



---

# EVENT MANAGEMENT SYSTEM

---

Documentation



NOVEMBER 17, 2016  
INSTITUTE OF INFORMATION TECHNOLOGY  
University of Dhaka

**Submitted to**  
Amit Seal Ami  
Lecturer  
Institute of Information Technology  
University of Dhaka

**Submitted by**  
**MIS Group – 6**

Moshiur Rahman Sadik (BSSE 0606)

Sakib Rezoan Reza (BSSE 0611)

Md. Ruhul Amin Rahat (BSSE 0616)

Feroz Ahmmed (BSSE 0618)

Md. Nazmul Haque (BSSE 0635)

**Submission date**

May 3, 2016

# Letter of Transmittal

May 3, 2016  
Amit Seal Ami  
Lecturer  
Institute of Information Technology  
University of Dhaka

Dear Sir,

I have prepared the report on Software Requirements Specification of '**Event Management System**' for your approval. This report details the requirements I gathered for the project.

The primary purpose of this report is to summarize my findings from the work that I completed as my Software Requirements Specification and Analysis course project. This report includes the details of each step I followed to collect the requirements excusable point.

Your sincerely

Md. Feroz Ahmmed

BSSE 0618

# Contents

Chapter 1: Introduction .....	7
1.1 Purpose.....	8
1.2 Intended Audience .....	8
1.3 Scope.....	9
1.4 Rational .....	9
Chapter 2: Inception.....	10
2.1 Introduction .....	11
2.1.1 Identifying Stakeholders .....	11
2.1.2 Asking the First Question .....	12
2.1.3 Recognizing Multiple Viewpoints .....	12
2.1.3 Working towards Collaboration.....	13
2.2 Conclusion .....	14
Chapter 3: Elicitation .....	15
3.1 Introduction.....	16
3.2 Eliciting Requirements.....	16
3.3 Collaborative Requirements Gathering.....	16
3.4 Quality Function Deployment.....	17
3.4.1 Normal Requirements .....	17
3.4.2 Expected Requirements.....	17
3.4.3 Exciting Requirements .....	18
3.5 Functional and Non-Functional Requirements .....	18
3.5.1 Functional Requirements .....	18
3.5.2 Non-Functional Requirements .....	18
3.6 Usage Scenarios .....	19

3.7 Elicitation Work Product .....	19
Chapter 4: Scenario-Based Model .....	20
4.1 Introduction .....	21
4.2 Use Case Scenario .....	22
4.3 Use Case Description .....	23
4.3.1 Authentication .....	25
4.3.2 Event .....	37
4.3.3 Sub event .....	41
4.3.4 Template .....	43
4.3.5 sub template of template .....	49
Chapter 5: Data Model .....	51
5.1 Introduction .....	52
5.2 Data Object Selection .....	52
5.3 Data Objects and Attributes .....	53
5.4 Relationship Between Data Objects .....	55
5.5 Entity Relationship(ER) Diagram .....	56
Chapter 6: Class Based Modeling .....	57
6.1 Introduction .....	58
6.2 General Classifications .....	58
6.3 Selection Criteria .....	59
6.4 Attribute Selection .....	59
6.5 Method Identification .....	60
6.6 Class Card .....	61
6.7 CRC Diagram .....	65
Chapter 7: Flow Oriented Model .....	66
7.1 Introduction .....	67
7.2 Data Flow Diagram (DFD) .....	67
Chapter 8: Behavioral Model .....	70
8.1 Introduction .....	71
8.2 Identifying Events .....	71
8.3 State Transition Diagram .....	72
8.4 Sequence Diagram .....	75
Chapter 9: Conclusion .....	76
References .....	78

## Figures

Figure 1:Use Case Diagram of EMS (Level-0) .....	23
Figure 2:Use Case Diagram of EMS (Level-1) .....	24
Figure 3: Use Case Diagram of Authentication (Level-1.1) .....	25
Figure 4: Use Case Diagram of EMS (Level-1.3) .....	26
Figure 5: Activity Diagram – Level 1.1.1 (Sign Up) .....	27
Figure 6: Swim Lane Diagram – Level 1.1.1 (Sign Up) .....	28
Figure 7: : Activity Diagram –Level 1.1.2 (Sign in) .....	29
Figure 8: : Swim Lane Diagram –Level 1.1.2 (Sign in) .....	30
Figure 9: Activity Diagram – Level 1.1.3 (Sign out) .....	31
Figure 10: Swim Lane Diagram – Level 1.1.3 (Sign out) .....	32
Figure 11: Activity Diagram – Level 1.1.4 (Change Password) .....	33
Figure 12:Swim Lane Diagram – Level 1.1.4 (Change Password) .....	34
Figure 13: Activity Diagram – Level 1.1.5 (Create User) .....	35
Figure 14: Swim Lane Diagram – Level 1.1.5 (Create User) .....	36
Figure 15: Activity Diagram – Level 1.2.1 (Create Event) .....	37
Figure 16: Swim Lane Diagram – Level 1.2.1 (Create Event) .....	38
Figure 17: Activity Diagram – Level 1.2.2 (Update Event) .....	39
Figure 18: Swim Lane Diagram – Level 1.2.2 (Update Event) .....	40
Figure 19: Activity Diagram – Level 1.2.3 (Sub Event) .....	41
Figure 20: Swim Lane Diagram – Level 1.2.3 .....	42
Figure 21: Activity Diagram – Level 1.3.1 (Create Template) .....	43
Figure 22: Swim Lane Diagram – Level 1.3.1 (Create Template) .....	44
Figure 23: Activity Diagram – Level 1.3.2 (Update Template) .....	45
Figure 24: Swim Lane Diagram – Level 1.3.2 (Update Template) .....	46
Figure 25: Activity Diagram – Level 1.3.3 (Use Template) .....	47
Figure 26: Swim Lane Diagram – Level 1.3.3 (Use Template) .....	48
Figure 27: Activity Diagram – Level 1.3.4 (Sub Event) .....	49
Figure 28: Swim Lane Diagram – Level 1.3.4 (Sub Event) .....	50
Figure 29: DFD ( Level-0) .....	67
Figure 30: DFD (Level- 1) .....	68
Figure 31: DFD ( Level-1.1-Authentication) .....	69
Figure 32: DFD ( Level-1.2-Event) .....	69
Figure 33: State transition diagram of Admin .....	72
Figure 34: : State transition diagram of Teacher .....	73
Figure 35: : State transition diagram of Student .....	74
Figure 36: Sequence Diagram .....	75

## Acronyms

Acronym	Definition
IIT	Institute of Information Technology
SRS	Software Requirement Specification
EMS	Event Management System
QFD	Quality Function Deployment
ER	Entity Relationship
CRC	Class Responsibility Collaboration
DFD	Data Flow Diagram

# Chapter 1: Introduction



## 1.1 Purpose

This document is the Software Requirements Specification (SRS) for the Event Management System (EMS). It contains detailed functional, non-functional, and support requirements and establishes a requirements baseline for development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

## 1.2 Intended Audience

This SRS is intended for several audiences, including the customer, as well as the project manager, designers, developer and tester.

- The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- The project manager of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system.
- The designer will use this SRS as a basis for creating the system's design. The designer will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.
- The developer will use this SRS as a basis for developing the system's functionality. The developer will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.
- The tester will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the tester will run his tests on that software to ensure that the software fulfills the requirements documented in this SRS. The tester will again run his tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

## 1.3 Scope

This document will address a few key themes in terms of the Event Management System – A web based authenticated interface of creating new events and a notification system for running and upcoming events. We intend to consider only students and faculty members of Institute of Information Technology (IIT).

## 1.4 Rational

Institute of Information Technology have such a busy schedule that students and faculty members are always conscious about events and tasks. But this consciousness is not always enough to guarantee the participation in events. Some of the websites like Facebook have features of creating events. But most of this websites are not made for professional people. Again, IIT demanding to be the Highest Echelon of Software Engineering in Bangladesh needed a mass customized website to manage their official events. Thus the service of our system comes into being.

## Chapter 2: Inception

## 2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how does a software project get started and what is the scope and nature of the problem to be solved. The goal of the inception phase is to identify concurrence needs and conflict requirements among the stakeholders of a software project. To establish the groundwork I have worked with the following factors related to the inception phases:

- Identifying stakeholders
- Asking the first questions
- Recognizing multiple viewpoints
- Working towards collaboration

### 2.1.1 Identifying Stakeholders

Stakeholder refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. To identify the stakeholders, we consulted with my Honorable teacher Amit Seal Ami and asked him following questions:

- Who is paying for the project?
- Who will be using the project outcomes?
- Who gets to make the decisions about the project (if this is different from the money source)?
- Who has resources I need to get the project done?
- Whose work will my project affect? (During the project and also once the project is completed)?

Concluding thoughts on Stakeholders, I identified following stakeholders for Event Management System:

1. **Student:** The largest user group of the system. The whole project become useless without them. They will create events, get notifications from other events and build a strong community.
2. **Teacher:** Faculty members are an essential part of this project. They have direct influence over Event Management System. They are mainly part of all official events of IIT.

3. **IIT:** IIT will finance the project and it has some rules and regulation to maintain our system. We have to follow them strictly. Besides IIT will decide who will be admins for this system.

### 2.1.2 Asking the First Question

We set the first set of context-free questions focuses on the customer and other stakeholders, overall project goals and benefits. The questions are mentioned above. These questions helped us to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped us to gain a better understanding of problem and allowed the customer to voice his or her perception about the solution. The final set of question focused on the effectiveness of the communication activity itself.

### 2.1.3 Recognizing Multiple Viewpoints

We collected these viewpoints by discussing with some students, teachers and officials from IIT.

1. IIT's viewpoint:

- No disruption of rules and regulation
- Highly reliable system minimum cost
- Tools and technologies are open source

2. Student's viewpoint:

- User friendly
- Web-Based Interfaces
- Option to see other's activity
- Supports mass customized template to reuse

3. Teacher's viewpoint:

- Web-Based Interfaces
- Maintain a database of all items in this system
- Strong Authentication
- User friendly
- Supports mass customized template to reuse

- Restrict access to functionality of the system based upon user roles. For example, only Administrators of the system will be provided functionality to change user types, configure how long items may be checked out

### 2.1.3 Working towards Collaboration

Every stakeholder has their own requirements. We followed following steps to merge these requirements:

- Identify the common and conflicting requirements
- Categorize the requirements
- Take priority points for each requirements from stakeholders and on the basis of this voting prioritize the requirements
- Make final decision about the requirements.

#### **Common requirements:**

- Web-Based Interfaces
- User friendly
- The application can be accessed from any computer that has Internet access
- Maintain a database of all information in this system
- Supports mass customized template to reuse

#### **Conflicting Requirements:**

We found some requirements conflicting each other. We had to trade-off between the requirements.

- Easy access and Strong Authentication
- Allow any user to use the system and restrict access

#### **Final Requirements:**

We finalized following requirements for the system by categorizing and prioritizing the requirements:

- Error free system (Maximum 5% error may be considerable)

- Web-based interfaces
- Accessible via the Internet.
- Maintain a database of all information in this system.
- Allow valid users to login and logout.
- Restrict access to functionality of the system based upon user roles
- Allow administrators of the system to change user types and configure parameters of the system
- Allow valid users that log in to see notification
- Restrict access to functionality of the system based upon user roles. For example, only Administrators of the system will be provided functionality to change user types and update database.

## 2.2 Conclusion

Inception phase helped us to establish basic understanding about Event Management System, identify the people who will be benefited if Event Management System establishes, define the nature of the Event Management System and establish a preliminary communication with stakeholders.

## Chapter 3: Elicitation



### 3.1 Introduction

Elicitation is a task that helps the customer to define what is required. To complete the elicitation step we faced many problems like problems of scope, problems of volatility and problems of understanding. However, this is not an easy task. To help overcome these problems, we have worked with the Eliciting requirements activity in an organized and systematic manner.

### 3.2 Eliciting Requirements

Unlike inception where Q&A (Question and Answer) approach is used, elicitation makes use of a requirements elicitation format that combines the elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements, we completed following four works.

1. Collaborative requirements gathering
2. Quality function deployment
3. Usage scenarios
4. Elicitation work products

### 3.3 Collaborative Requirements Gathering

Many different approaches to collaborative requirements gathering have been proposed. Each makes use of a slightly different scenario. We completed following steps to do it.

- The meetings were conducted with our honorable teacher. He was questioned about his requirements and expectations from the Event Management System.
- We also had meetings along with our team members to reconsider the requirements.
- Teachers were asked about the problems they are facing with the current manual system.
- At last we selected the final requirement list from the meetings with our teacher.

## 3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. It concentrates on maximizing customer satisfaction from the Software engineering process. With respect to this project the following requirements are identified by a QFD.

### 3.4.1 Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of this project are: -

1. Accessible via the Internet.
2. Allow valid users to login and logout
3. Restrict access to functionality of the system based upon user roles.
4. Allow valid users to see notification.

### 3.4.2 Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them. Their absence will be a cause for dissatisfaction.

1. Maintain a database of all information in the Event Management System.
2. The system shall allow the user to sign in based upon email and password.
3. The user interface of the system shall be easy to use and shall make use of selectable fields wherever possible instead of fields that require the user to type in data.
4. Support mass customized template to reuse.

### 3.4.3 Exciting Requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present.

1. Even unauthenticated users can view a calendar with the information of public holidays.
2. The user interface should provide appropriate error messages for invalid input.
3. The user interface should follow standard web practices such that the web interface is consistent with typical internet applications.

## 3.5 Functional and Non-Functional Requirements

All requirements are classified into two parts – Functional and Non-Functional.

### 3.5.1 Functional Requirements

1. Accessible via the internet.
2. Allow only authenticated users to connect in this system.
3. The user interface should provide appropriate error messages for invalid input.

### 3.5.2 Non-Functional Requirements

1. Allow valid users to login and logout.
2. The system shall allow the user to sign in based upon email and password.
3. The user interface of the system shall be easy to use and shall make use of selectable fields wherever possible instead of fields that require the user to type in data.
4. The user interface should follow standard web practices such that the web interface is consistent with typical internet applications.
5. Support mass customized template to reuse.

## 3.6 Usage Scenarios

An Event Management System is a system that serves the functionalities related to an event. This is proposed to be a web based system. Only authenticated users can enter into the system.

Students are recommended to sign up to be a user. Sign up form includes name, roll, email and password. IITians are the only users of this system, so this system is required two step verification for students which are email and admin verification. No sign up is required for teachers as they will be added by system admins with their name, designation, email and password.

This system provides most of the basic functionalities required for an event. It allows any user to create an event with an event name, description, start date and duration. Event creator is basically the event manager who have the right to delete and update the event. Each event may contain multiple sub-events which will also be managed by the event manager. Sub events can be added by their name, description, start date and duration.

Event manager can turn off / on notification of own events, even for sub-events. It is allowed to change ownership of an event or even delete it.

To avoid repetitive creation of similar events admin can add some template events. These templates are reusable for creating further similar events. Templates will have all similar fields and sub-events like an event except start date and description. Use of templates have advantages like auto filled fields and auto added sub-events.

Members have choice to receive notification of events. Users will be notified for running and upcoming events. Every membership approval request will be sent to the event manager.

System admins have full control over the system.

## 3.7 Elicitation Work Product

The output of the elicitation task can vary depending on size of the system or product to be built.

Our elicitation work product includes:

- Make a statement of our requirements for Event Management System.
- Make a bounded statement of scope for our system.
- Make a list of customer, user and other stakeholder who participated in requirements elicitation.
- Set of usage scenarios.
- Description of the system's technical environment

## Chapter 4: Scenario-Based Model

## 4.1 Introduction

In this model the system is described from the user's point of view. As this is the first model, it serves as input for creation of other modeling elements.

## 4.2 Use Case Scenario

Table 4.1: Use Case Scenario

Level - 0	Level - 1	Level - 2	Actors
Event Management System	Authentication	Sign Up	Student
		Verification	Admin
		Sign In	Admin, Teacher, Student
		Sign Out	Admin, Teacher, Student
		Change Password	Admin, Teacher, Student
		Create User	Admin
		Remove User	Admin
	Event	Create event	Admin, Teacher, Student
		Update event	Admin, Teacher, Student
		Sub event	Admin, Teacher, Student
	Template	Create template	Admin
		Update template	Admin
		Use Template	Admin, Teacher, Student
		Sub event of template	Admin, Teacher, Student
	User activity		Admin, Teacher, Student

### 4.3 Use Case Description

We will elaborate use case scenario to use case diagram, description, activity diagram and swim - lane diagram. Here is the use case diagram of level-0 for Event Management System.

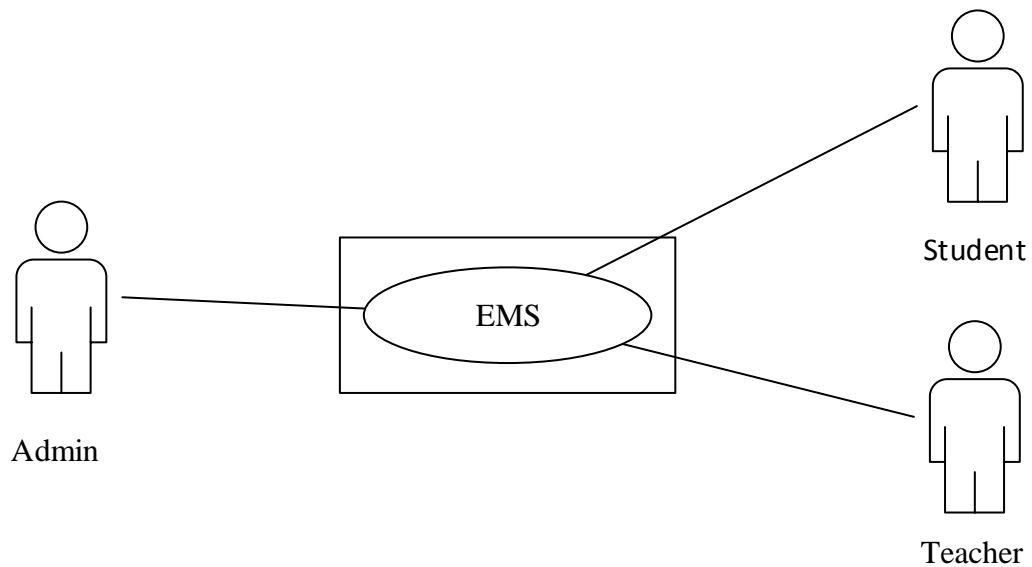


Figure 1:Use Case Diagram of EMS (Level-0)



This is the elaborated form of level-0 for Event Management System.

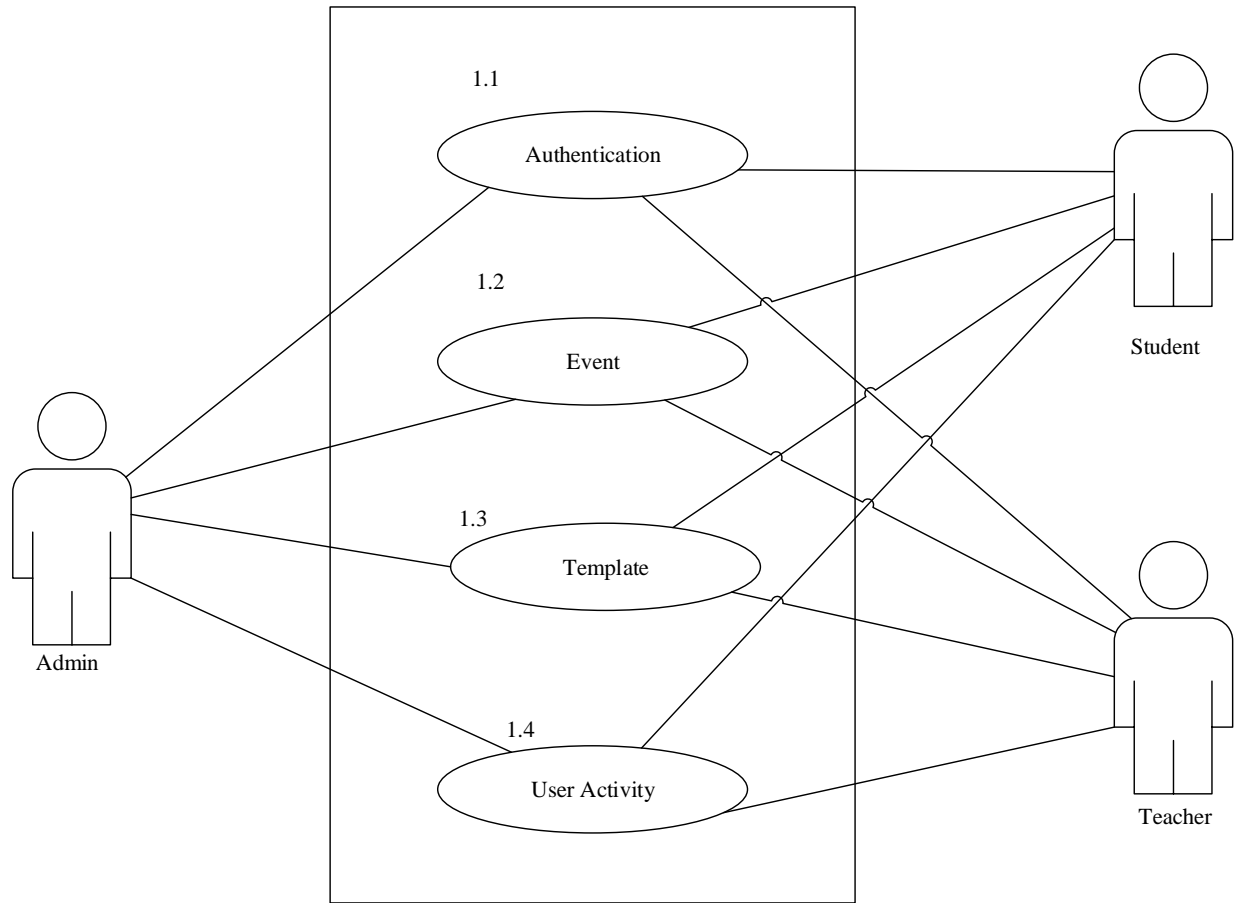


Figure 2:Use Case Diagram of EMS (Level-1)

### 4.3.1 Authentication

We can further section Authentication system into sub-systems.

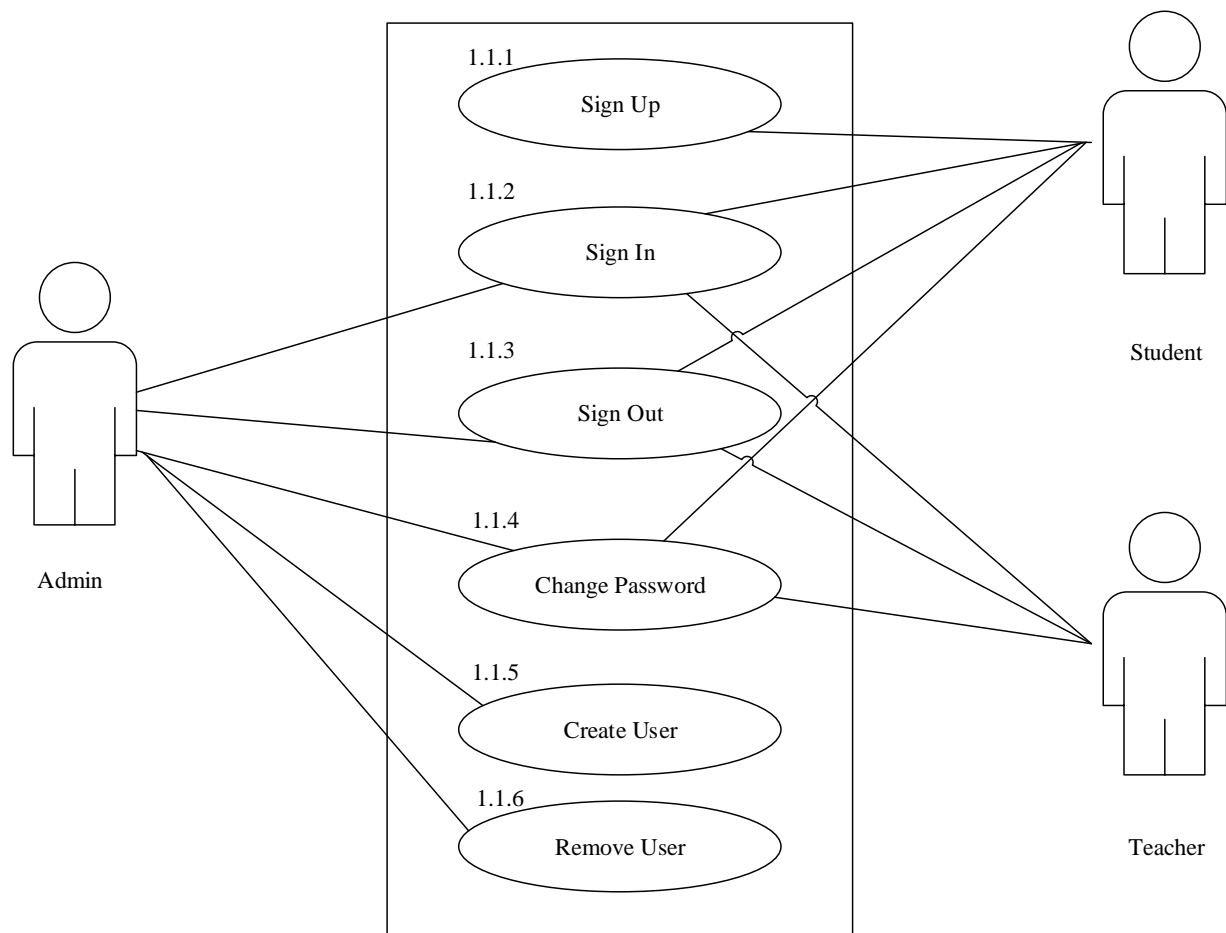


Figure 3: Use Case Diagram of Authentication (Level-1.1)

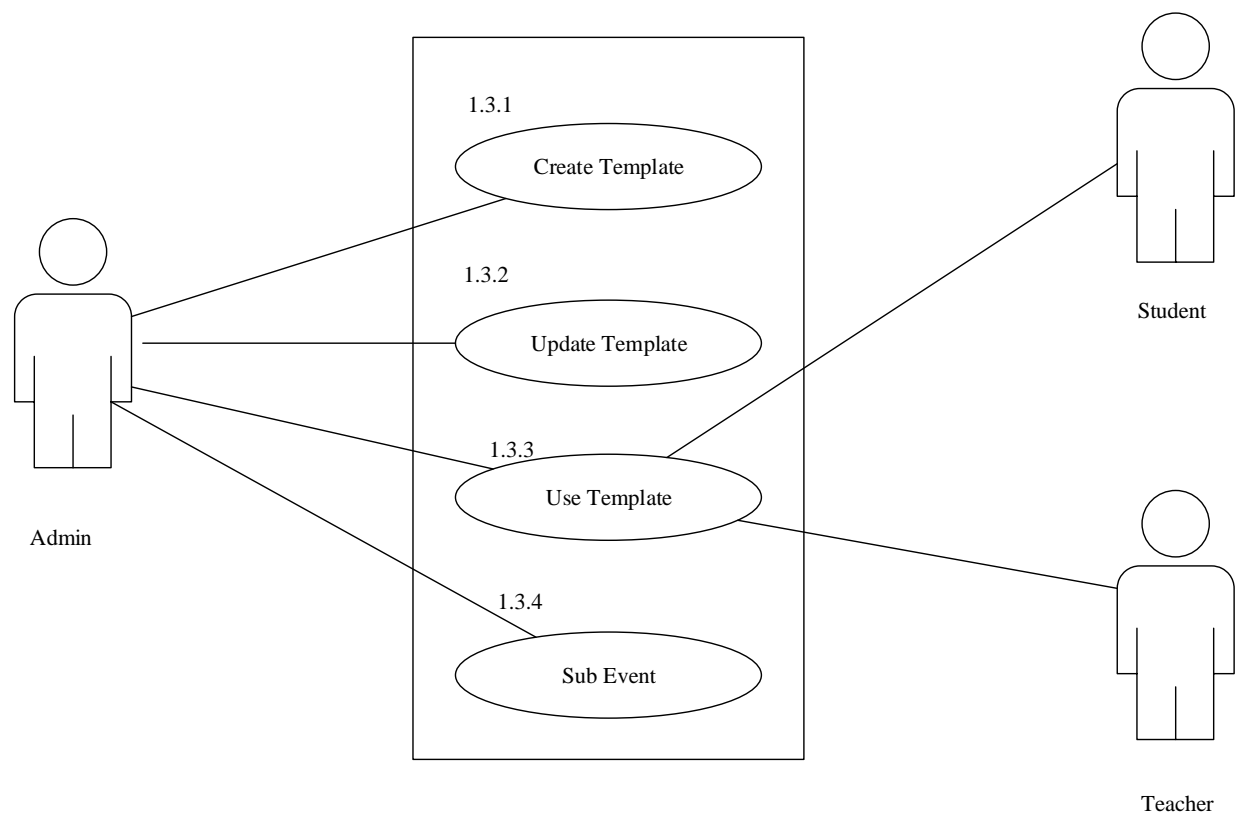


Figure 4: Use Case Diagram of EMS (Level-1.3)

#### 4.3.1.1 Sign up

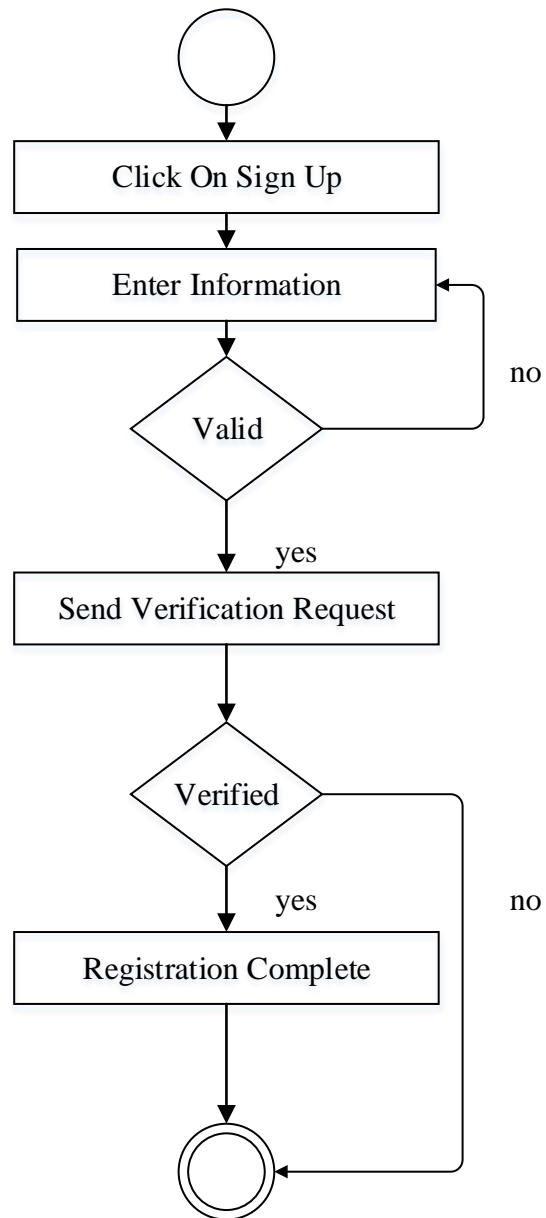


Figure 5: Activity Diagram – Level 1.1.1 (Sign Up)

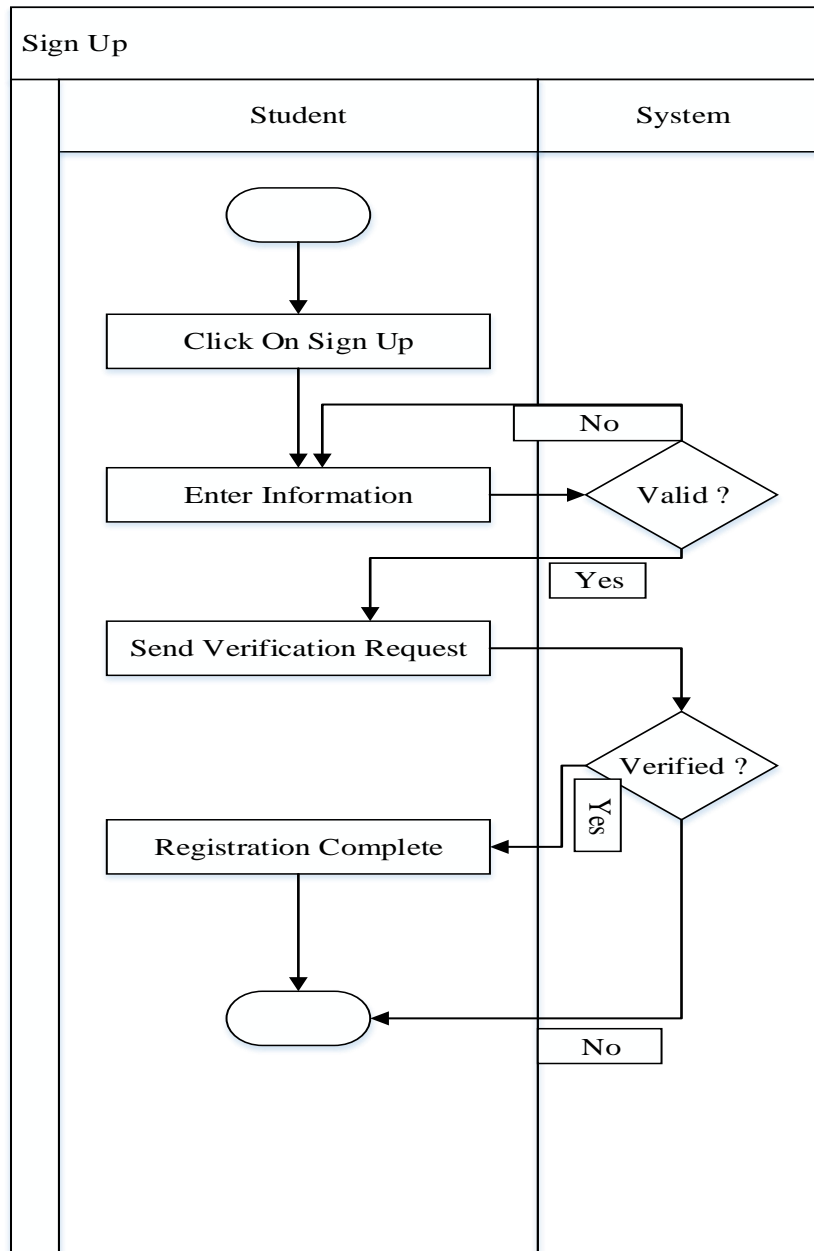


Figure 6: Swim Lane Diagram – Level 1.1.1 (Sign Up)

#### 4.3.1.2 Sign in

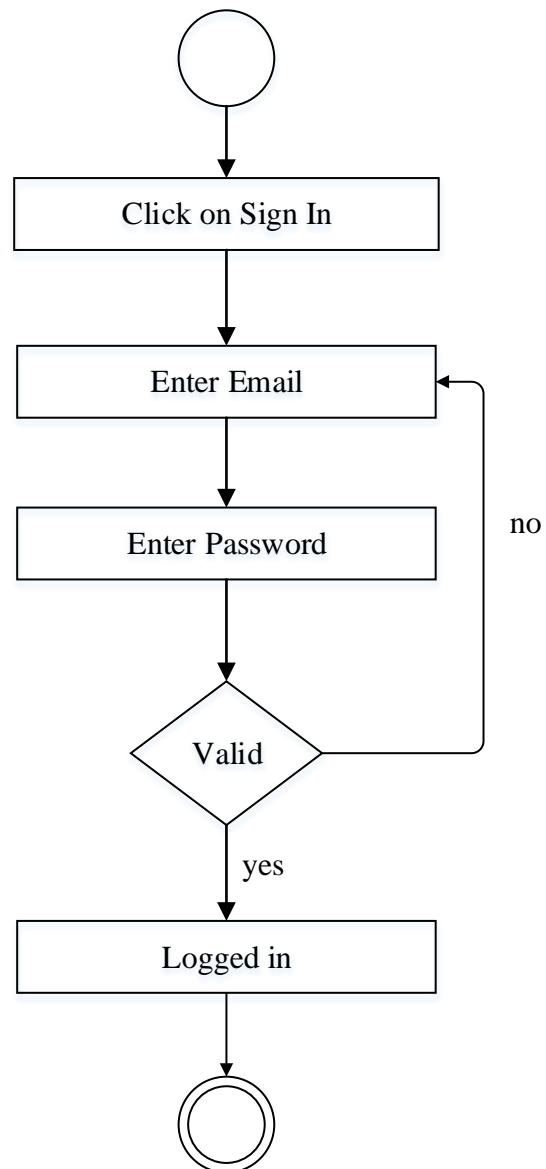


Figure 7: : Activity Diagram –Level 1.1.2 (Sign in)

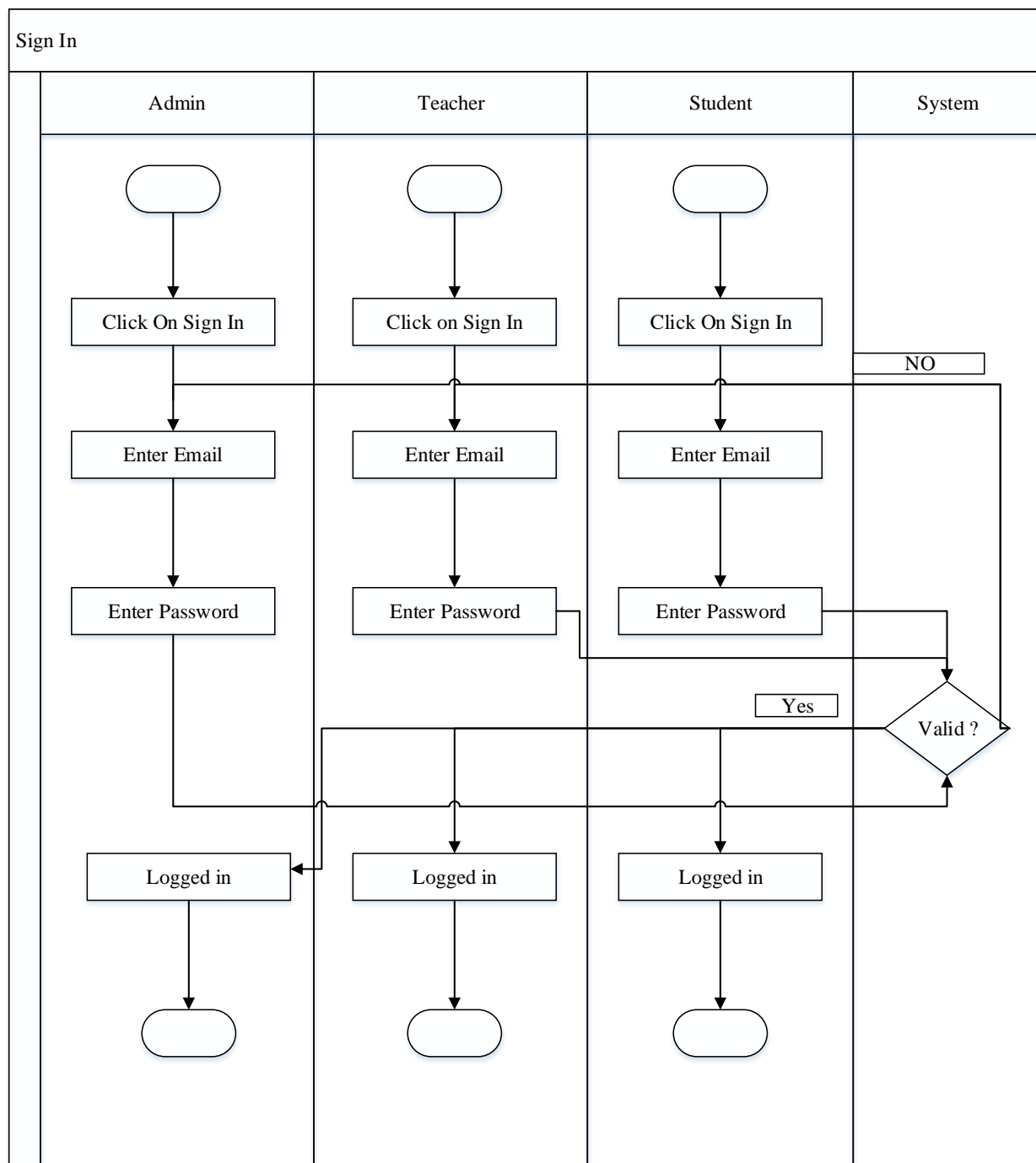


Figure 8: : Swim Lane Diagram –Level 1.1.2 (Sign in)

#### 4.3.1.3 Sign out

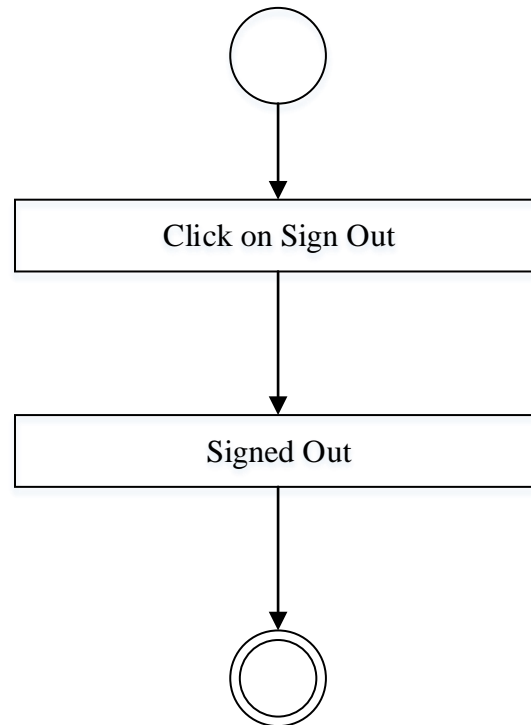


Figure 9: Activity Diagram – Level 1.1.3 (Sign out)



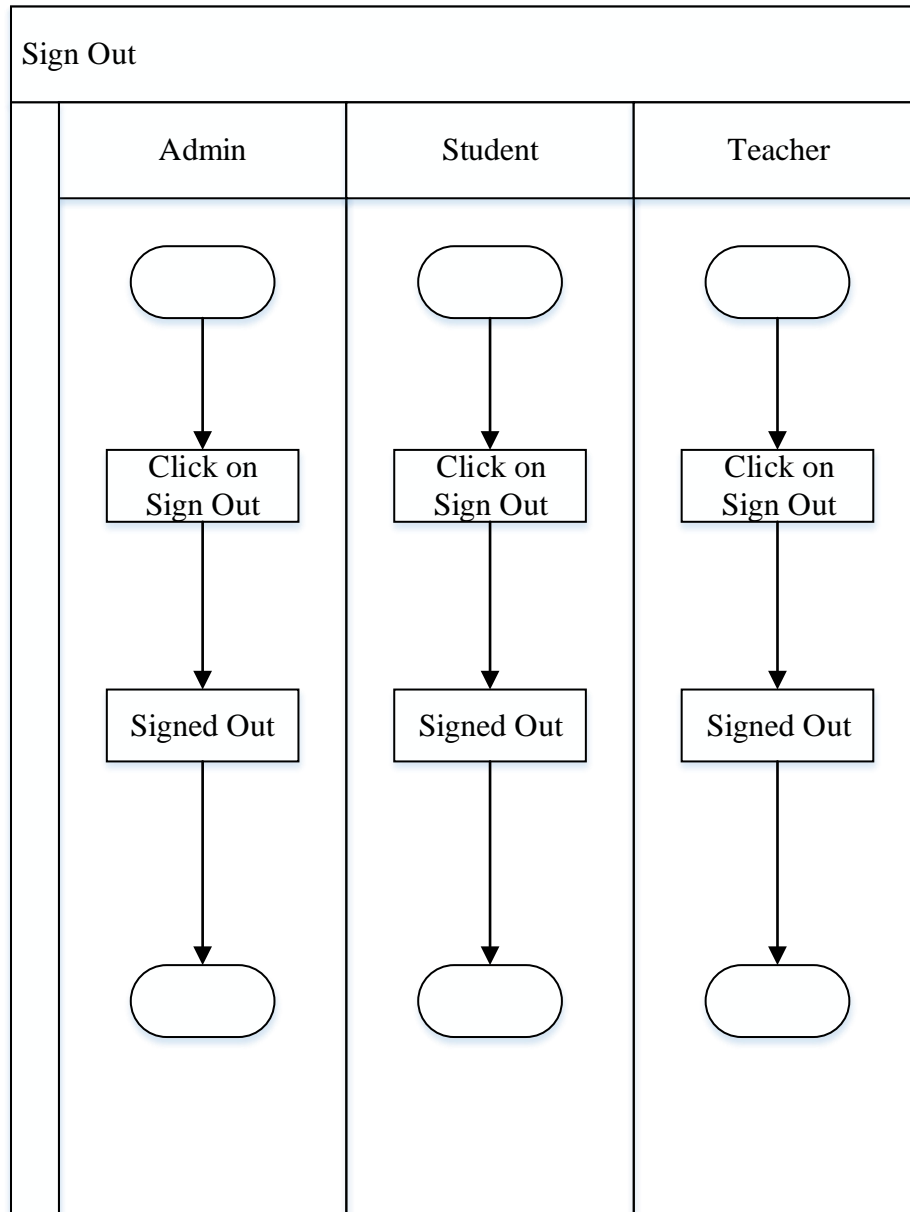


Figure 10: Swim Lane Diagram – Level 1.1.3 (Sign out)

#### 4.3.1.4 change password

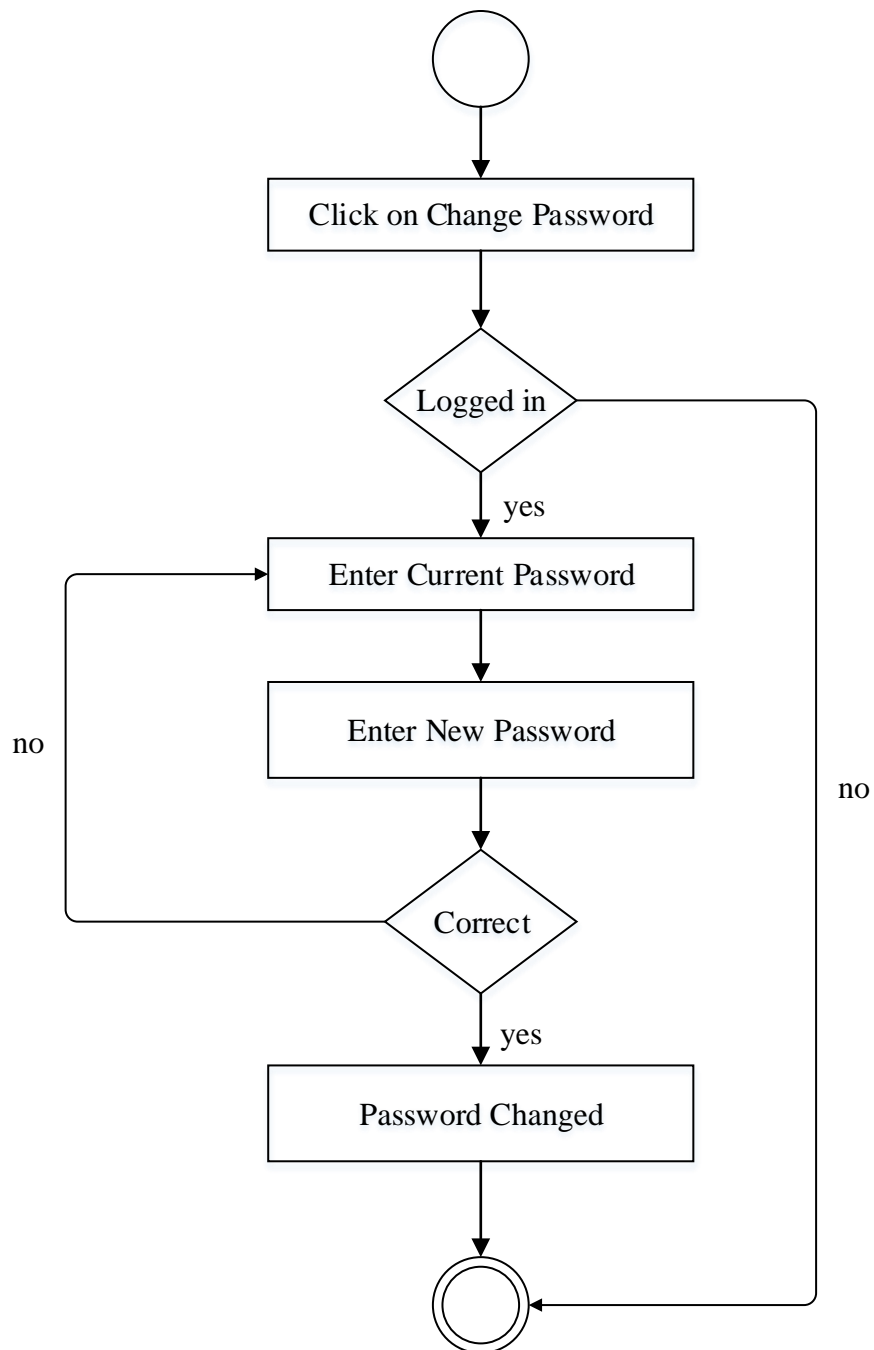


Figure 11: Activity Diagram – Level 1.1.4 (Change Password)

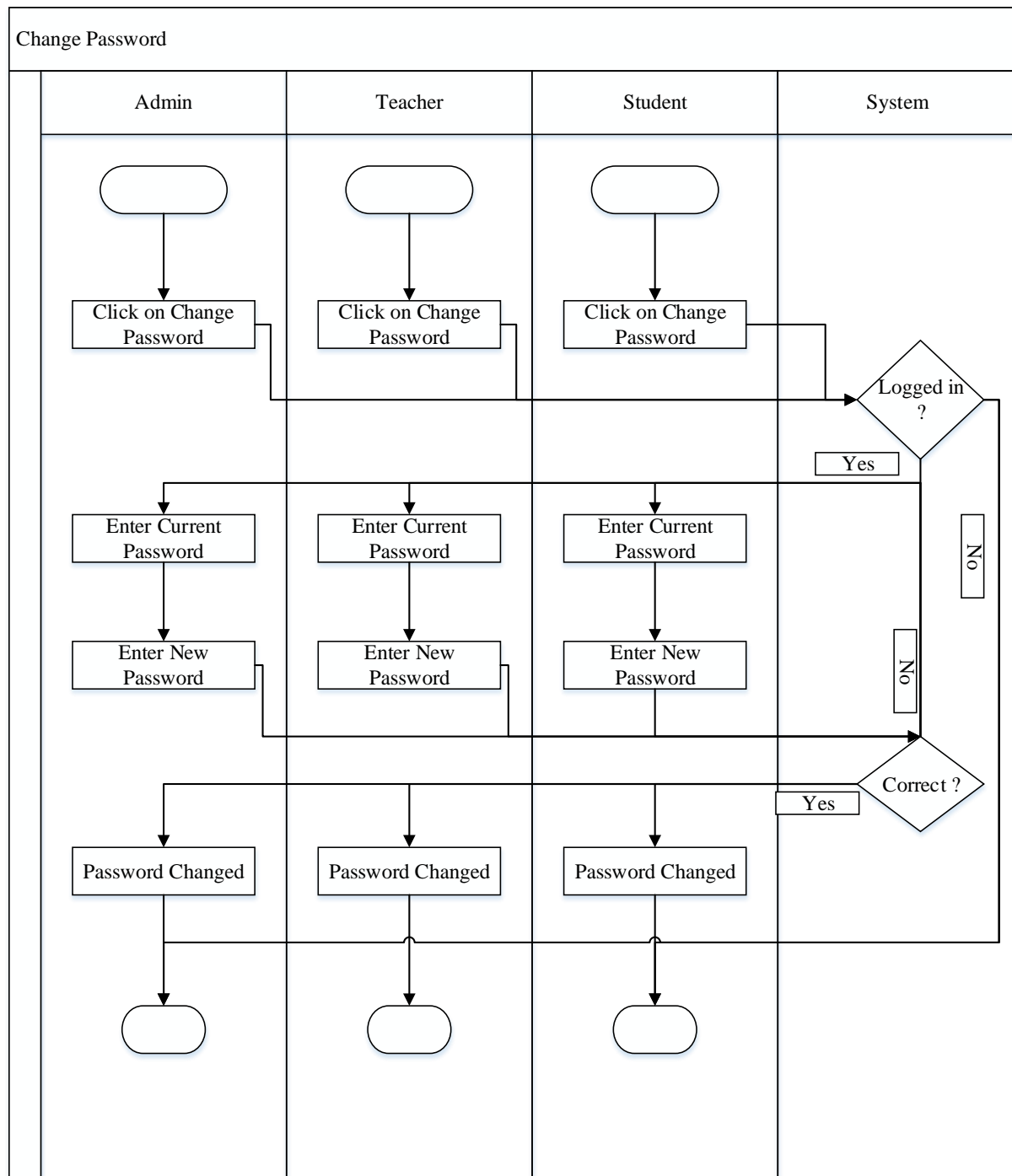


Figure 12:Swim Lane Diagram – Level 1.1.4 (Change Password)

#### 4.3.1.5 create user

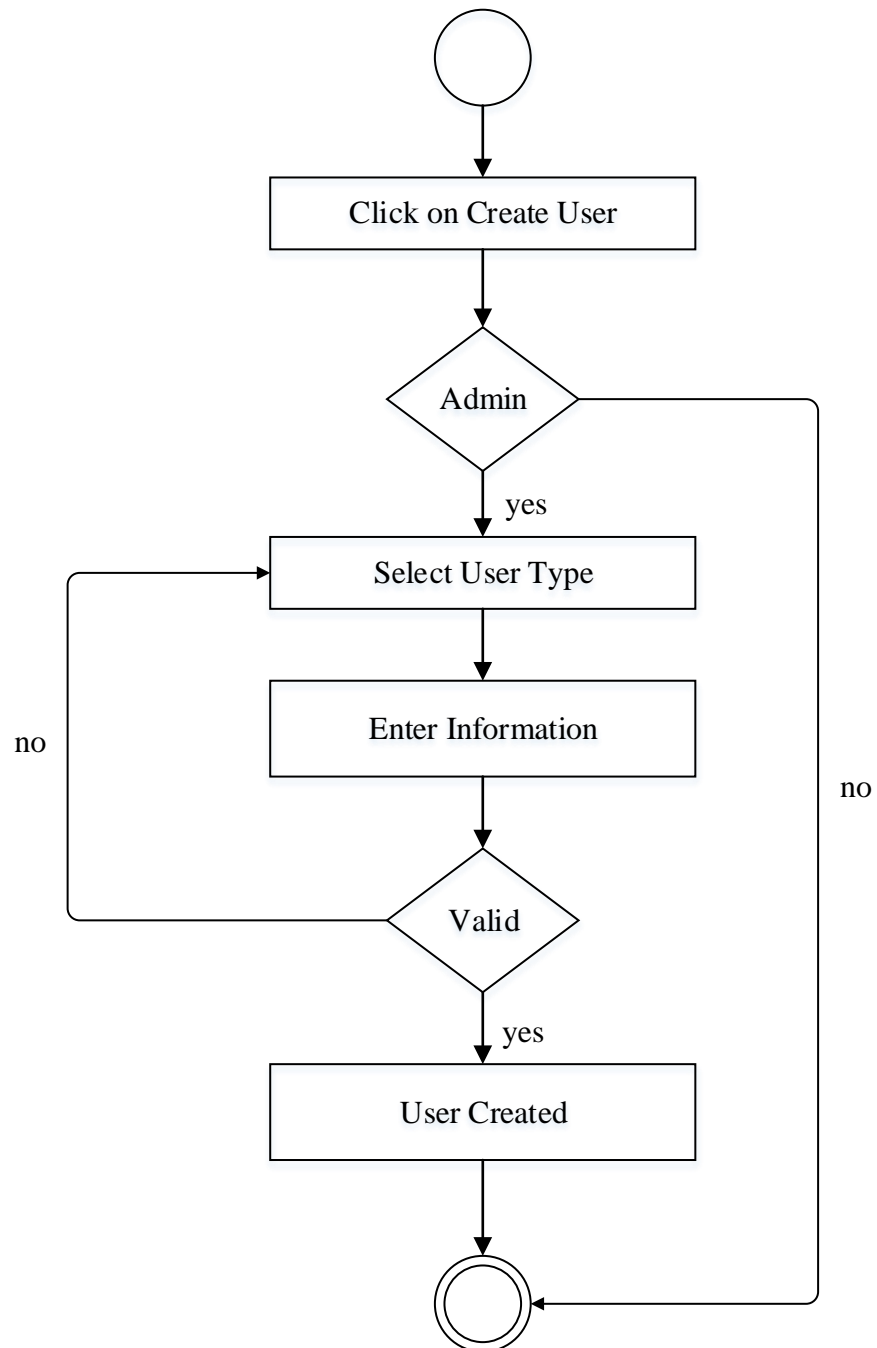


Figure 13: Activity Diagram – Level 1.1.5 (Create User)

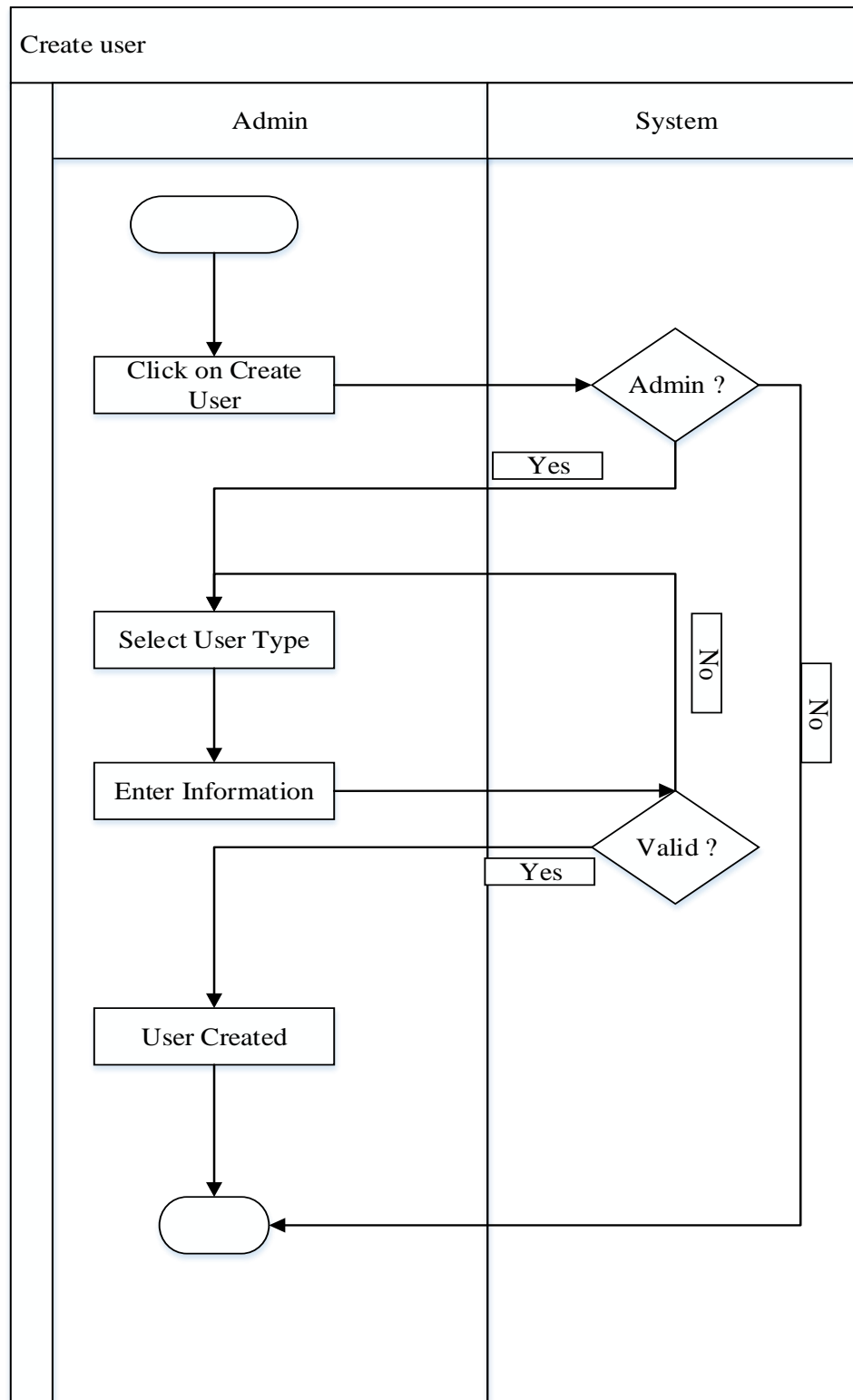


Figure 14: Swim Lane Diagram – Level 1.1.5 (Create User)

## 4.3.2 Event

### 4.3.2.1 Create Event

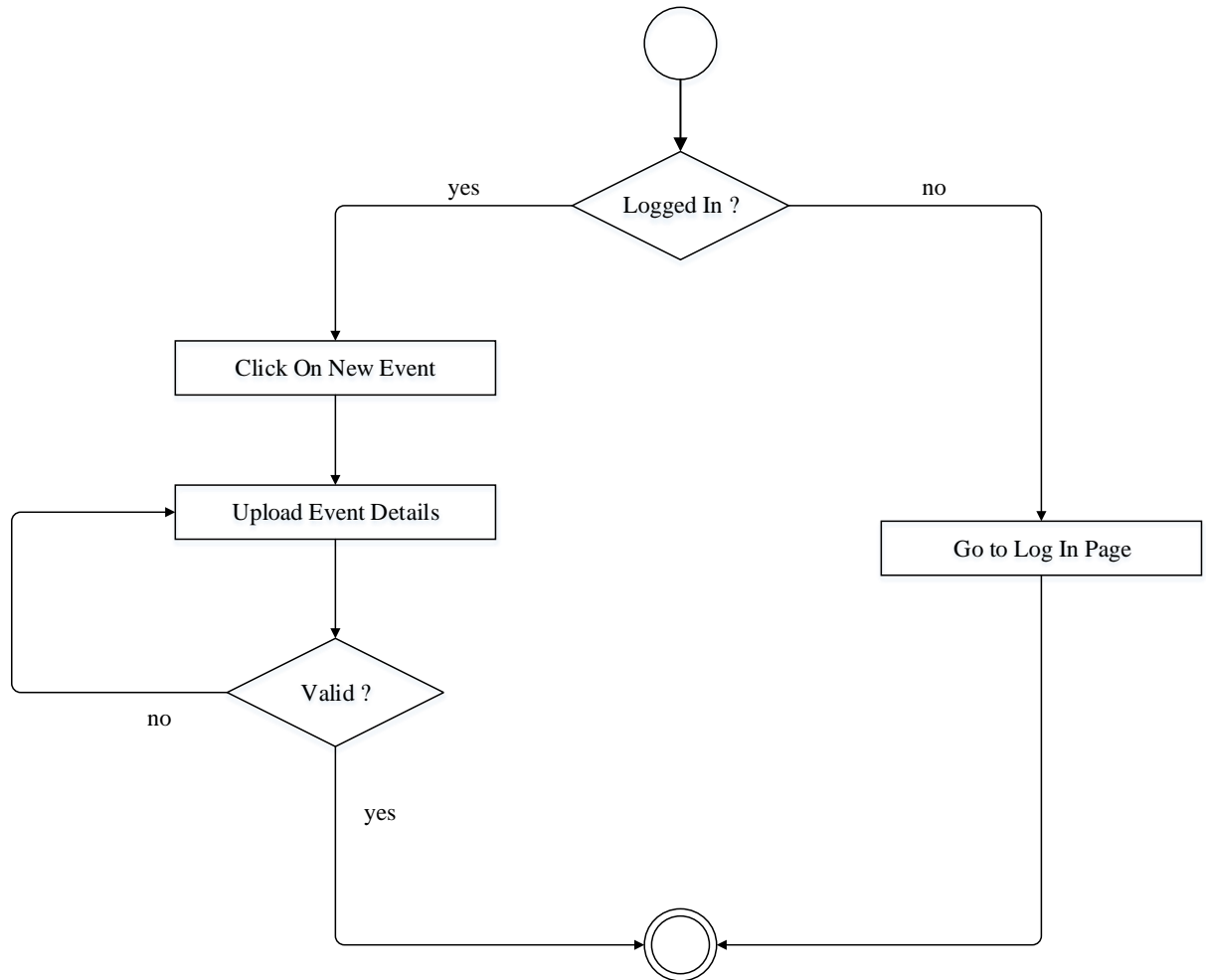


Figure 15: Activity Diagram – Level 1.2.1 (Create Event)

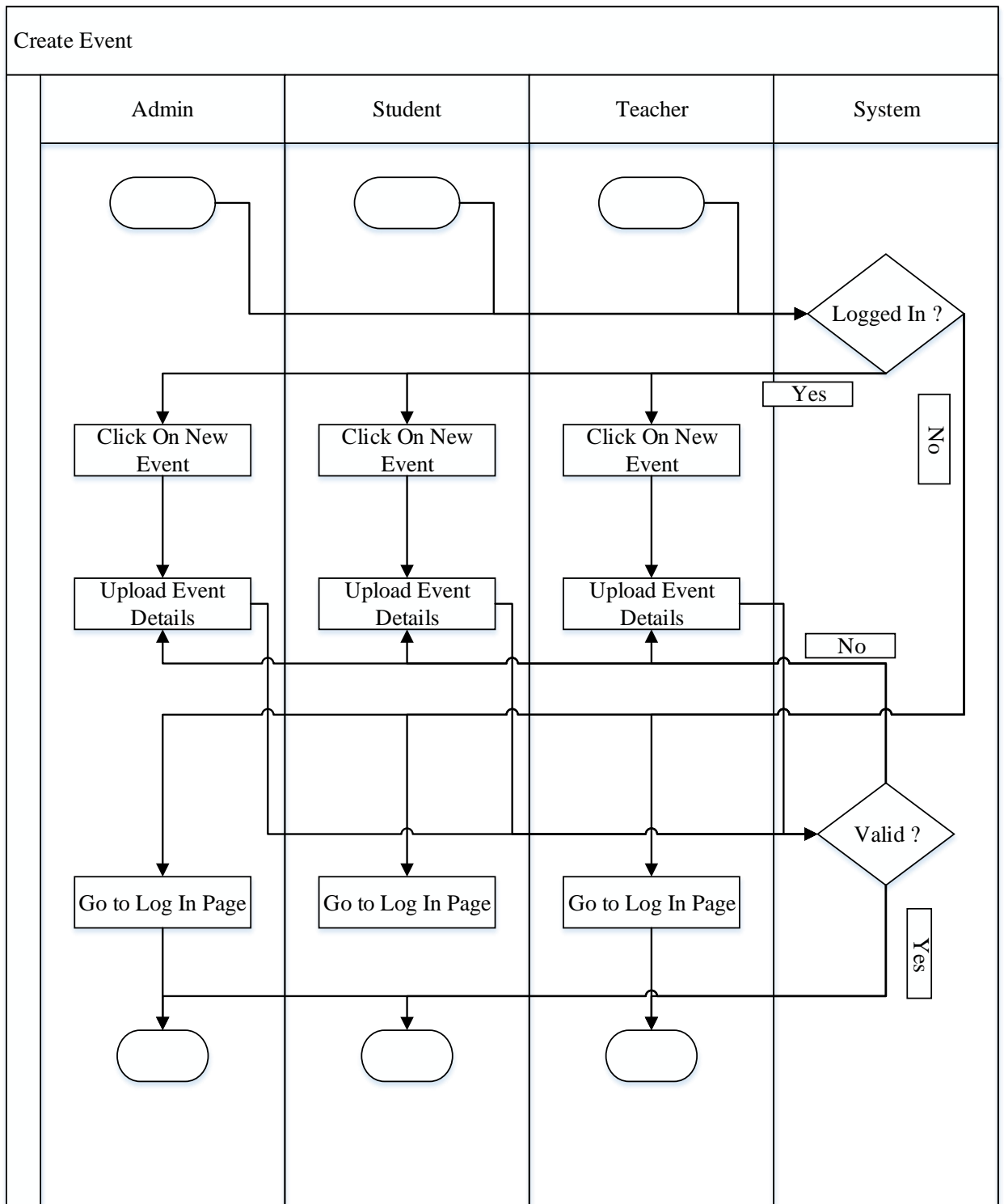


Figure 16: Swim Lane Diagram – Level 1.2.1 (Create Event)

#### 4.3.2.2 Update Event

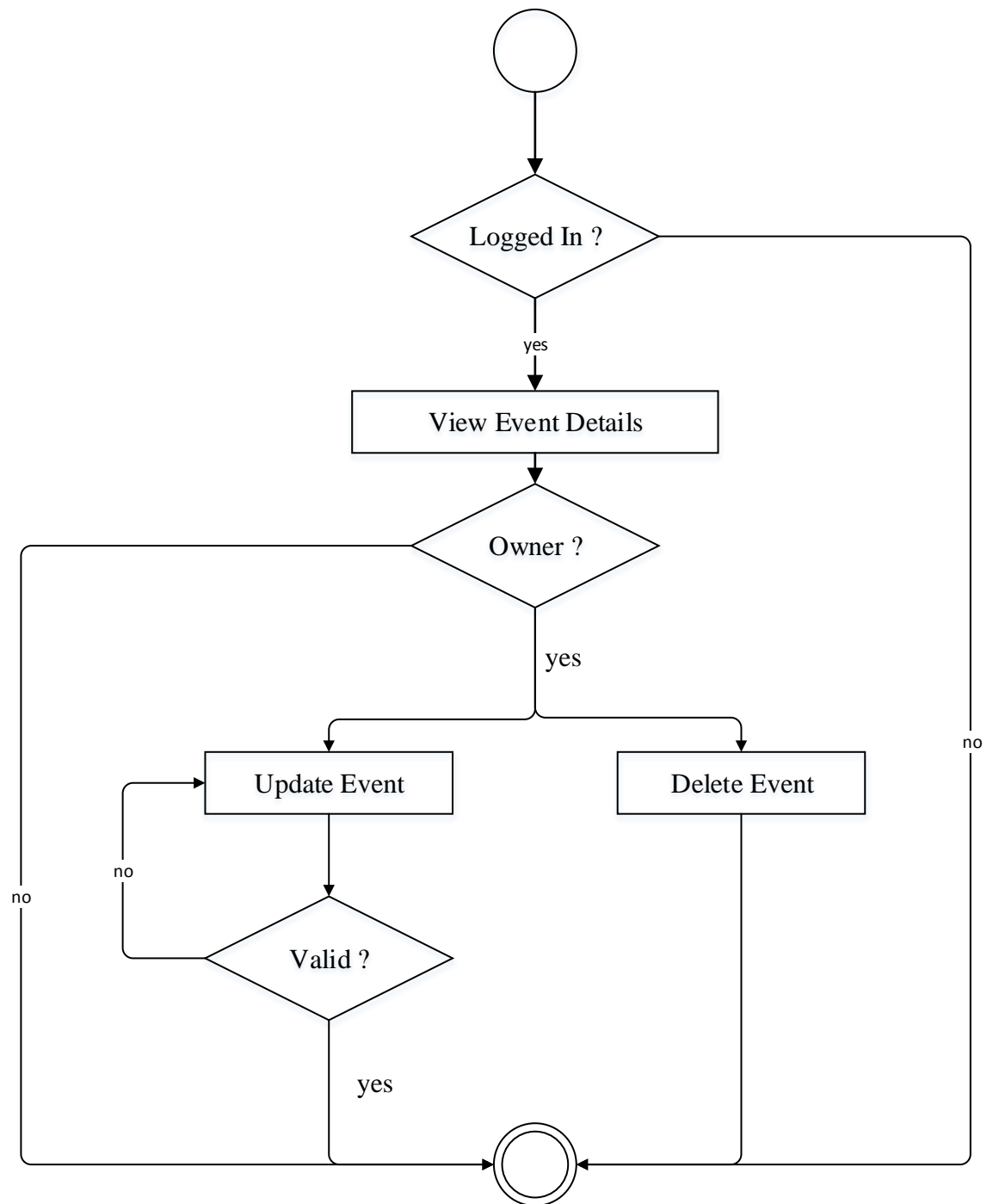


Figure 17: Activity Diagram – Level 1.2.2 (Update Event)



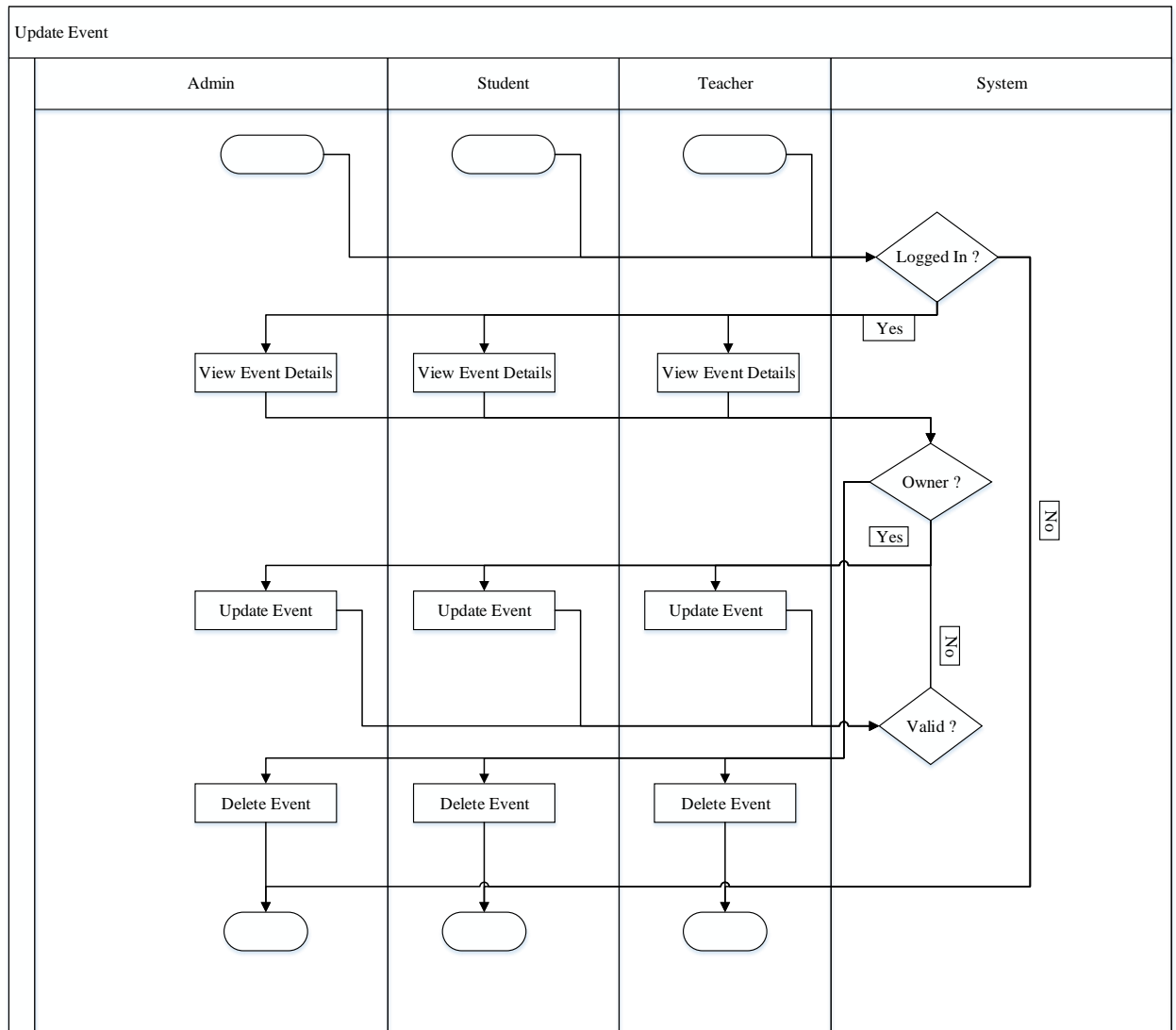


Figure 18: Swim Lane Diagram – Level 1.2.2 (Update Event)

### 4.3.3 Sub event

#### 4.3.3.1 add and delete Sub event

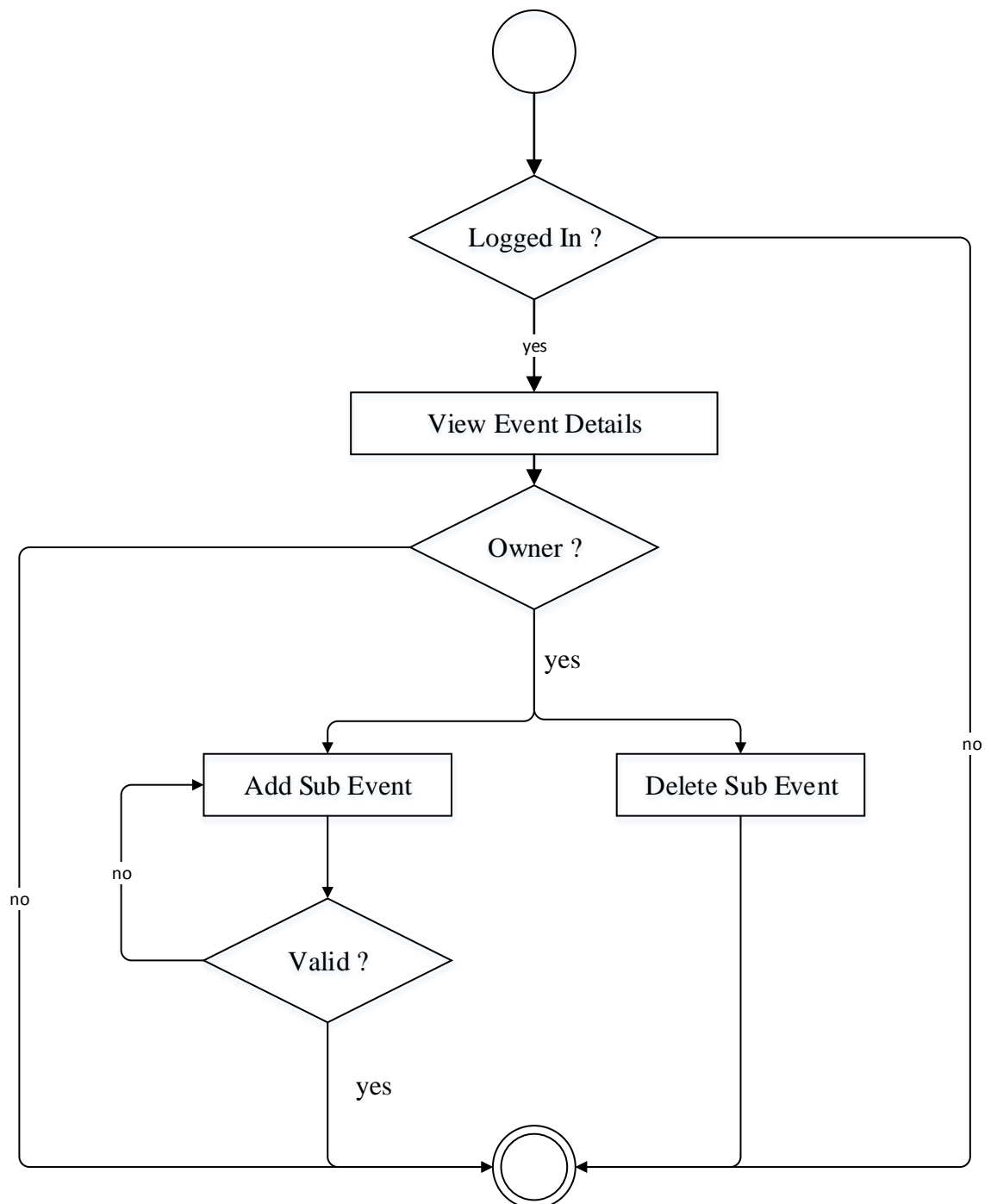


Figure 19: Activity Diagram – Level 1.2.3 (Sub Event)

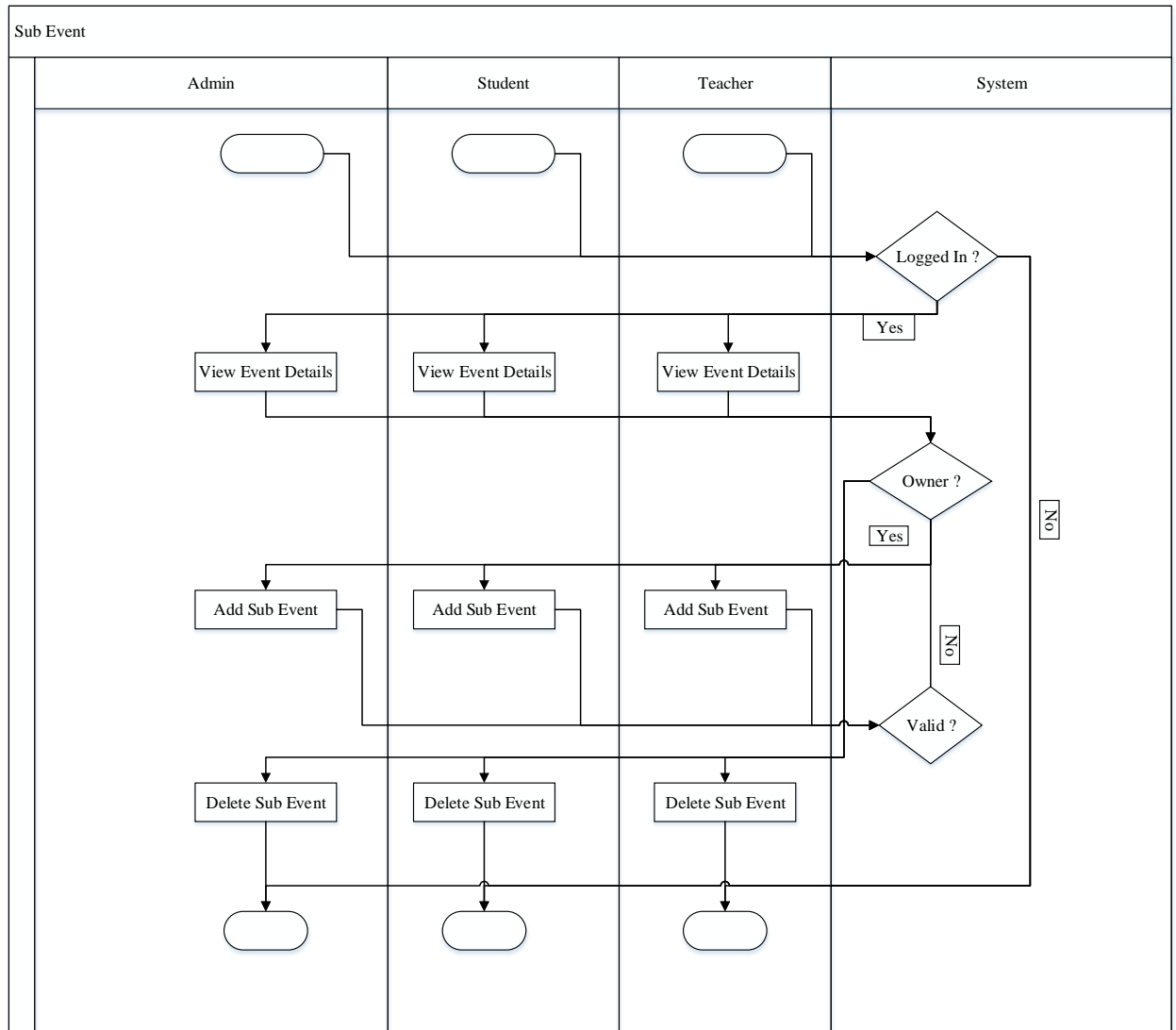


Figure 20: Swim Lane Diagram – Level 1.2.3

## 4.3.4 Template

### 4.3.4.1 Create Template

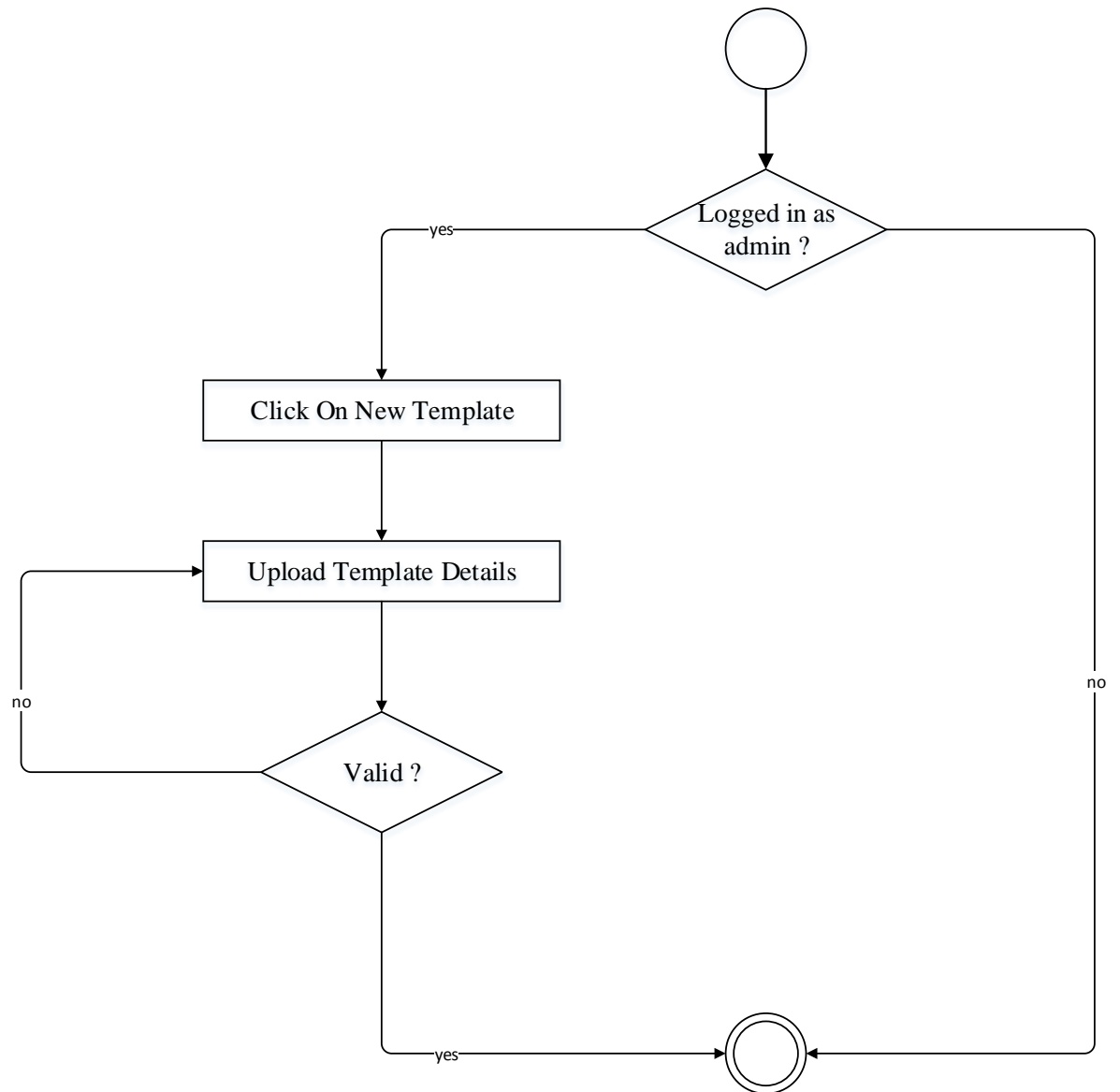


Figure 21: Activity Diagram – Level 1.3.1 (Create Template)

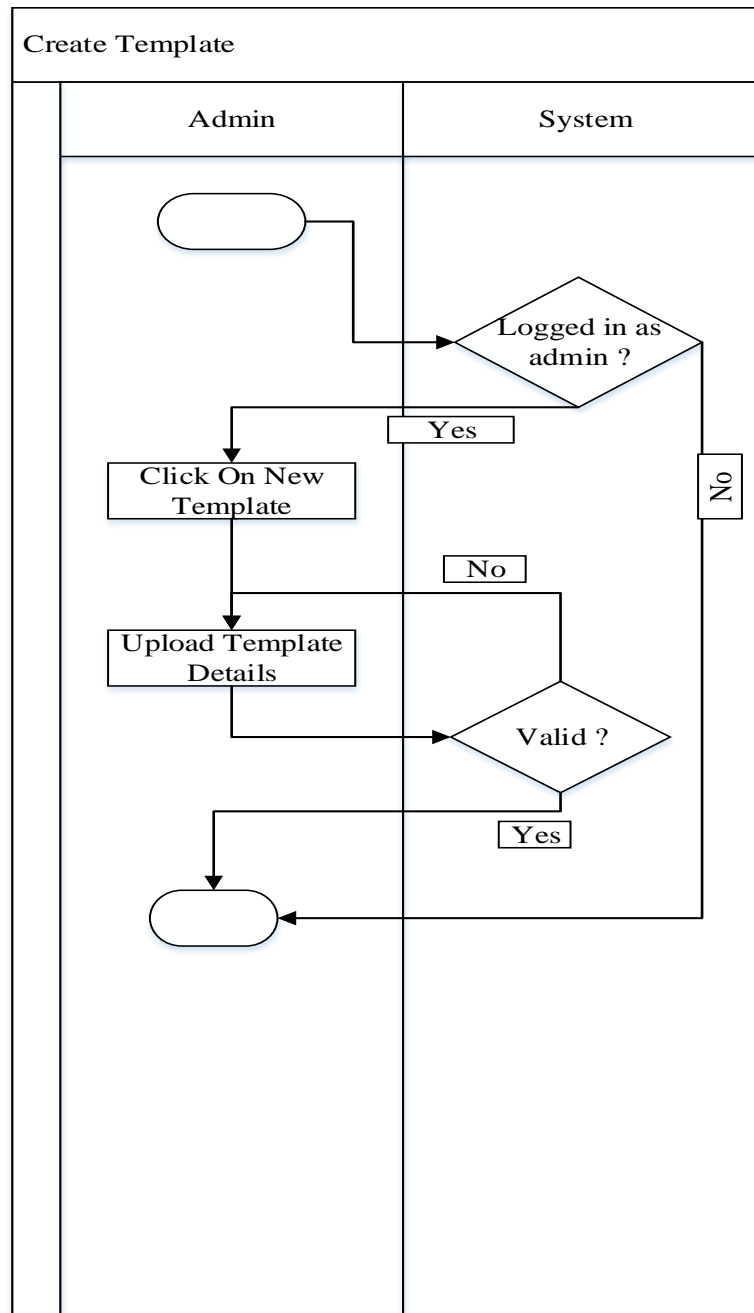


Figure 22: Swim Lane Diagram – Level 1.3.1 (Create Template)

#### 4.3.4.2 Update Template

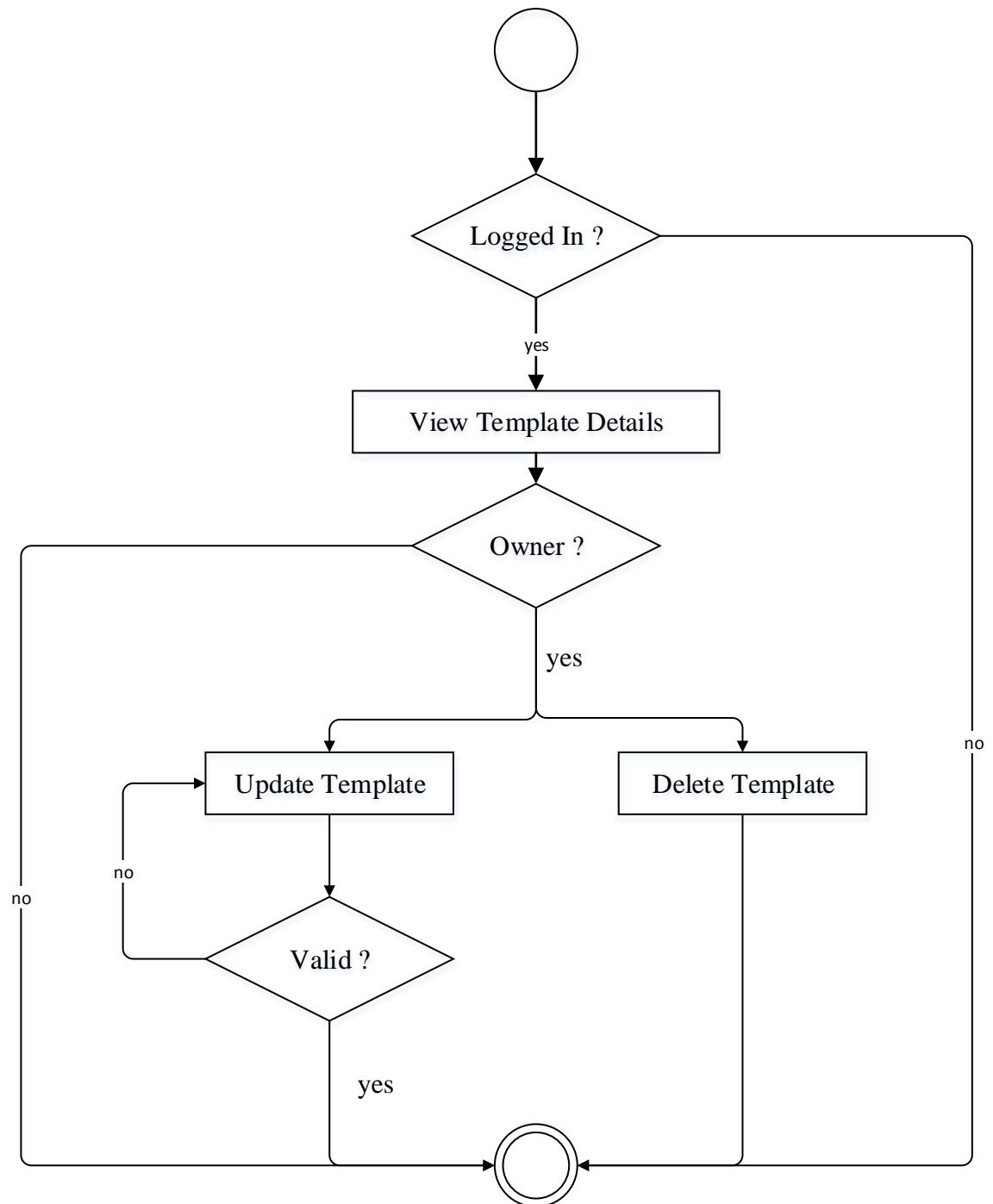


Figure 23: Activity Diagram – Level 1.3.2 (Update Template)

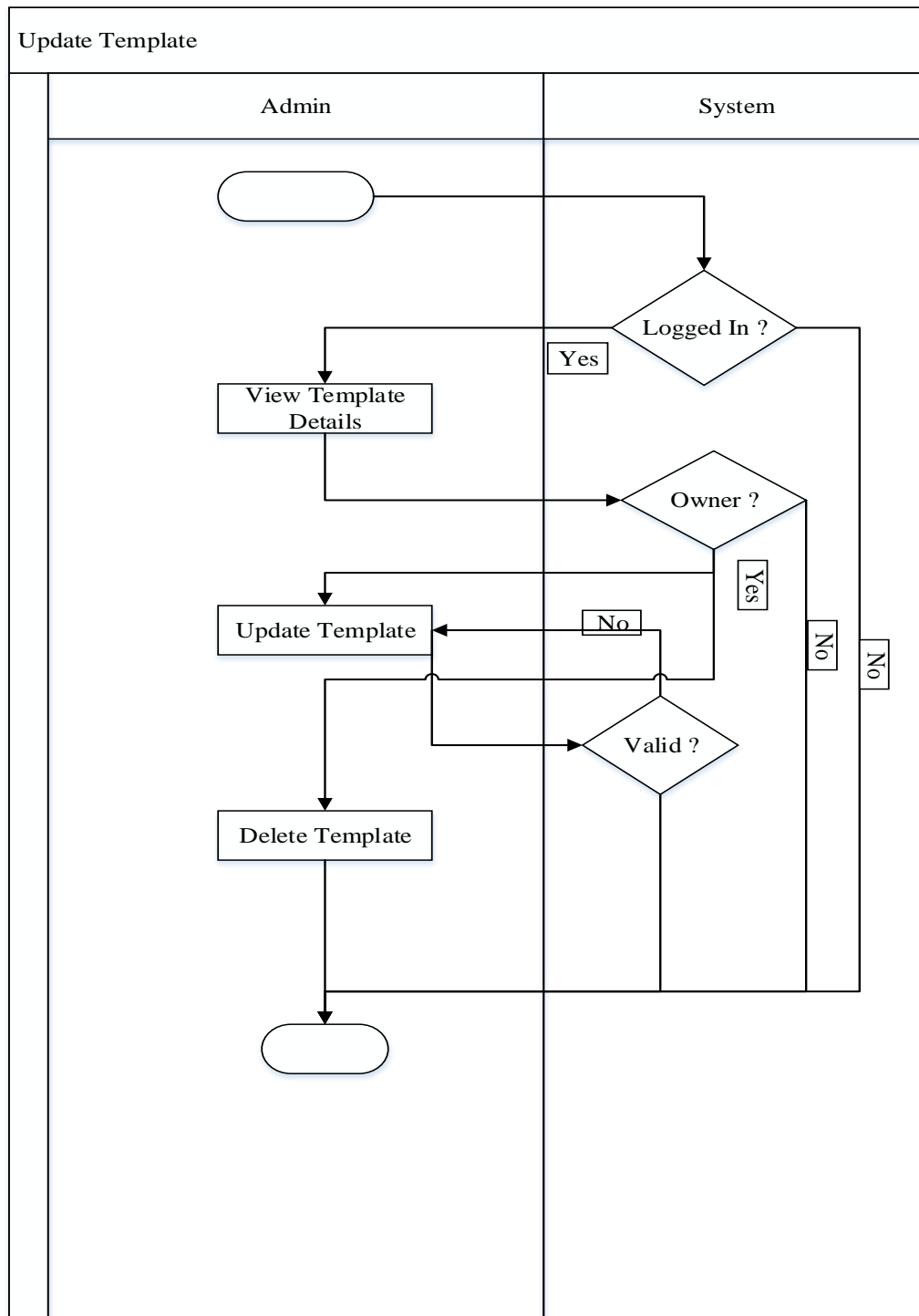


Figure 24: Swim Lane Diagram – Level 1.3.2 (Update Template)

#### 4.3.4.3 Use Template

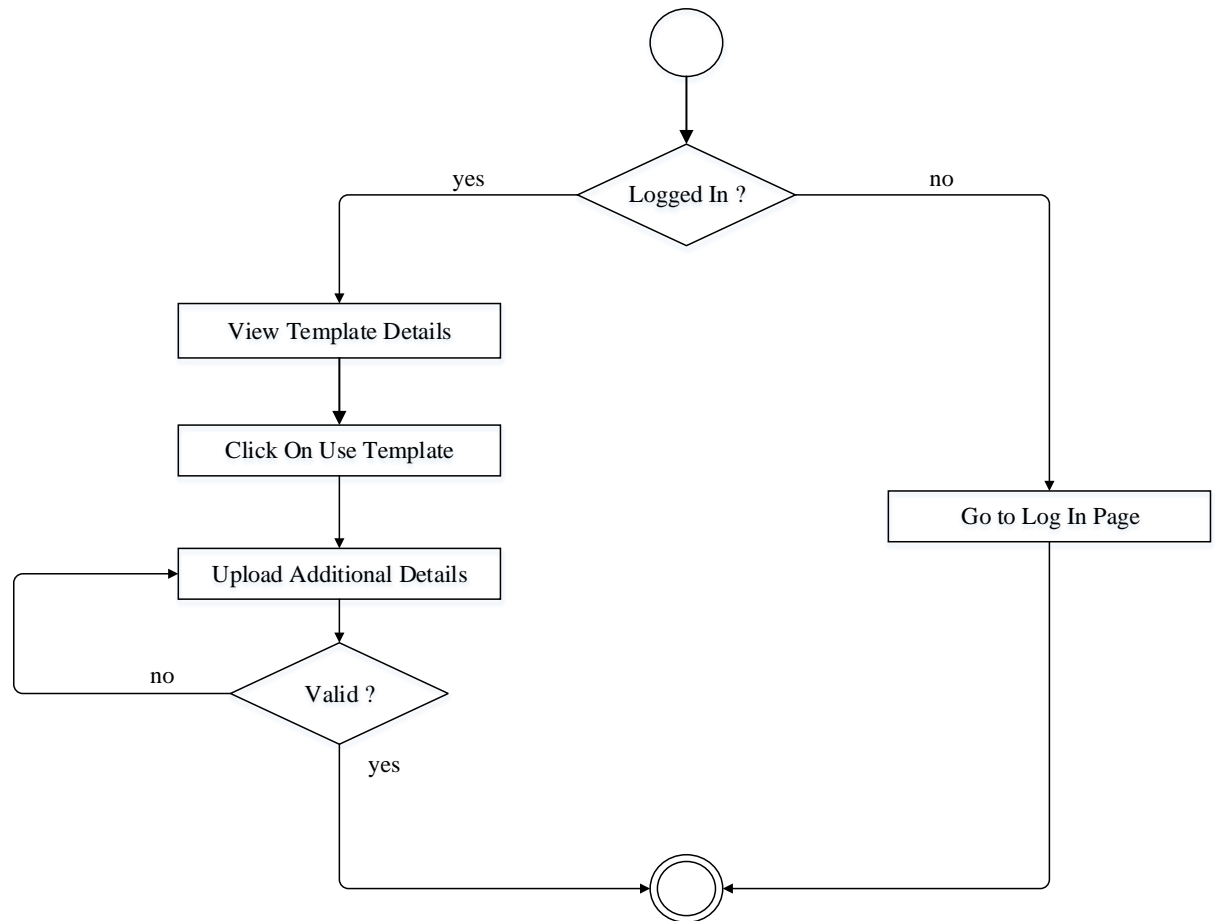


Figure 25: Activity Diagram – Level 1.3.3 (Use Template)



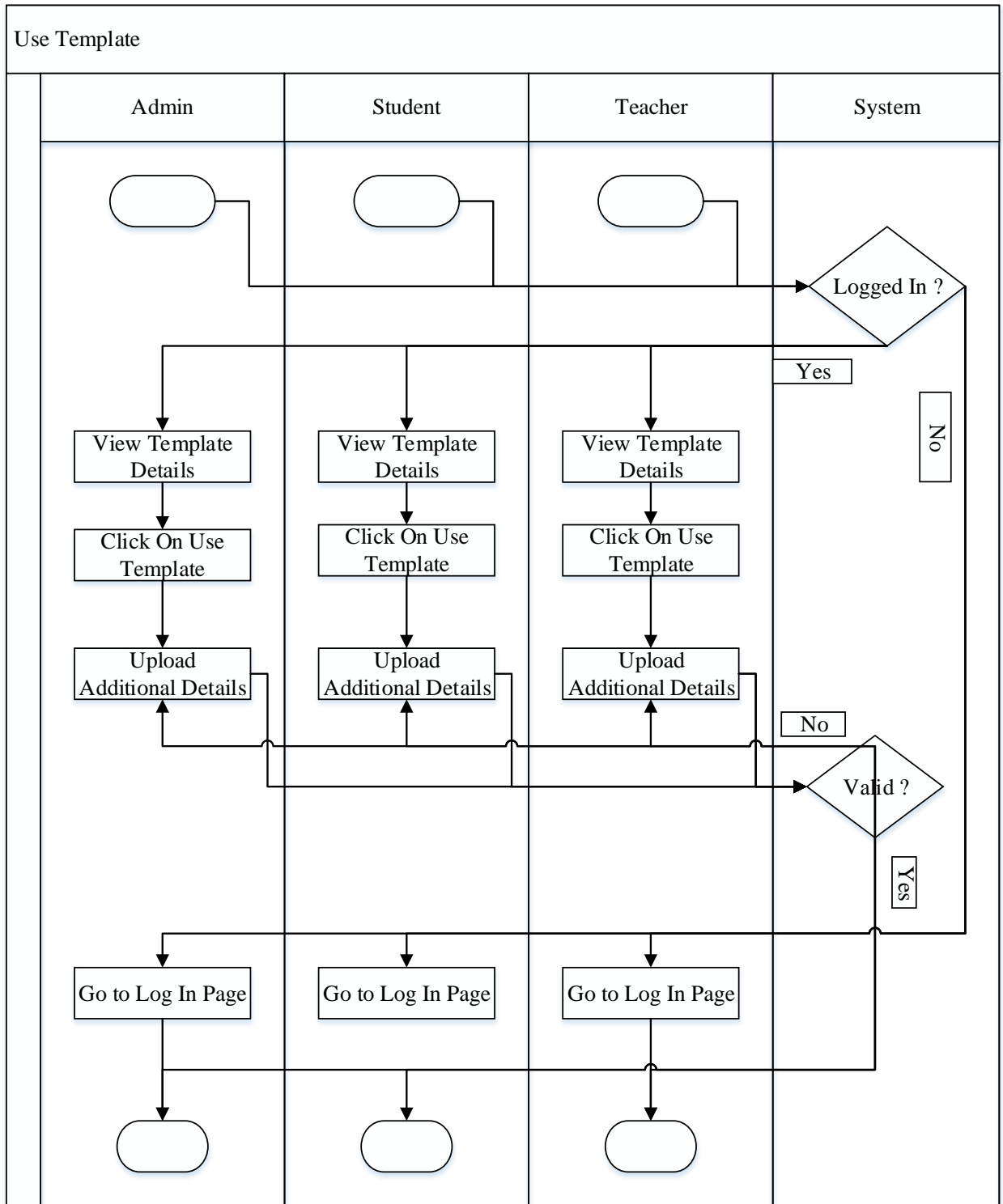


Figure 26: Swim Lane Diagram – Level 1.3.3 (Use Template)

### 4.3.5 sub template of template

#### 4.3.5.1 add and delete sub template of template

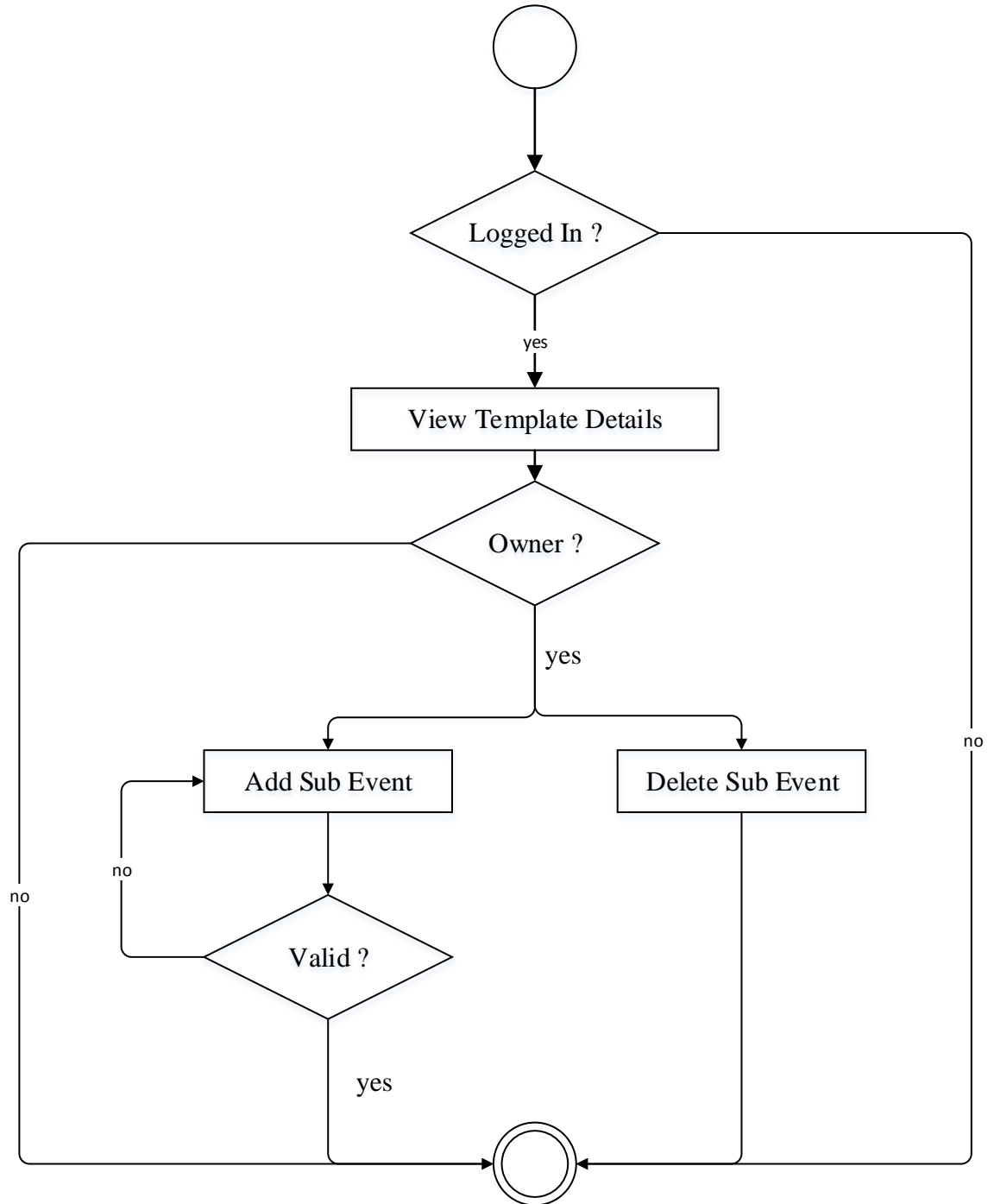


Figure 27: Activity Diagram – Level 1.3.4 (Sub Event)

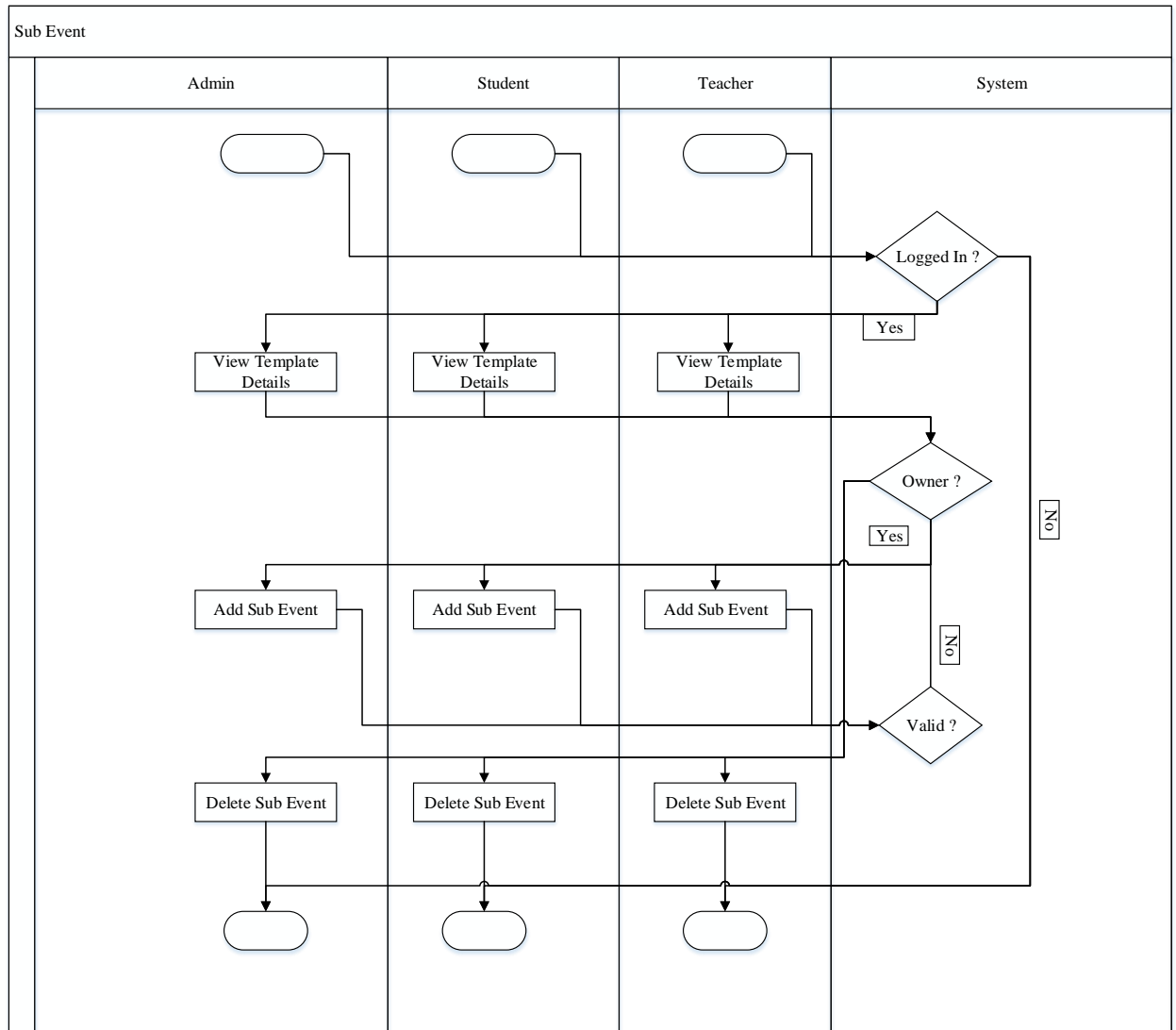


Figure 28: Swim Lane Diagram – Level 1.3.4 (Sub Event)

## Chapter 5: Data Model

## 5.1 Introduction

If software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

## 5.2 Data Object Selection

A data object is a representation of information which has different properties or attributes that must be understood by software. Here is the table of potential data objects. Some attributes were not present in the scenario, but I included them as they are necessary attributes in this system.

Table 5.1: Data Object Selection

Noun	Attributes	Description	Remarks
Student	Name, roll, email, password, list(event url)	Potential Data Object	Accepted
Name		An Attribute of Student, Teacher, Admin	Rejected
Email		An Attribute of Student, Teacher, Admin	Rejected
Password		An Attribute of Student, Teacher, Admin	Rejected
Roll		An Attribute of Student	Rejected
Admin	Name, designation, email, password, list(event url)	Potential Data Object	Accepted
Teacher	Name, designation, email, password, list(event url)	Potential Data Object	Accepted
Designation		An Attribute of Teacher and Admin	Rejected
Event	Event url, name, description, list (sub-event), start date, duration, notification	Potential Data Object	Accepted

Description		An Attribute of Teacher, Admin and Event	Rejected
Start date		An Attribute of Event	Rejected
Duration		An Attribute of Event	Rejected
Sub-Event	Name, start, duration, notification	Potential Data Object	Accepted
Template	Template url, name, list(sub event of template), duration	Potential Data Object	Accepted
Sub Event of Template	Name, start, duration	Potential Data Object	Accepted

## 5.3 Data Objects and Attributes

This is a brief view of all attributes I have found so far.

Student = Name + Email + Roll+ Password +list (event url)

Admin = Name + Email + designation + Password + list (event url)

Teacher = Name + Email + designation + Password + list (event url)

Event = event url + name + description + list(sub-event) + duration + start date + notification

Sub-Event = name + description + duration + start date + notification

Template = template url + name +list(Sub event of template) + duration

Sub event of template = name + start + duration

Student	
Attributes	Types
Name	Varchar (30)
Roll	Int (10)
Email	Varchar (30)
Password	Varchar (30)
List<Event url>	

Admin	
Attributes	Types
Name	Varchar (30)
Designation	Varchar (30)
Email	Varchar (30)
Password	Varchar (30)
List<Event url>	

Teacher	
Attributes	Types
Name	Varchar (30)
Designation	Varchar (30)
Email	Varchar (30)
Password	Varchar (30)
List<Event url>	

Event	
Attributes	Types
Event url	Varchar (30)
Name	Varchar (30)
description	Varchar (500)
duration	Int (10)
start date	Date
notification	Boolean
list< Sub event>	

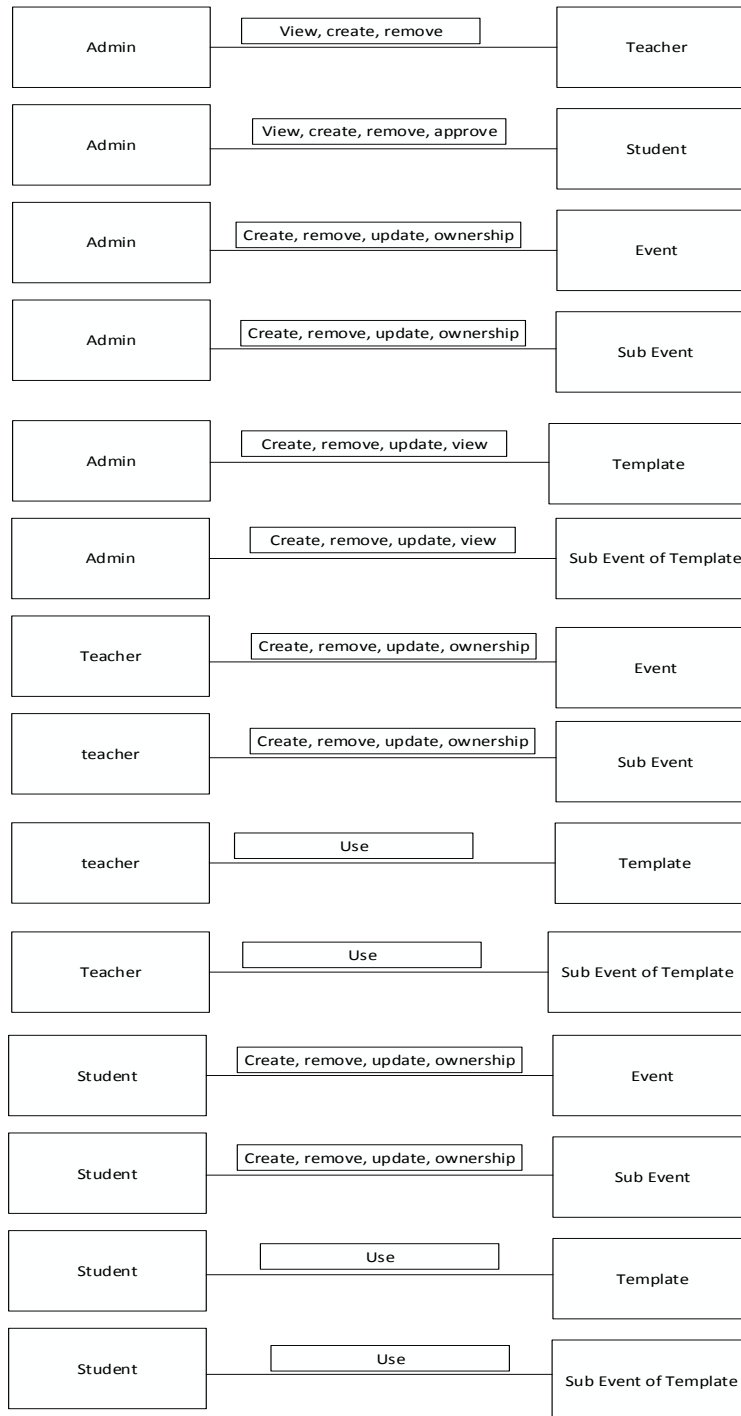
Sub-Event	
Attributes	Types
Name	Varchar (30)
description	Varchar (30)
duration	Int (10)
start	Date
notification	Boolean

Template	
Attributes	Types
Template url	Varchar (30)
Name	Varchar (30)
duration	Int (10)
list<Sub event of template>	

Sub event of template	
Attributes	Types
Name	Varchar (30)
Start	Int (10)
Duration	Int (10)

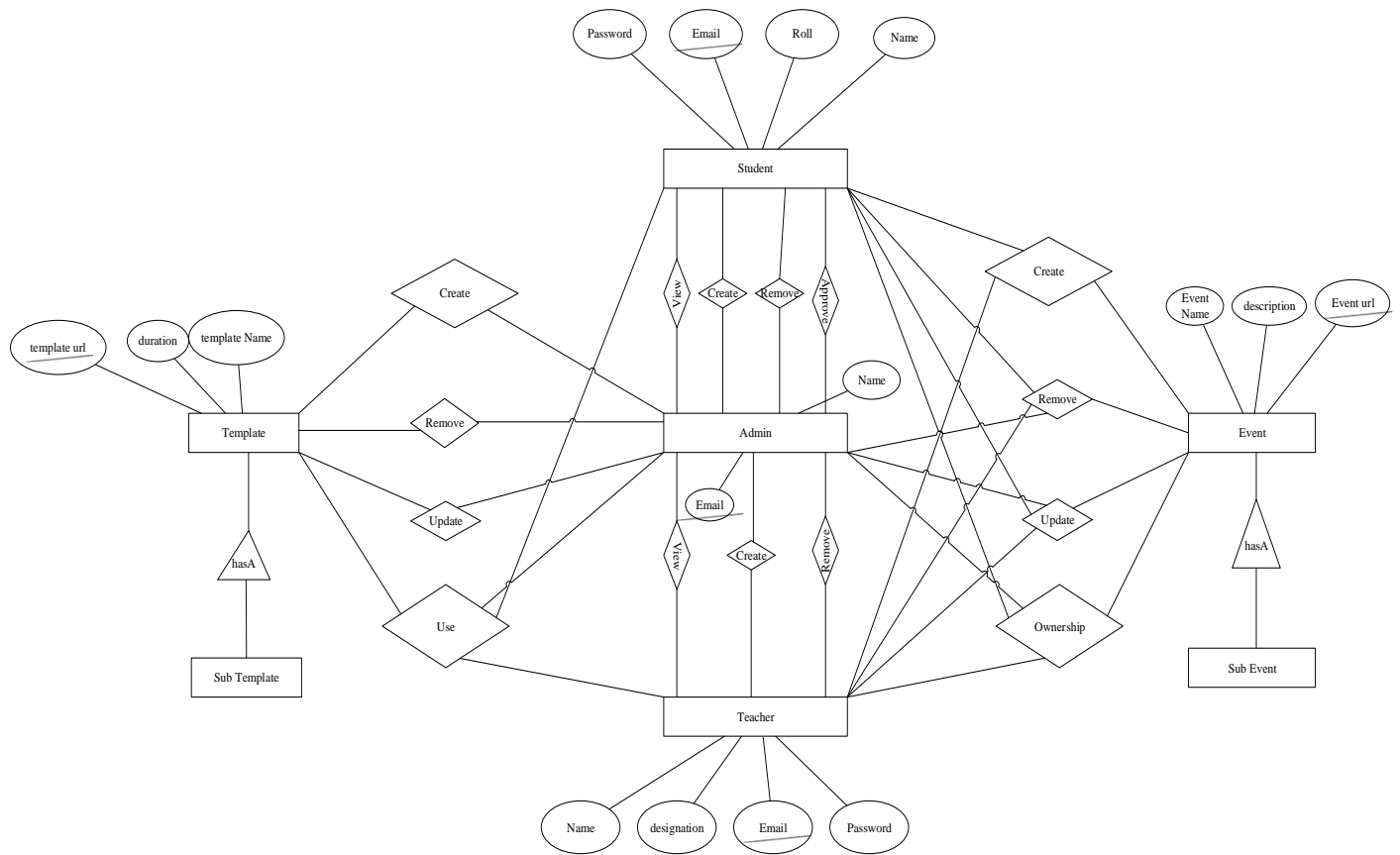
## 5.4 Relationship Between Data Objects

Here we have shown pair wise relation between two entities.





## 5.5 Entity Relationship(ER) Diagram



## Chapter 6: Class Based Modeling

## 6.1 Introduction

Class based modeling is designed to demonstrate the whole software on the view or perspective of object oriented concept. In this model what the objects are and what their responsibilities will be, how they will interact with each other is defined very clearly.

## 6.2 General Classifications

After selecting the nouns by grammatical parsing from the solution space of the story, these are characterized in seven general classifications. The seven general characteristics are as follow:

1. External entities
2. Things
3. Occurrences
4. Roles
5. Organizational units
6. Places
7. Structures

Here the ‘passed’ nouns are the potential classes and the ‘failed’ nouns become the attributes of the classes.

Table 6.1: General Classification

Noun	General Classifications	Remarks
Student	4, 5, 7	Yes
Name		No
Email		No
Password		No
Roll		No
Admin	4, 5, 7	Yes
Teacher	4, 5, 7	Yes
Designation		No
Event	7	Yes
Event name		No
Description		No
Start date		No
Duration		No
Event owner		No
Sub event	7	Yes
Notification		No
Template	7	Yes
Sub Event of Template	7	Yes

## 6.3 Selection Criteria

The potential classes are then selected as class by six 'selection characteristics'. A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

Table 6.2: Selection Criteria

Noun	Selection Criteria	Remarks
Student	1, 2, 3, 4, 5, 6	Yes
Admin	1, 2, 3, 6	Yes
Teacher	1, 2, 3, 4, 5, 6	Yes
Event	1, 2,3, 4,5,6	Yes
Sub event	1,2,3,4,5,6	Yes
Template	1,2,3,4,5,6	Yes
Sub Event of Template	1,2,3,4,5,6	Yes

Primary classes:

1. Student
2. Admin
3. Teacher
4. Event
5. Sub Event
6. Template
7. Sub Event of Template

I will also need Database class to manage all operations with database.

## 6.4 Attribute Selection

Table 6.3: Attribute Selection

Class Name	Attributes
Student	Name, roll, email, password, list <Event url>
Admin	Name, designation, email, password, list <Event url>
Teacher	Name, designation, email, password, list <Event url>

Event	Event name, description, start date, duration, list<sub Event>
Sub Event	Sub Event name, description, start, duration, notification
Template	Template name, duration, list<sub Event of Template>
Sub Event of Template	Sub Event of Template name, duration, start
Database	All attributes

## 6.5 Method Identification

Table 6.4: Method Identification

Class Name	Methods
Student	Get() Set() Sign Up() Sign_in() Sign_out() Change_password() Create event() Remove event() Update event()
Admin	Get() Set() Sign_in() Sign_out() Change_password() Create_user() Remove_user() Add user() Remove user() Verify_user_requesst() Delete_user_request() Create event() Remove event() Update event() Create template() Remove template() Update template()
Teacher	Get()

	Set() Sign_in() Sign_out() Change_password() Create event() Remove event() Update event()
Event	Insert() Delete() Update() View()
Sub Event	Insert() Delete() Update() View()
Template	Insert() Delete() Update() View()
Sub Event of Template	Insert() Delete() Update() View()
Database	Insert() Delete() Update() View()

## 6.6 Class Card

Student	
Attributes	Methods
Name Roll Email	Get() Set() Sign Up()

Password list <Event url>	Sign_in() Sign_out() Change_password() Create event() Remove event() Update event()
Responsibilities	Collaborative Class
Create / Remove / Update Event	Database, Event, Sub event, Template, Sub event of template

Admin	
Attributes	Methods
Name Designation Email Password list <Event url>	Get() Set() Sign_in() Sign_out() Change_password() Create_user() Remove_user() Remove user() Verify_user_requesst() Delete_user_request() Create event() Remove event() Update event() Create template() Remove template() Update template()
Responsibilities	Collaborative Class
Create / Remove / Update Event	Database, Event, Sub event, Template, Sub event of template
Create / Remove / Update Event	Database, Template, Sub event of template
Create User	Database, Admin, Student, Teacher
Remove User	Database, , Admin, Student, Teacher

Teacher	
Attributes	Methods
Name Designation Email Password list <Event url>	Get() Set() Sign_in() Sign_out() Change_password() Create event() Remove event() Update event()
Responsibilities	Collaborative Class
Create / Remove / Update Event	Database, Event, Sub event, Template, Sub event of template

Database	
Attributes	Methods
All attributes	Insert() Delete() Update() View()
Responsibilities	Collaborative Class
Insertion	
Deletion	
Updating	
Viewing	Student, Teacher, Admin, Event, Template

Event	
Attributes	Methods
Event name Description Start date Duration list<sub Event>	Insert() Delete() Update() View()
Responsibilities	Collaborative Class
Viewing	Student, Teacher, Admin, Event, Template



Sub Event	
Attributes	Methods
Sub Event name Description Start Duration Notification	Insert() Delete() Update() View()
Responsibilities	Collaborative Class
Viewing	Student, Teacher, Admin, Event, Template

Template	
Attributes	Methods
Template name Duration list<sub Event of Template>	Insert() Delete() Update() View()
Responsibilities	Collaborative Class
Viewing	Student, Teacher, Admin, Event, Template

Sub event of Template	
Attributes	Methods
Sub Event of Template name Duration Start	Insert() Delete() Update() View()
Responsibilities	Collaborative Class
Viewing	Student, Teacher, Admin, Event, Template

## 6.7 CRC Diagram

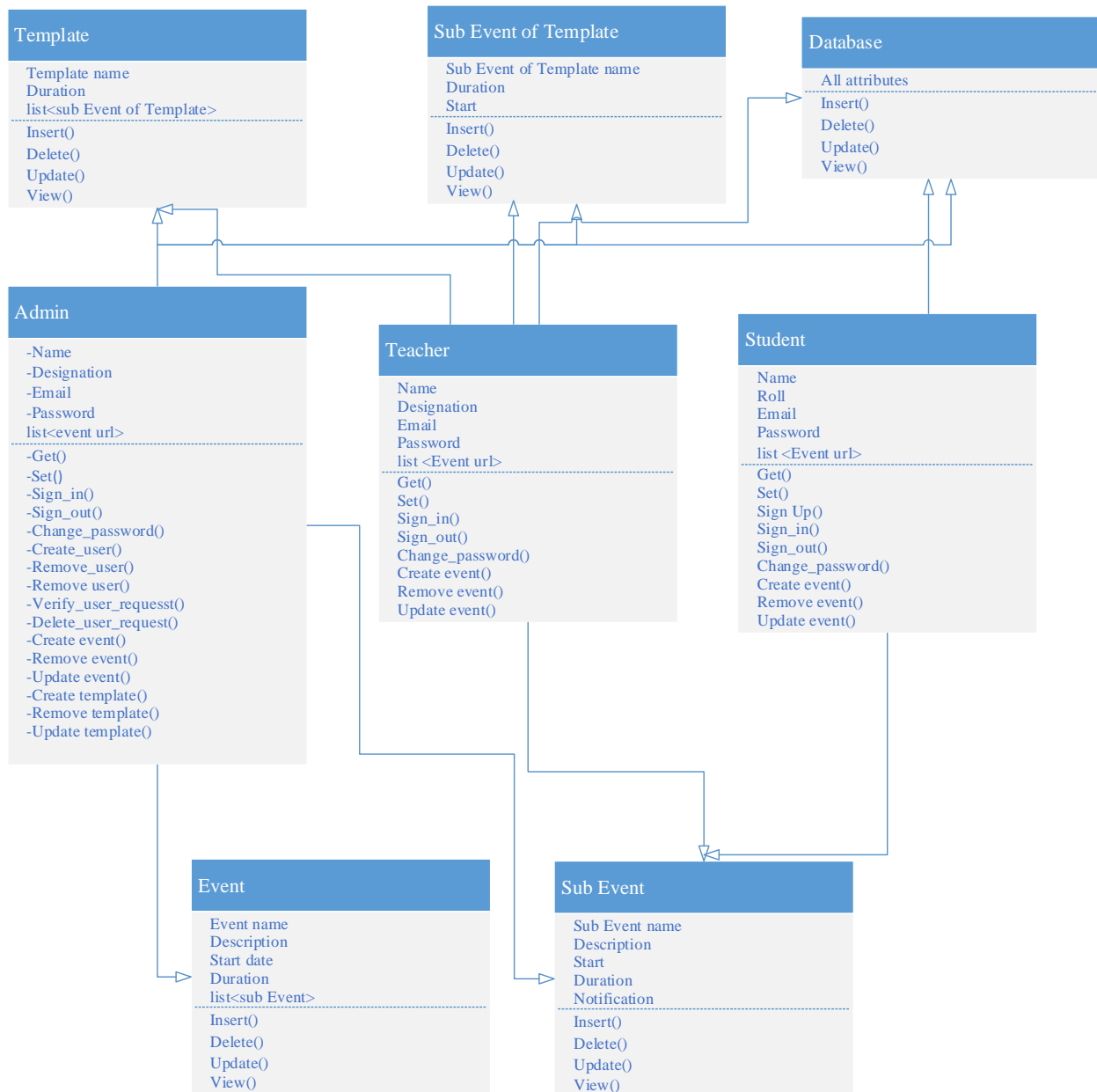


Fig 6.1: CRC diagram

## Chapter 7: Flow Oriented Model

## 7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today.

## 7.2 Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) takes an input-process-output view of a system. Data objects flow into the software, are transformed by processing elements and resultant data objects flow out of the software. Data objects are represented by labeled arrows and transformations are represented by circles.

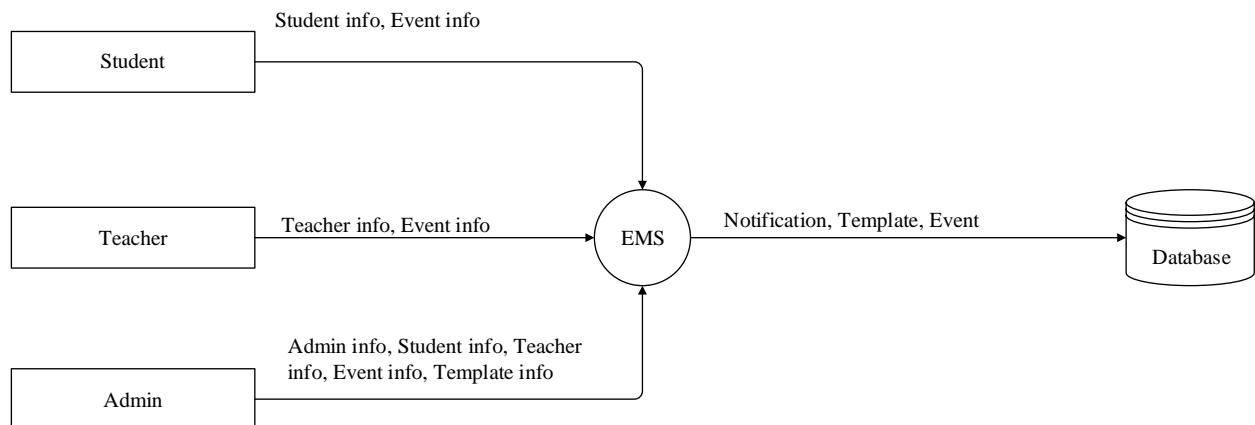


Figure 29: DFD ( Level-0)

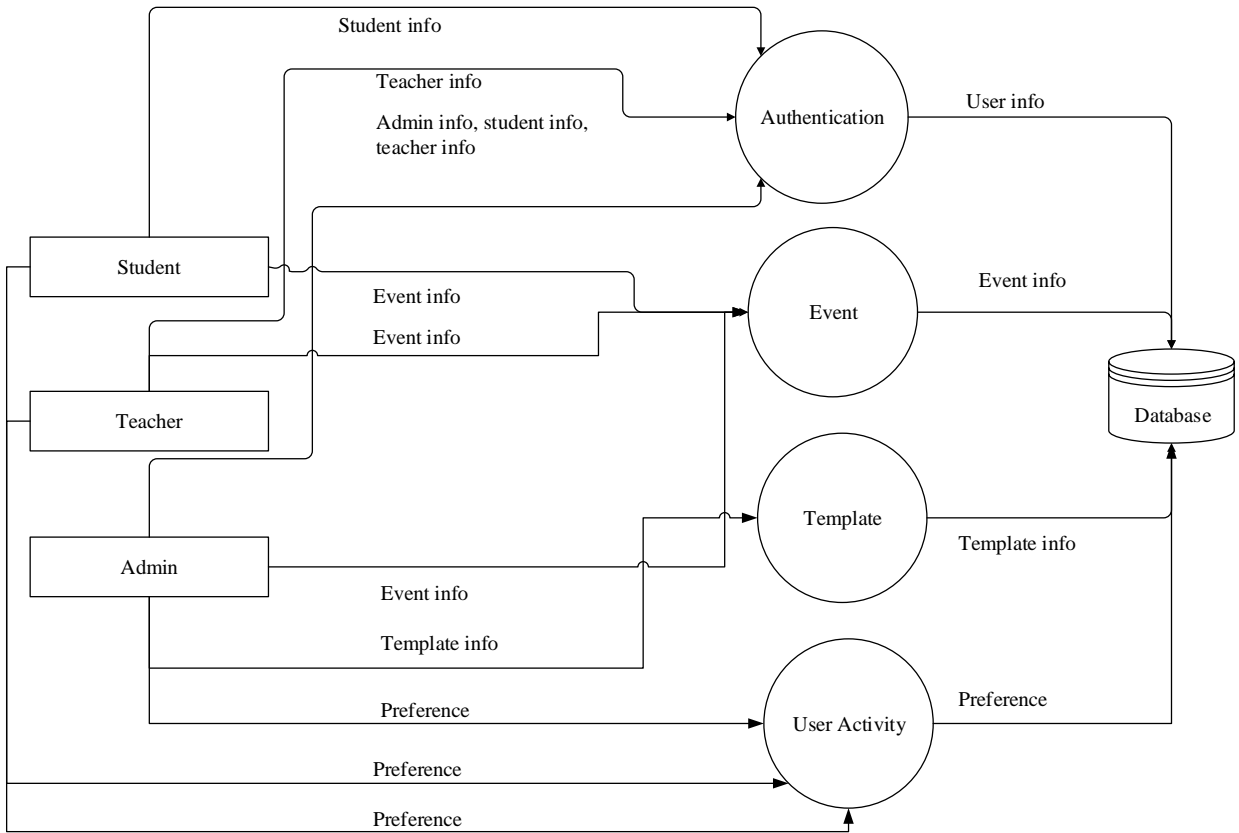


Figure 30: DFD (Level- 1)

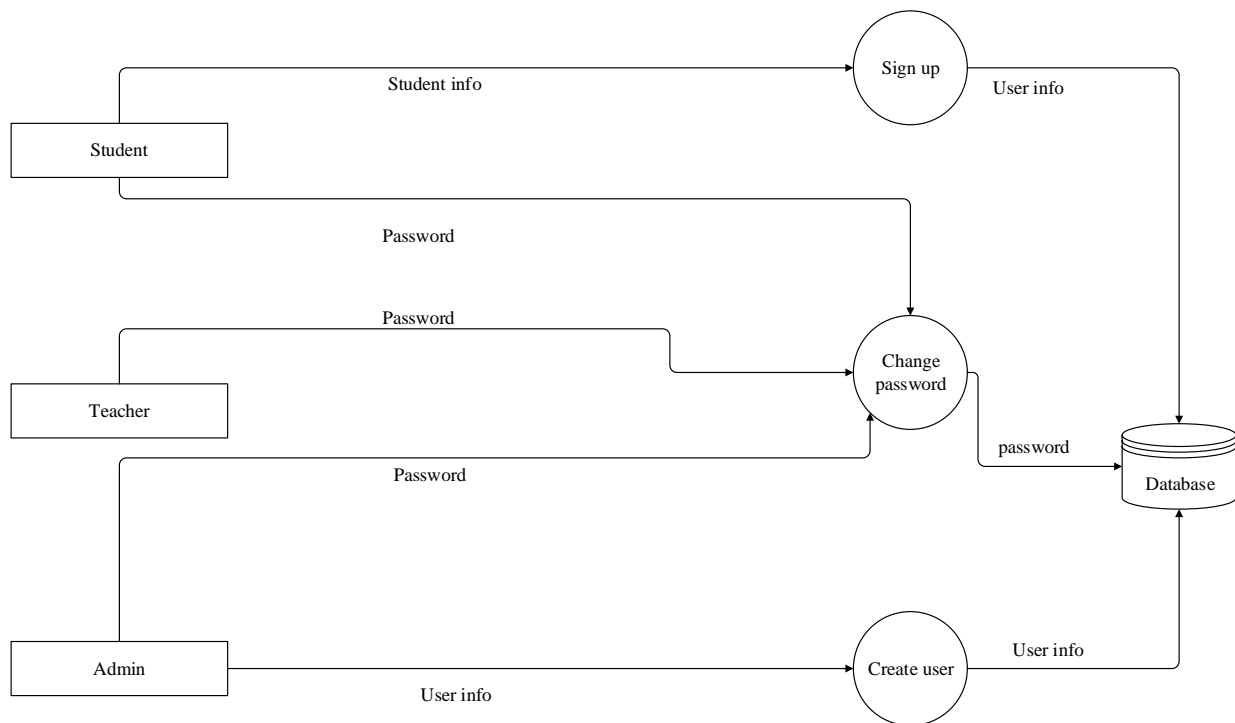


Figure 31: DFD ( Level-1.1-Authentication)

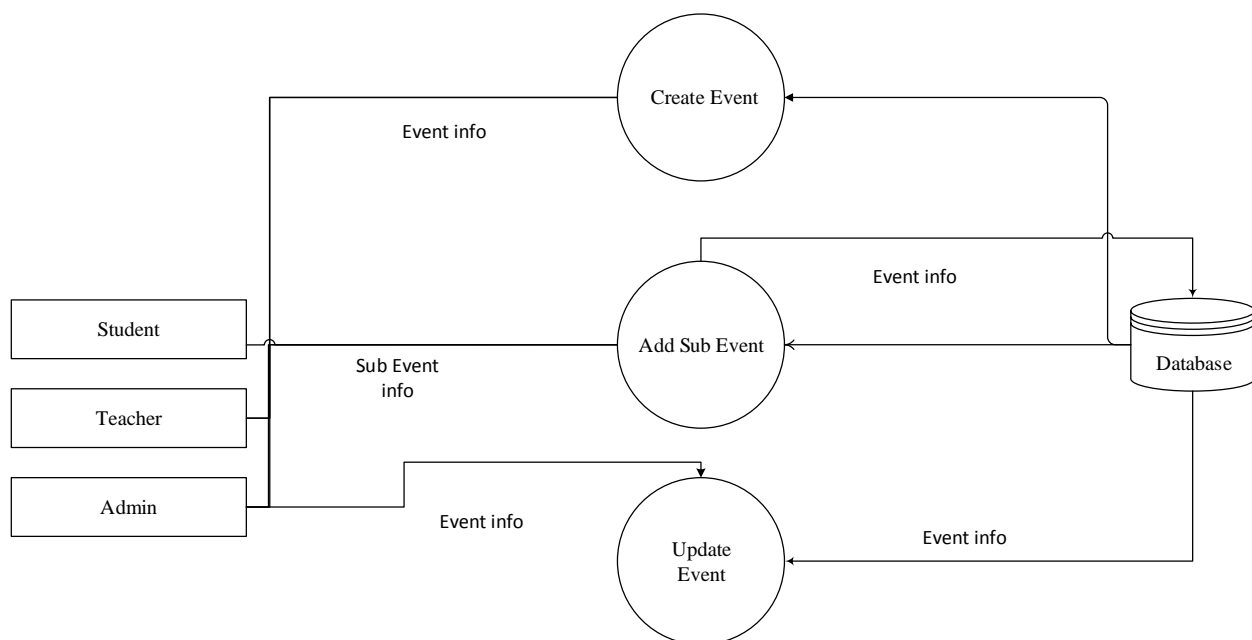


Figure 32: DFD ( Level-1.2-Event)

## Chapter 8: Behavioral Model

## 8.1 Introduction

Behavior modeling is also referred to as State modeling, State machines and State transition matrix. Behavior modeling is when one thinks of his ideas in terms of states and transitions. This requires both identifying all of the interesting states of being that software or its components are likely to be in. And also, at a high level abstracting what events are likely to cause software or its components to change between states of being.

## 8.2 Identifying Events

Here we have identified events from the Usage Scenario and listed their corresponding initiators & collaborators.

Table 8.1: Identifying Events

Event	Initiator	Collaborator
Sign up	Student	Admin, Database
Verification	Admin	Student, Database
Create User	Admin	Admin, Student, Teacher, Database
Remove User	Admin	Admin, Student, Teacher, Database
Change password	Student, Teacher, Admin	Database
Sign in	Student, Teacher, Admin	Database
Sign out	Student, Teacher, Admin	Database
Create Event	Student, Teacher, Admin	Database
Remove Event	Student, Teacher, Admin	Database
Delete Event	Student, Teacher, Admin	Database
View Event	Student, Teacher, Admin	Database
Create sub event	Student, Teacher, Admin	Database
Remove sub event	Student, Teacher, Admin	Database
Delete sub event	Student, Teacher, Admin	Database
View sub event	Student, Teacher, Admin	Database
Change ownership	Student, Teacher, Admin	Database
Create template	Admin	Database
Remove template	Admin	Database
Update template	Admin	Database
View Template	Student, Teacher, Admin	Database
Notification status	Student, Teacher, Admin	Database



## 8.3 State Transition Diagram

State Transition Diagram represents active states for each class and the events (triggers) that cause changes between these active states. Here I have provided diagram for each of the actors.

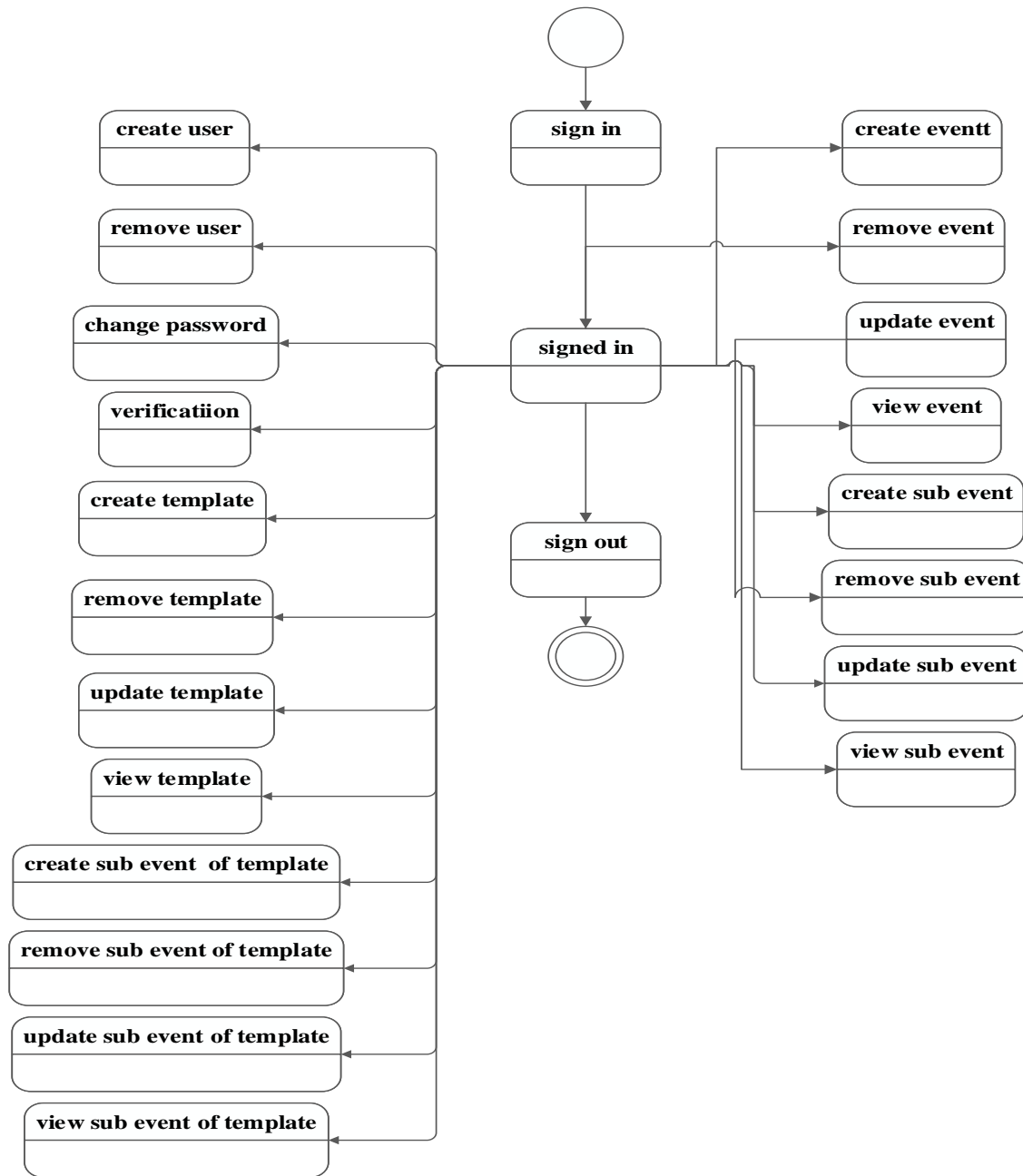


Figure 33: State transition diagram of Admin

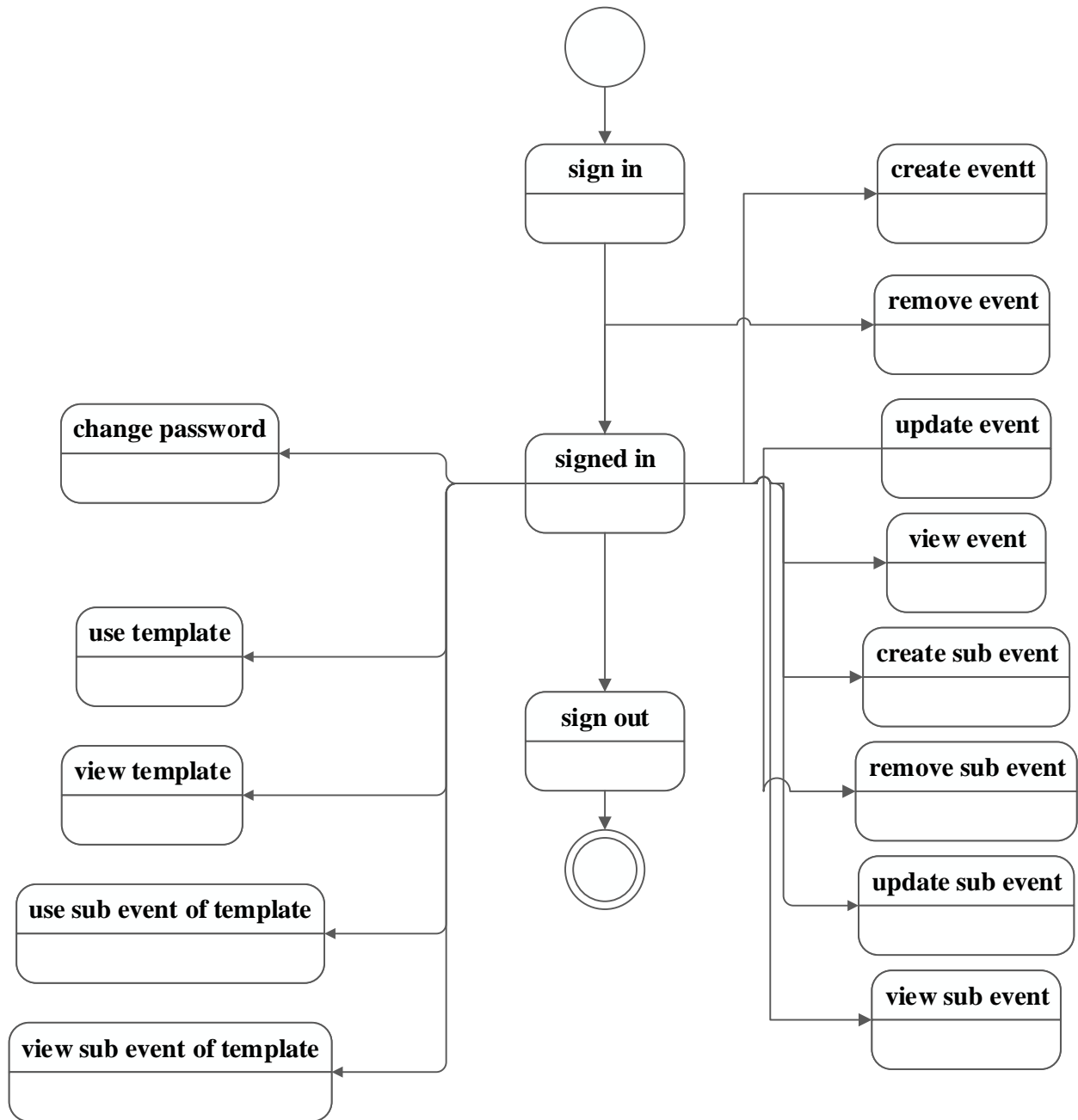


Figure 34: : State transition diagram of Teacher

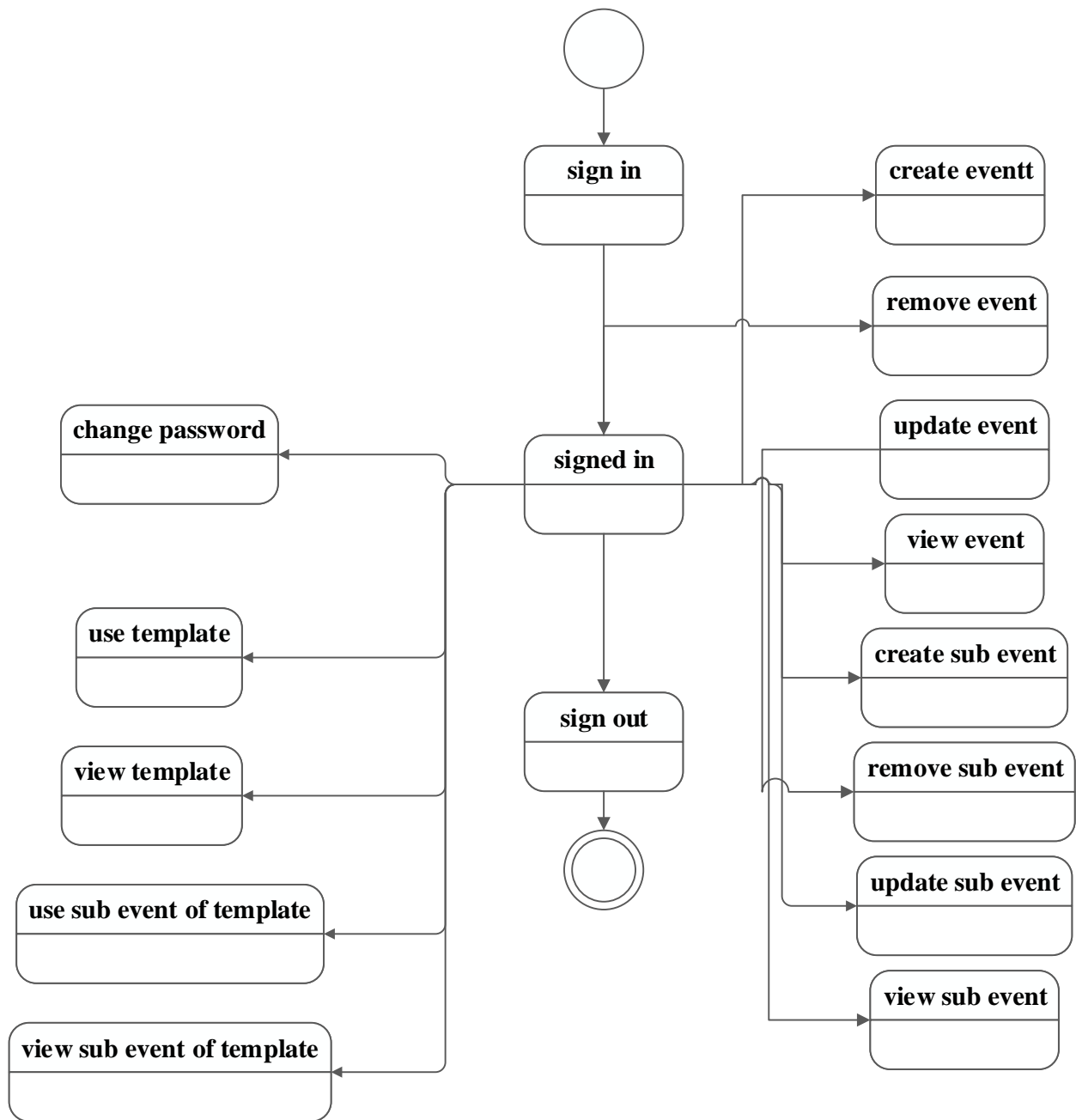


Figure 35: : State transition diagram of Student

## 8.4 Sequence Diagram

Sequence Diagram indicates how events cause transitions from object to object. It is actually a representation of how events cause flow from one object to another as a function of time.

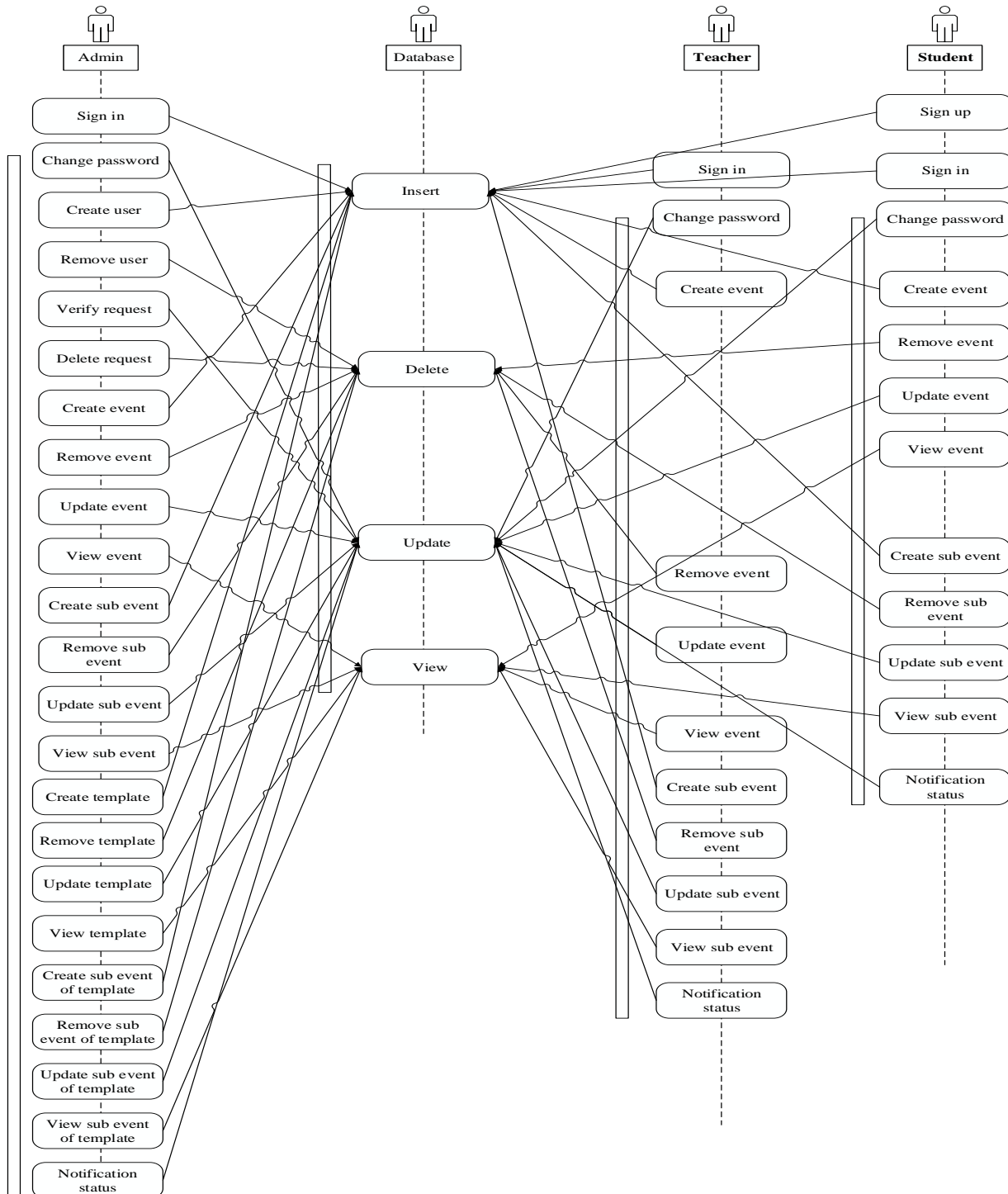


Figure 36: Sequence Diagram

## Chapter 9: Conclusion

Institute of Information Technology have such a busy schedule that students and faculty members are always conscious about events and tasks. But this consciousness is not always enough to guarantee the participation in events. Some of the websites like Facebook have features of creating events. But most of this websites are not made for professional people. Again, IIT demanding to be the Highest Echelon of Software Engineering in Bangladesh needed a mass customized website to manage their official events. Thus the service of our system comes into being. This SRS document can be used effectively to maintain software development cycle. It will be very easy to conduct the whole project using this SRS. Hopefully, this document can also help my junior BSSE batch students. We tried my best to remove all dependencies and make effective and fully designed SRS. We believe that reader will find it in order.

# References

- Books
  - Pressman, Roger S. Software Engineering: A Practitioner's Approach (7<sup>th</sup> Edition)
- URLs
  - <http://www.wikihow.com/Write-Software-Documentation> (Last Accessed: November 17, 2016)
  - [https://en.wikipedia.org/wiki/Activity\\_diagram](https://en.wikipedia.org/wiki/Activity_diagram) (Last Accessed: November 17, 2016)
  - [https://en.wikipedia.org/wiki/Use\\_Case\\_Diagram](https://en.wikipedia.org/wiki/Use_Case_Diagram) (Last Accessed: November 17, 2016)
  - [https://en.wikipedia.org/wiki/State\\_diagram](https://en.wikipedia.org/wiki/State_diagram) (Last Accessed: November 17, 2016)