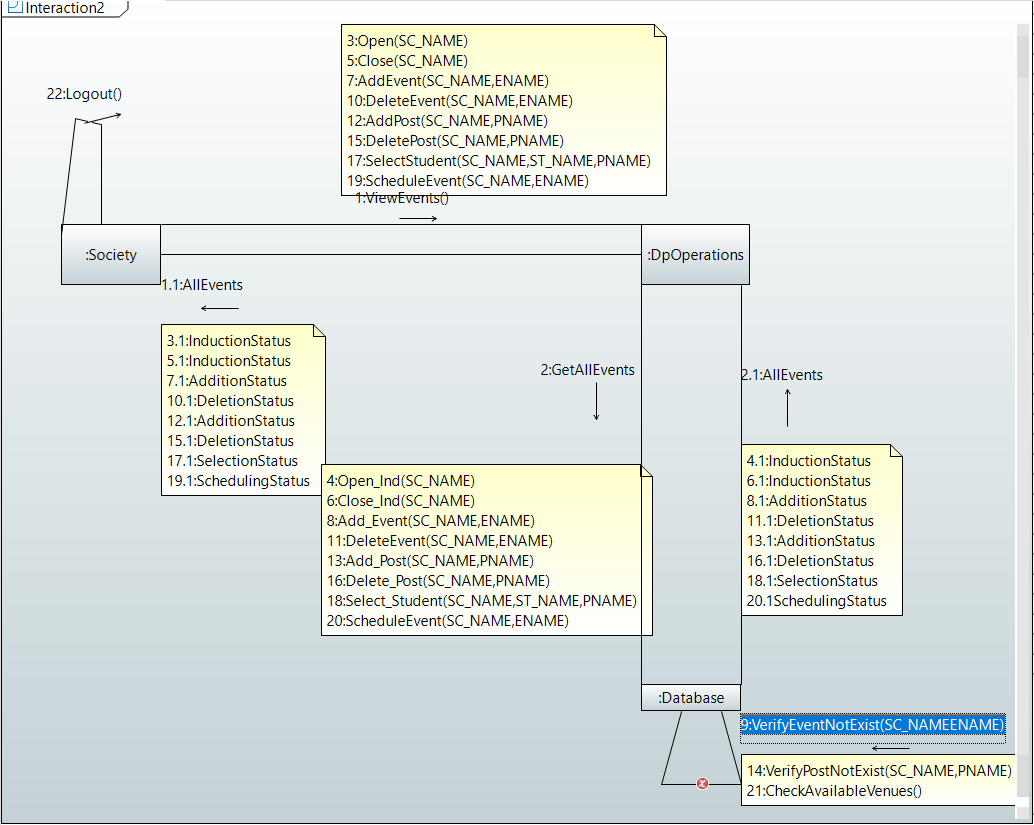
******

#### Communication Diagram

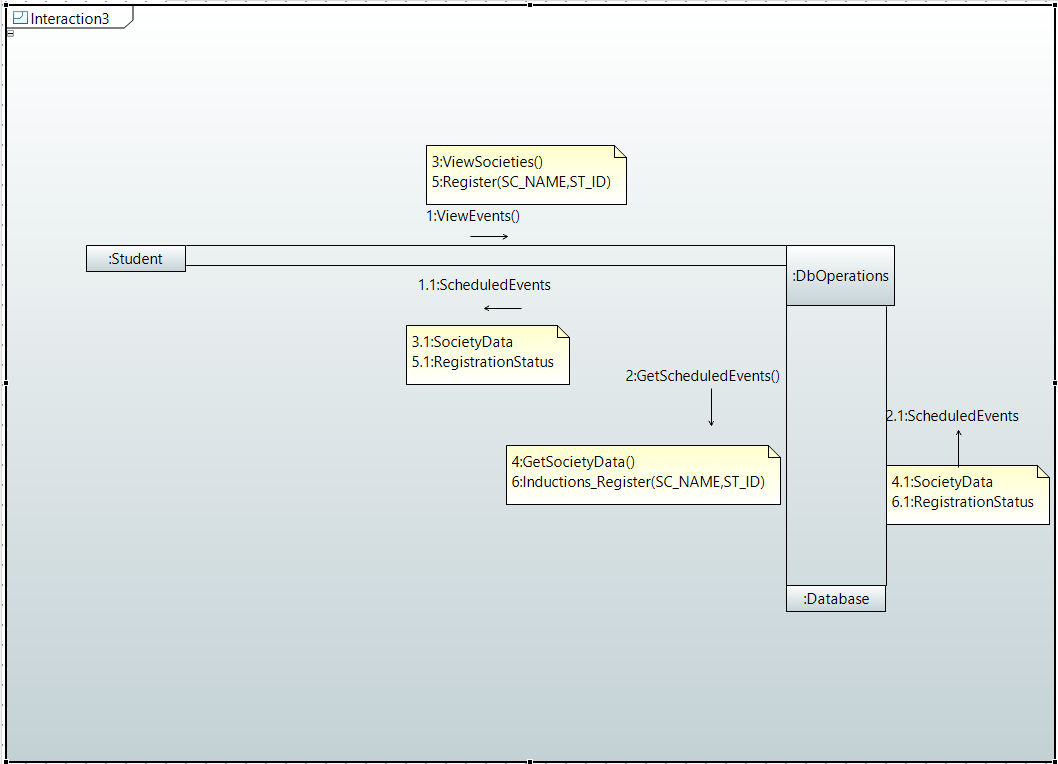
##### ADMIN

**

##### SOCIETY

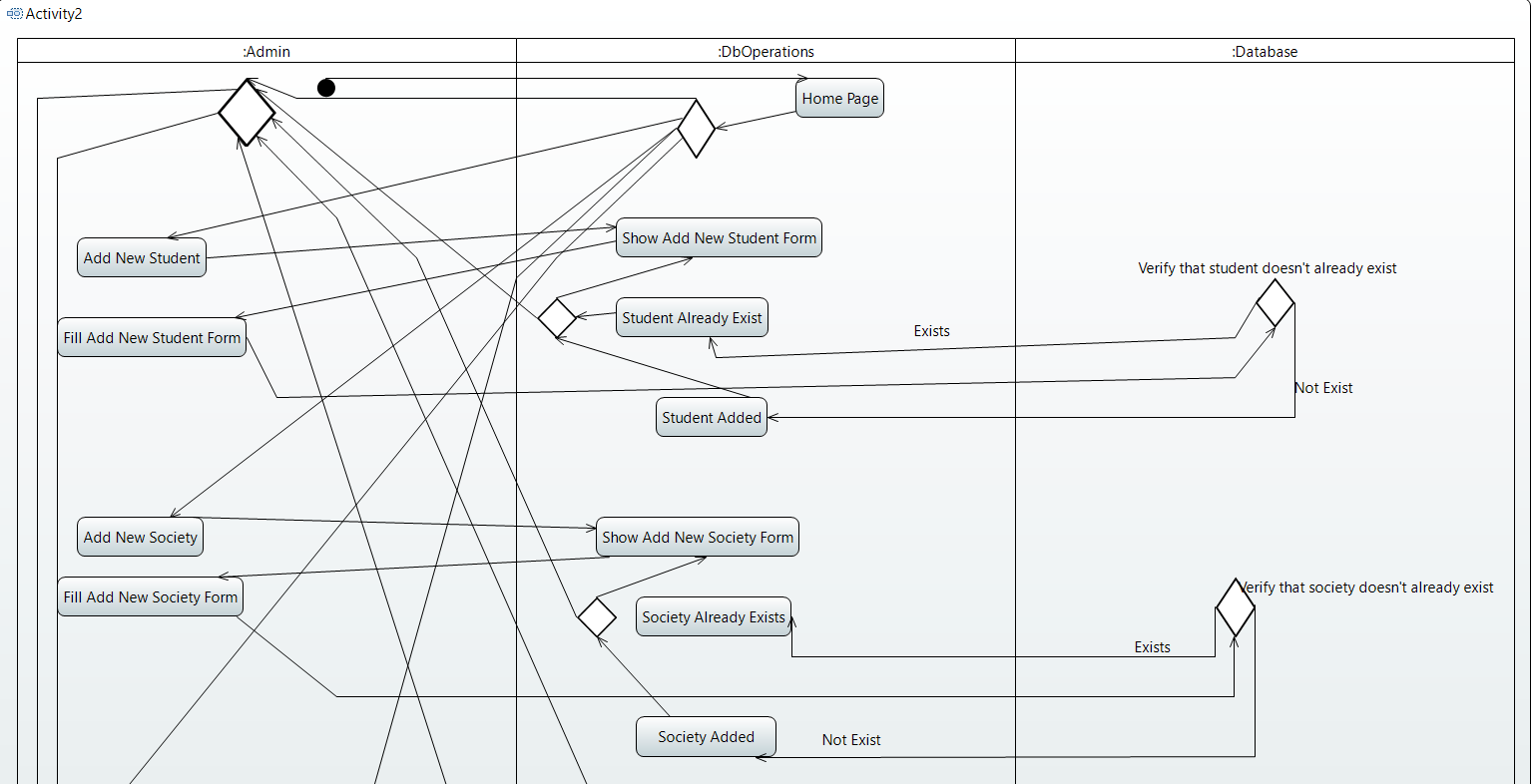
******

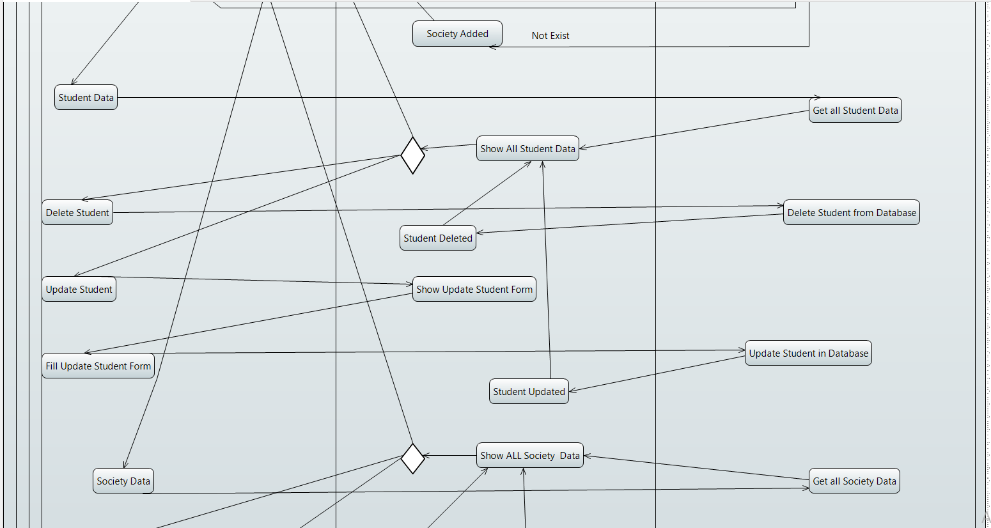
##### STUDENT

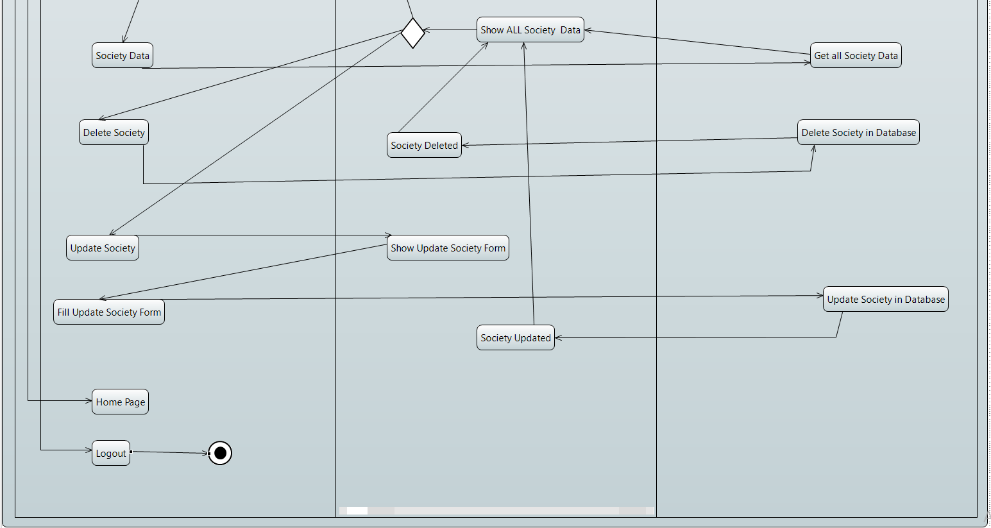
******

#### Activity Diagram

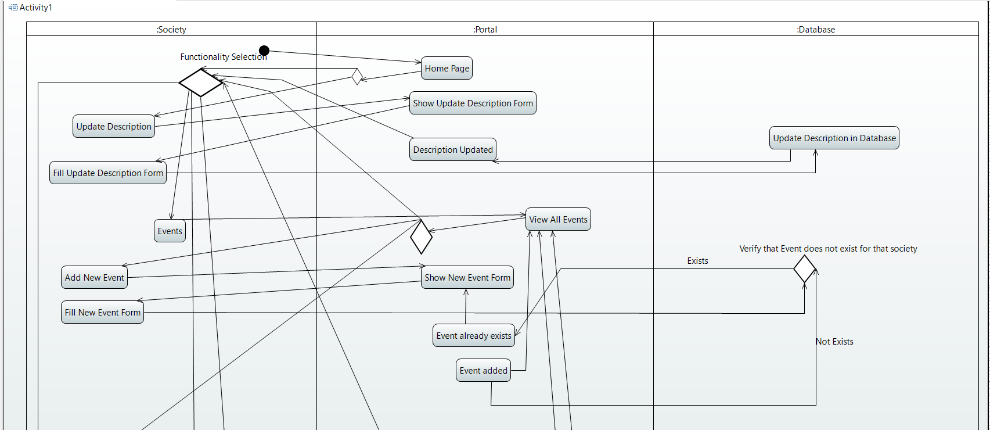
##### ADMIN

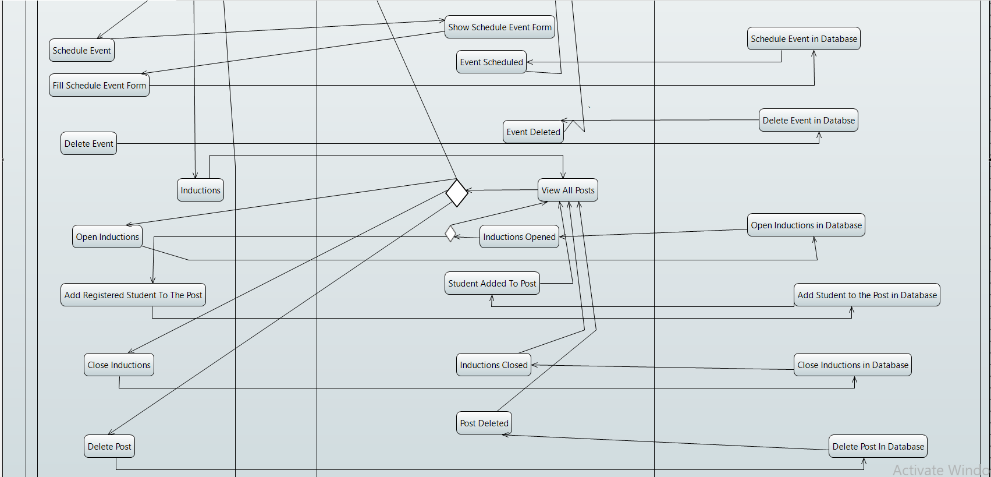
******

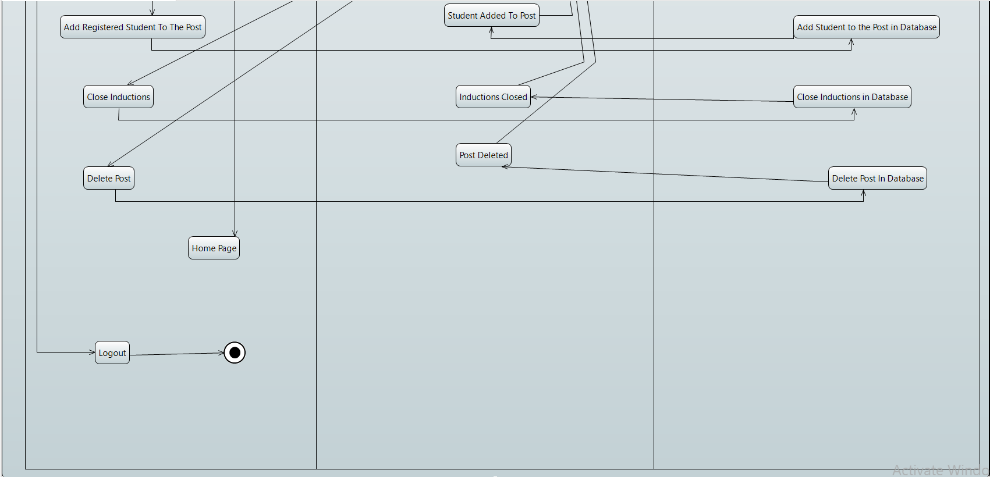
******

******

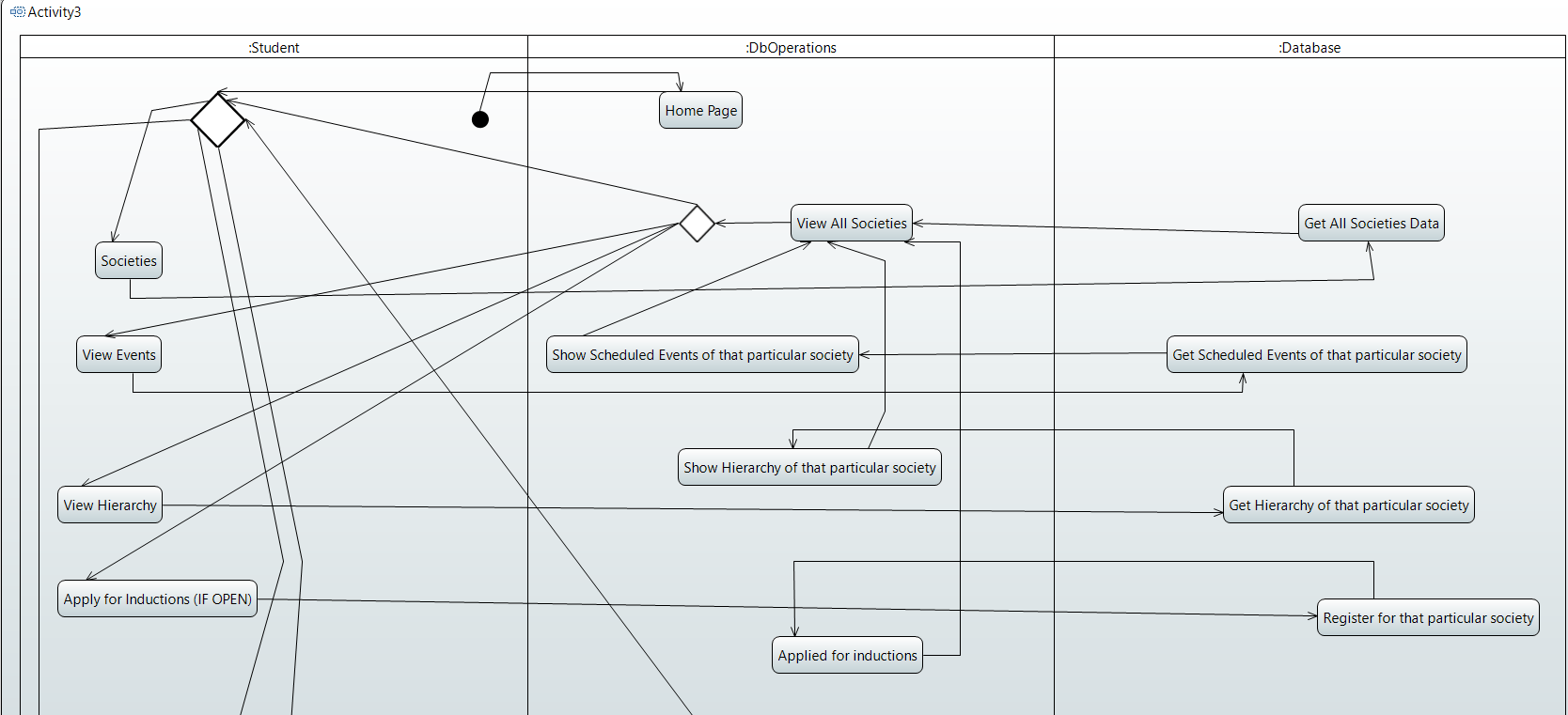
##### SOCIETY

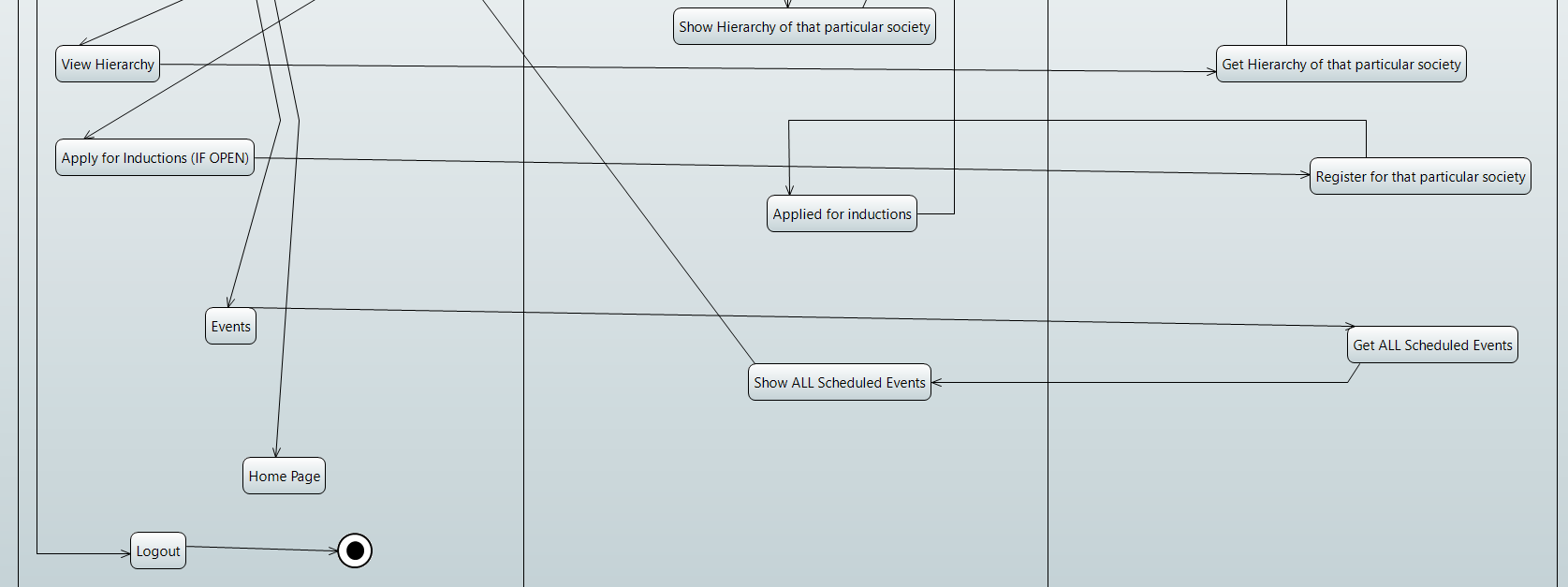
******

******

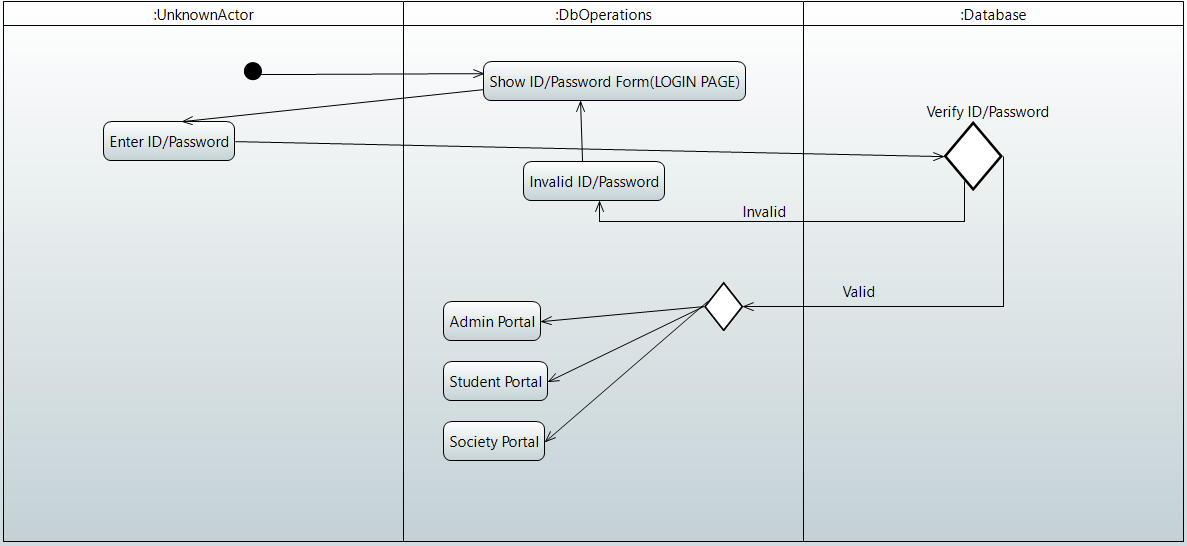
******

##### STUDENT

******

******

##### LOGIN

******

#### State Diagram

