



# Software Requirements Specification

On

A Learning Tool for Autistic Children

Supervised By

---

Dr. Mohammad Shoyaib

Professor

Institute Of Information Technology

University of Dhaka

Submitted By

Shakh Mohibul Alam  
Bsse1004

Mahdee Hasan  
Bsse1010

## Acknowledgement

We are highly indebted for getting such a wonderful opportunity to prepare the Software Requirements Specifications report of the project 'A Learning Tool For Autistic Children'. We would like to thank whole-heartedly our supervisor, Dr. Mohammad Shoyaib, Professor, Institute of Information Technology, University of Dhaka for giving us guidelines on preparing this report. We are also thankful to the students and teachers of "Communication Disorder" department of University of Dhaka, who gave requirements to accomplish this job.

## Abstract

Special children like those of having autism require special care, centralised effort in their learning and interaction with the world. Our study is to develop a system to smooth their task of learning and interaction by using the appropriate approach for them. So, the objective is to make an SRS (Software Requirements Specification) of 'A Learning Tool For Autistic Children' from the gathered requirements. The report represents the objective and also describes the need of this trendy strategy.

## Table of Contents

<b>Chapter 1</b>	<b>1</b>
Introduction	1
1.1 Purpose	2
1.2 Intended Audience	2
1.3 Conclusion	3
<b>Chapter 2</b>	<b>4</b>
Inception	4
2.1 Introduction	4
2.1.1 Identifying Stakeholders	4
2.1.2 Recognizing Multiple Viewpoints	5
2.1.3 Working towards collaboration	6
2.1.4 Asking the First Questions	8
2.2 Conclusion	8
<b>Chapter 3</b>	<b>10</b>
<b>Elicitation</b>	<b>10</b>
3.1 Introduction	10
3.2 Eliciting Requirements	10
3.3 Collaborative Requirements Gathering	11
3.4 Quality Function Deployment	11
3.4.1. Normal Requirements	11
3.4.2. Expected Requirements	12
3.4.3 Exciting Requirements	13
3.5 Usage Scenario	13
<b>Chapter 4</b>	<b>16</b>
<b>Scenario Based Modeling</b>	<b>16</b>
4.1 Introduction	16
4.2 Definition of Use Case	16
Primary Actor	17
Secondary Actor	17
4.3 Use Case Diagrams	17
4.3.1 Level-0 Use Case Diagram - A Learning Tool for Autistic Children	18
4.3.2 Level-1 Use Case Diagram - System	19
4.3.3 Level-1 Use Case Diagram - Authentication	20
4.3.4 Level-1.1.1 Use Case Diagram - Sign up	21

4.3.5 Level-1.1.2 Use Case Diagram - Sign in	22
4.3.6 Level-1.2 Use Case Diagram - Student info management	24
4.3.7 Level-1.2.1 Use Case Diagram - Add student info	27
4.3.8 Level-1.3 Use Case Diagram - Lesson management	28
4.3.9 Level-1.3.1 Use Case Diagram - Add new lesson	29
4.3.10 Level-1.3.2 Use Case Diagram - Add lesson elements	32
4.3.10 Level-1.4 Use Case Diagram - Show lesson module	34
4.4 Activity Diagrams of Study Learning Tool for Autistic Children	35
4.4.1 A Learning Tool for Autistic Children Activity Diagram	35
4.4.2 Sign up Activity Diagram	36
4.4.3 Sign in Activity Diagram	38
4.4.4 Change password Activity Diagram	39
4.4.5 Student info management Activity Diagram	40
4.4.6 Add student info Activity Diagram	41
4.4.7 Lesson management Activity Diagram	42
4.4.8 Add new lesson Activity Diagram	43
4.4.9 Add lesson elements Activity Diagram	44
4.4.10 Show lesson module Activity Diagram	45
4.5 Swim Lane Diagrams of Study Tool for Autistic Children	46
4.5.1 Study Tool for Autistic Children Swimlane Diagram	47
4.5.2 Sign up Swimlane Diagram	48
4.5.3 Sign in Swimlane Diagram	49
4.5.4 Change password Swimlane Diagram	50
4.5.5 Student management Swimlane Diagram	51
4.5.6 Add student info Swimlane Diagram	52
4.5.7 Lesson management Swimlane Diagram	53
4.5.8 Add new lesson Swimlane Diagram	54
4.5.9 Add lesson elements Swimlane Diagram	55
4.5.10 Show lesson module Swimlane Diagram	56
<b>Chapter 5</b>	<b>58</b>
<b>DATA MODELLING</b>	<b>58</b>
5.1 Data Modelling Concept	58
5.1.1 Data Objects	58
5.1.1.1 Noun Identification	58
5.1.1.2 Potential Data Objects	61
5.1.1.3 Analysis for finalizing Data Objects	61
5.1.1.4 Final Data Objects	62
5.2 DATA OBJECT RELATIONSHIPS	62
5.3 ENTITY RELATIONSHIP DIAGRAM	64

5.4 SCHEMA DIAGRAM	65
<b>Chapter 6</b>	<b>68</b>
<b>CLASS-BASED MODELING</b>	<b>68</b>
6.1 CLASS BASED MODELING CONCEPT	68
6.2 Noun List	68
6.3 General Classification	69
6.4 Selection Criteria	72
6.4 Attribute Selection	73
6.5 Method Identification	74
6.6 Analyzing Classes	75
6.7 Class Card	75
6.8 CRC Diagram	79
<b>Chapter 7</b>	<b>81</b>
<b>Behavioral Modeling</b>	<b>81</b>
7.1 State Transition Diagram	81
7.1.1 Event Identification	81
7.1.2 Events After Analysis	85
7.1.3 State Transition Diagrams	88
Teacher	88
Student	89
Lesson	90
Lesson Element	91
Show Lesson Module	92
Database	93
7.2 Sequence Diagram	94





# Chapter 1

## Introduction

This chapter is a part of “A Learning Tool for Autistic Children” intended to specify the purpose of this document and the intended audience of it.

### 1.1 Purpose

This document is the Software Requirements Specification (SRS) for “A Learning Tool for Autistic Children”. It contains detailed functional, non-functional, and support requirements and establishes a requirements baseline for development of the system. 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 managers, designers, developers, and testers.

- ❖ The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.
- ❖ The project managers 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 designers will use this SRS as a basis for creating the system’s design. The designers will continually refer back to this SRS

to ensure that the system they are designing will fulfill customer's needs.

- ❖ The developers will use this SRS as a basis for developing the system's functionality. The developers 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 testers will use this SRS to derive test plans and test cases for each documented requirement.

## 1.3 Conclusion

This analysis of the audience helped us to focus on the users who will be using our product. This overall document will help each and every person related to this project to have a better idea about the project.

# Chapter 2

## Inception

In this chapter, the Inception part of the SRS will be discussed briefly.

### 2.1 Introduction

Inception is the beginning phase of requirements engineering. It focuses on how a software project gets started and the scope and nature of the problem to be solved. The goal of the inception phase is to identify individual and common needs and conflict requirements among the stakeholders of a software project. To establish the groundwork we have worked with the following factors related to the inception phases:

- Identifying Stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the first question

#### 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 who are involved with its installation. The following stakeholders were identified at the inception phase for the “A Learning Tool for Autistic Children”.

❑ **Client:** As the requirements of this project is given by the “Communication Disorder” department of University of Dhaka, they are our main clients. They needed the technical support.

❑ **End-user:** The end users are who will be using the software. Here, we are making the software for the special children who are unable to make social interaction easily. They are unable to learn from the surroundings. So this software will be a tool for them to make learning easier for them.

❑ **Software Developer:** A software developer is concerned with facets of the software development process, including the research, design, programming, maintenance and testing of computer software. She will be responsible for the outcomes of the software

### 2.1.2 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

#### → **Client’s Viewpoint:**

- Error free system
- Attractive Graphical User Interface for Autistic Children
- Making questions easier
- User friendly
- Making puzzles easier
- User’s overall improvement tracking
- Most used objects from the surroundings should be taught

→ **User's Viewpoint:**

- Easy to use
- Error free system
- Keeping track of user's improvement

→ **Software Developer's Viewpoint:**

- Easy to built
- Error free effective software
- No conflicting requirement
- Getting a decent amount of money for project budget

### 2.1.3 Working towards collaboration

While working with different stakeholders, some conflicting and common viewpoints can be noticed. For this reason, final requirements can be gotten by the collaboration between the stakeholders. We followed following steps to merge these requirements:

- ❖ Identify the common and conflicting requirements
- ❖ Categorize the requirements

- ❖ Giving priority to the most useful/important features if conflicts arise
- ❖ Make final decision about the requirements

#### Common Requirements:

1. Standalone Desktop Software
2. 100 learning objects for autistic children
3. Database to store user's response data
4. Attractive graphical user interface for uses

#### Conflicting Requirements:

1. Authentication System required or not
2. Which parameters should be chosen to measure improvement

#### Final Requirements:

1. Error free easy accessible software
2. Desktop standalone software
3. Attractive graphical user interface
4. Sign up and Login required
5. 100 learning objects selected from surroundings
6. User able to learn through video, audio and images
7. Maintain a database for all user

8. Allow user to check improvement
9. User answers question regarding objects
10. User can solve puzzle

## 2.1.4 Asking the First Questions

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

## 2.2 Conclusion

The Inception phase helped us to establish basic understanding about “A Learning Tool Autistic Children”. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.





# Chapter 3

## Elicitation

### 3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering requirements from the users, customers and other stakeholders.

In our case, we faced some problems in requirements elicitation like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

### 3.2 Eliciting Requirements

The main task of this phase is to combine the elements of problem solving, elaboration, negotiation and specification. The collaborative working approach of the stakeholders is required to elicit the requirements. The following tasks were done for eliciting requirements:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. Usage Scenarios
4. Elicitation work products

### 3.3 Collaborative Requirements Gathering

We are doing the project by collaborating with the 'Communication Disorder' department of University of Dhaka. We conducted meetings with them and they gave us requirements over email. We completed the following steps to do it.

- The meetings were conducted with the students and teachers of the Communication Disorder Department of University Of Dhaka and they were questioned about the requirements and expectations.
- They told us about the purpose of the Software.
- At last we selected our final requirements from the meetings

### 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 our project the following requirements are identified by a QFD.

#### 3.4.1. Normal Requirements

The normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers' satisfaction. These are the normal requirements for the project.

1. Standalone Software
2. User needs to authenticate to access the system
3. Multiple instance of users can be created
4. User can select module they want to use
5. Learn objects through images
6. Learn objects through videos
7. Sound-clip regarding the object to know the object

### 3.4.2. Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

1. Error free software
2. Strong authentication system
3. User friendly
4. Effective system
5. No ambiguous feature
6. Questions regarding the objects user has already learnt
7. Storing user's response to the questions

8. Storing user's response time to the questions

### 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. Puzzles regarding the objects user has already learnt
2. Storing user's solving time of the puzzles
3. Keep track of improvements of the users

## 3.5 Usage Scenario

This application is for specially abled childs . Teachers can make lessons with digital contents and teach vocabulary to special children.

Authentication is required for teachers to interact with this application. A teacher has to sign up with his information including name ,username and password . A teacher has to create his/her student list with proper student information including name ,image and reward list. Later the teacher can add or remove students from his/her student list.

Teachers can make lessons to represent learning objects. Every lesson has a name and thumbnail which is an image type to easily identify the lesson.

Every lesson will be made with any number of elements while the types of elements are noun,verb,association and activity.

To design any type of lesson element , element category (noun,verb,association,activity) , element name, type ,word(text to represent the vocabulary ) are required . Teachers can select image ,video or audio along with text to present a new word . There will be a check box

in the noun element part for association . If the teacher selects the check box , he has to design the other type elements and present the association between the noun and verb . An activity element presents a daily activity .For example brushing teeth . For the activity element , an activity name and video is required to present a daily activity . After making a lesson, all the lesson related data will be stored in the database.

In this application , there is a show lesson module where one can view the lesson contents . In the show lesson module , while a user completes the lesson , some quizzes will be automatically appeared by the system . There two types of quizzes will be appeared .These are Jigsaw puzzle and mcq with the images . Both jigsaw puzzle and mcq quiz will be made from the lesson images .

After making a lesson, System will create a lesson type file with the “.lesson ” extension which is recognizable to this app. The teacher makes a lesson and provides the lesson file to the student. A student can enter this application with his name . The student can open the lesson file and see the lesson .

To learn objects, students can show images of lessons. Students can view 4 or 5 images of a particular lesson. S/he can navigate to the previous/next image of that particular lesson by pressing the PREVIOUS/NEXT button. And at the beginning of each lesson, there will be an additional button named SKIP, pressing it will take the user to a new object skipping the current lesson. Students can play audio and video to understand a lesson. For the first time, audio and video will play automatically, by clicking the reply button one can replay the audio and video.

After viewing the images of the lesson, a question window will appear regarding the object and the student will have to answer it. The user will be shown if his/her answer was correct or wrong.Point based on correct /wrong answer will be stored against the student id. Their response and response time will be stored for their improvement measurement later. The

same question will come to the student several times but not at a time. Every time their response and response time will be measured.

Students can play a jigsaw puzzle based on the objects they have learned. Jigsaw puzzles involve constructing an object by dragging its pieces and assembling them in an exact place. When a Student solves the puzzle he's shown how much time he's taken this time to solve the puzzle. The same puzzle will come to the Student several times. Every time his/her solving time will be stored.

Response time in the quiz part will be recorded in the student list for every student. This response time will indicate the outcome of the learning.

Teachers also can set rewards at the end of the lesson based on the student's preferences. Rewards can be many types for example music, image, cartoon, animation, video or audio. Students who like animation, an animation is automatically played when s/he will be able to succeed in the quiz section.

Teachers can evaluate every student based on their point and response time in the quiz section. To evaluate a student, the teacher will have to collect the database file from the student computer and make a decision for the student.

# Chapter 4

## Scenario Based Modeling

This chapter describes the Scenario Based Model for the “A Learning Tool for Autistic Children”.

### 4.1 Introduction

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swimlane diagrams.

### 4.2 Definition of Use Case

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

## Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software.

## Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

## 4.3 Use Case Diagrams

A Use Case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different Use Cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Use Case diagrams to give a non-technical view of the overall system.



### 4.3.1 Level-0 Use Case Diagram - A Learning Tool for Autistic Children

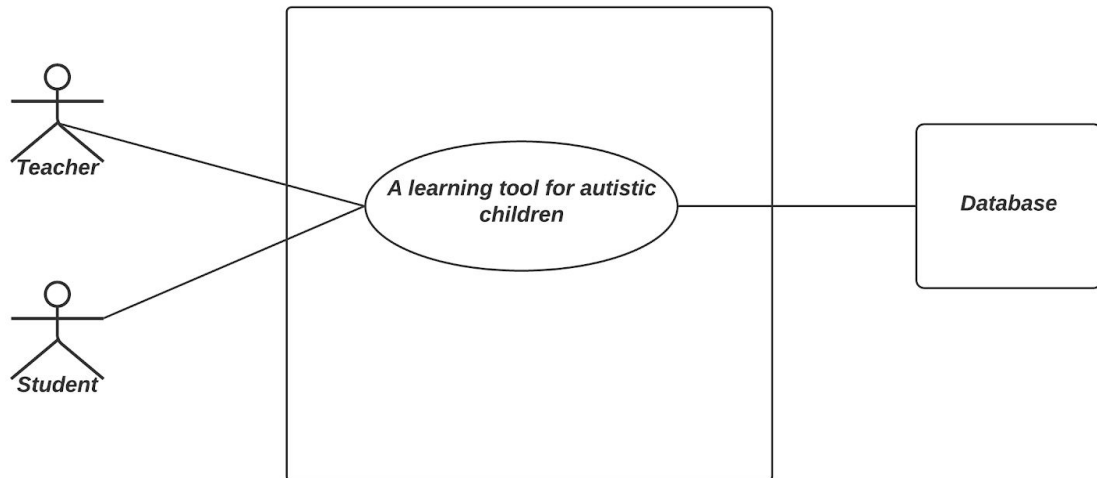


Figure 1: Level-0 A Learning Tool for Autistic Children.

#### **Description of Use Case diagram level-0:**

After analyzing the user's story, we found 2 actors that directly or indirectly with the system. Primary actors are those who will play action and get replies from the system whereas secondary actors only produce or consume information. The actors are -

- Teacher
- Student

### 4.3.2 Level-1 Use Case Diagram - System

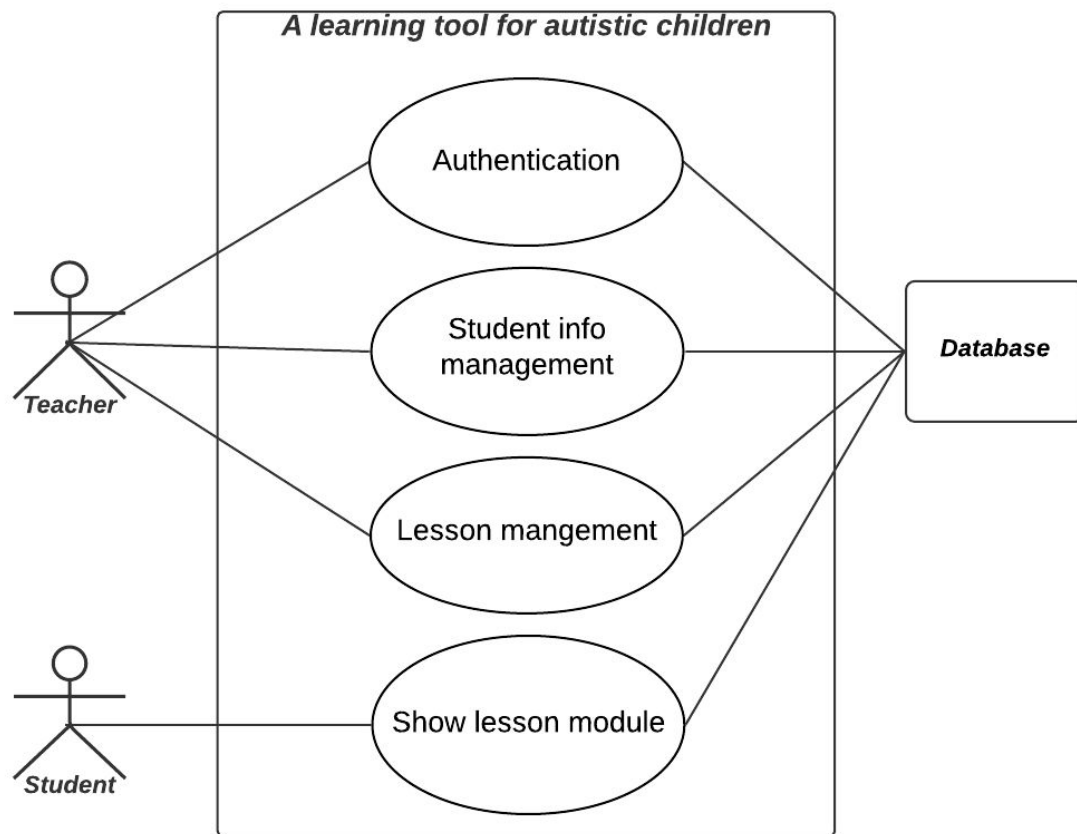


Figure 2: Level-1 System.

#### **Description of Use Case diagram level-1:**

There are four sub-system in “ Learning Tool for Autistic Children”. They are follows :

1. Authentication
2. Student management
3. Lesson management

#### 4. Show lesson module

The four subsystems are further decomposed, in level 1.1, 1.2, 1.3, 1.4 respectively.

### 4.3.3 Level-1 Use Case Diagram - Authentication

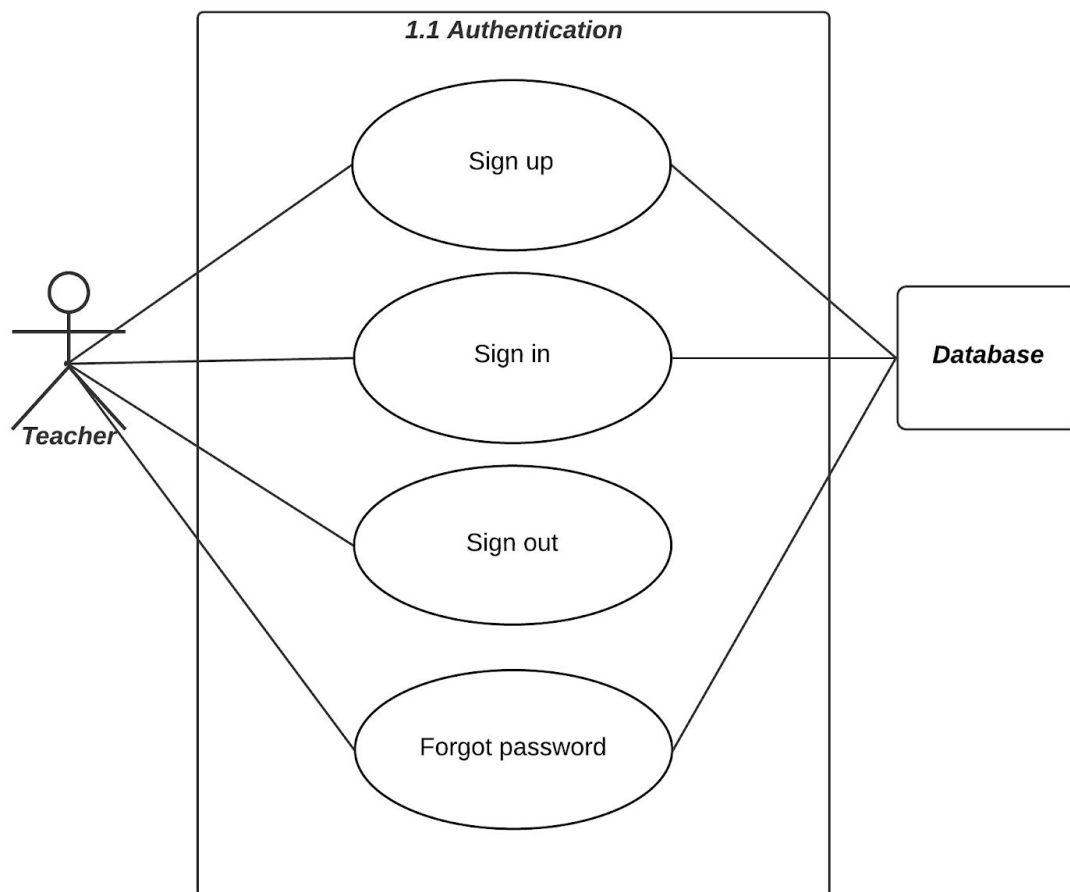


Figure 3: Level-1.1 Authentication.

#### Description of Use Case diagram level-1.1:

Authentication is a process in which credentials provided are compared to those on files in a database of authorized user's information. The authentication subsystem can be divided into two parts. They are as follows:

1. Creating Account
2. Accessing Account

The first two subsystems are further decomposed, in level 1.1.1 and 1.1.2 respectively.

#### 4.3.4 Level-1.1.1 Use Case Diagram - Sign up

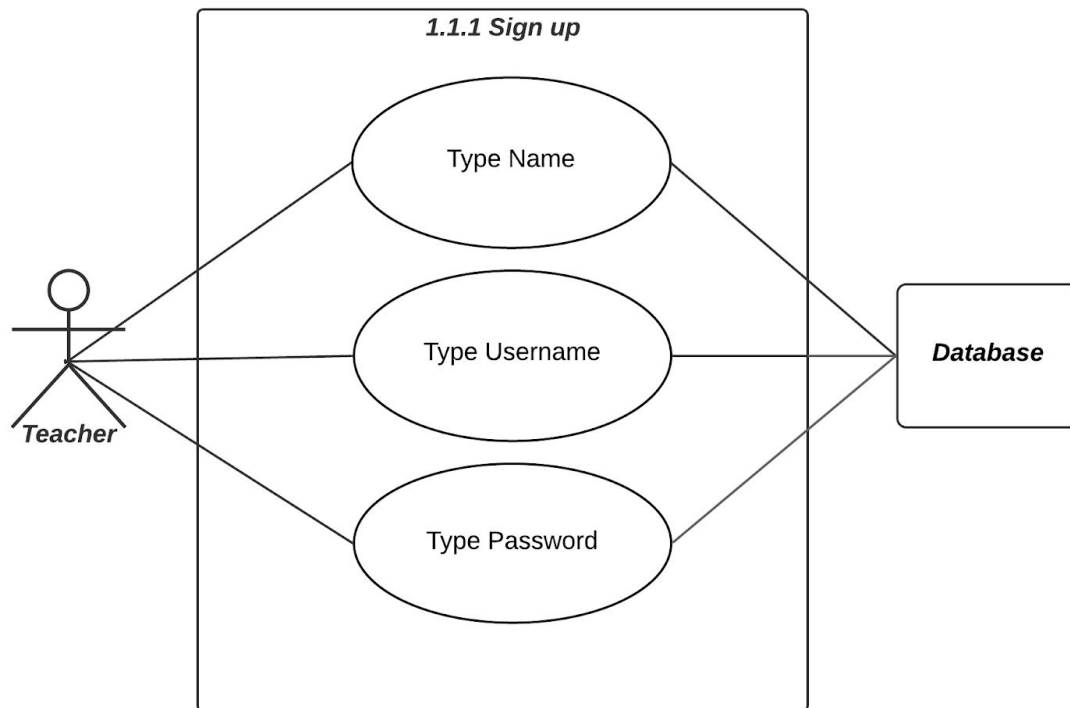


Figure 4: Level-1.1.1 Sign up.

**Description of Use Case diagram level-1.1.1:**

Authentication is required for teachers to interact with this application. A new teacher has to give his/her name, username and password to sign up. Authentication is not necessary for the student(specially-abled child). Here the username given by the teacher should be unique.

**Action and Reply of Use Case diagram level-1.1.1:****Teacher:**

- A1: Teacher creates an account filled with valid information.  
R1: System creates an administrator account and the account information is stored.
- A2: Teacher creates an account filled with invalid information.  
R2: System allows the shopkeeper to try again for account creation.

#### 4.3.5 Level-1.1.2 Use Case Diagram - Sign in

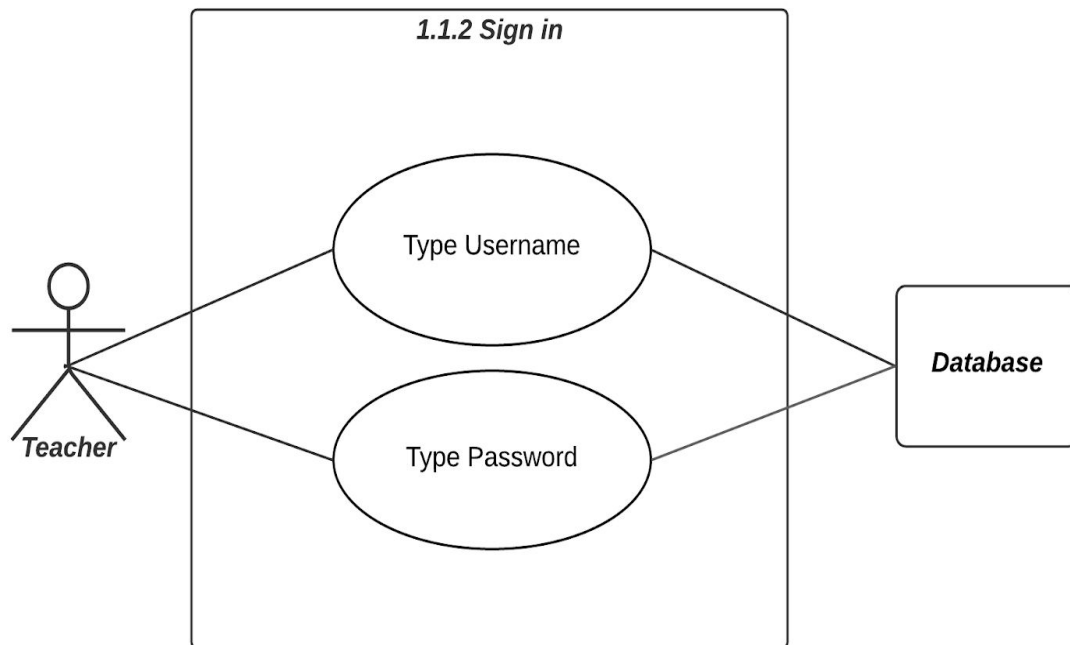


Figure 5: Level-1.1.2 Sign in.

**Description of Use Case diagram level-1.1.2:**

Username and password are required for sign in. If a teacher forgets his/her password, he/she can change his/her password using his username. For changing passwords, teachers should give twice the password for confirming password. Students don't need to sign in to the system.

**Action and Reply of Use Case diagram level-1.1.2:**

## Teacher:

- A1: Teacher provides username and password.  
R1: System will check validity. A valid information system will allow access.
- A2: Teacher provides invalid information.  
R2: System will show error messages and allow him/her to try again.
- A3: Teacher gives commands for log out from the system.  
R3: System will search for unsaved file(s). If an unsaved file(s) is found, the system will ask for confirmation whether the account should be logged out or not. Otherwise, the account will be logged out automatically.
- A4: Teacher forgets his/her password, he/she will change password. To change the password, the teacher needs to give his valid username, type new password and retype password.  
R4: System will allow the user to change the password. System will want the user username to confirm whether the username exists or not. After that system will check the password and retype password are the same or not. Then the system will store updated information to the database.

## 4.3.6 Level-1.2 Use Case Diagram - Student info management

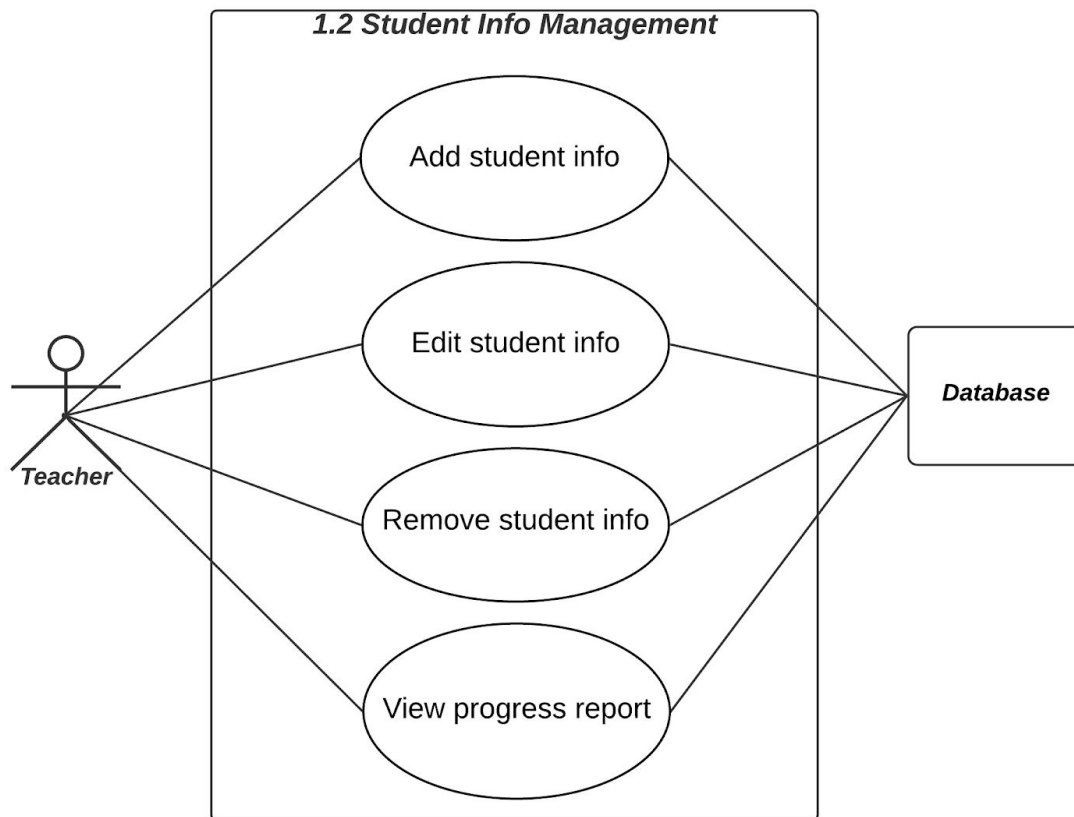


Figure 6: Level-1.2 Student info management.

### **Description of Use Case diagram level-1.2:**

Teachers can add, edit and remove information from the student list. Teachers can also view students progress reports from the system. Based on the report teachers can evaluate the students. The first subsystem is further decomposed, in level 1.2.1.

### **Action and Reply of Use Case diagram level-1.2:**



## Teacher:

- A1: Teacher provides required student's information to add to the student list.  
R1: System will check validity. A valid information system will allow store information.
- A2: Teacher provides invalid student's information.  
R2: System will show error messages and allow him/her to try again.
- A3: Teacher gives commands to remove student from the student list.  
R3: System will remove the student from the student list and store updated information.
- A4: Teacher gives commands to edit student's information.  
R4: System will store updated information.
- A5: Teacher wants to view the student progress report to evaluate the student.  
R5: System will show student response time and point in the given test.

### 4.3.7 Level-1.2.1 Use Case Diagram - Add student info

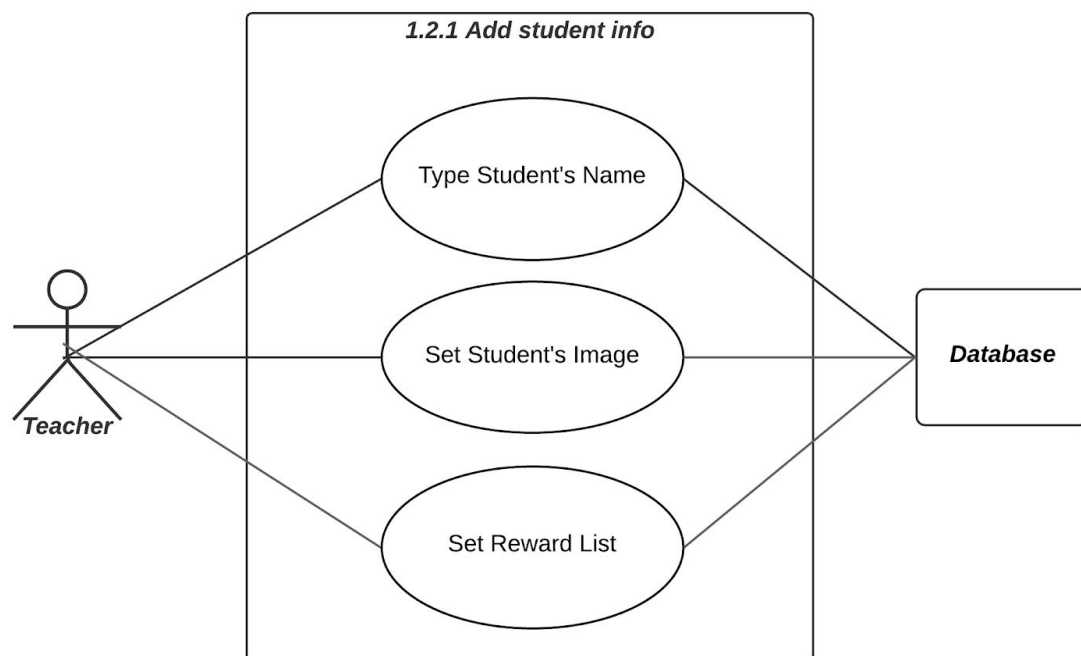


Figure 7: Level-1.2.1 Add student info.

#### **Description of Use Case diagram level-1.2.1:**

Teacher gives required information to add the student to the student list. Student's name, student's image and reward list are required for each

student to add to the student list. Teacher gives those information to add the student to the student list.

### Action and Reply of Use Case diagram level-1.2.1:

#### Teacher:

- A1: Teacher provides required student's information to add to the student list.  
R1: System will check validity. A valid information system will allow store information.
- A2: Teacher provides invalid student's information.  
R2: R2: System will show error messages and allow him/her to try again to provide correct information.

### 4.3.8 Level-1.3 Use Case Diagram - Lesson management

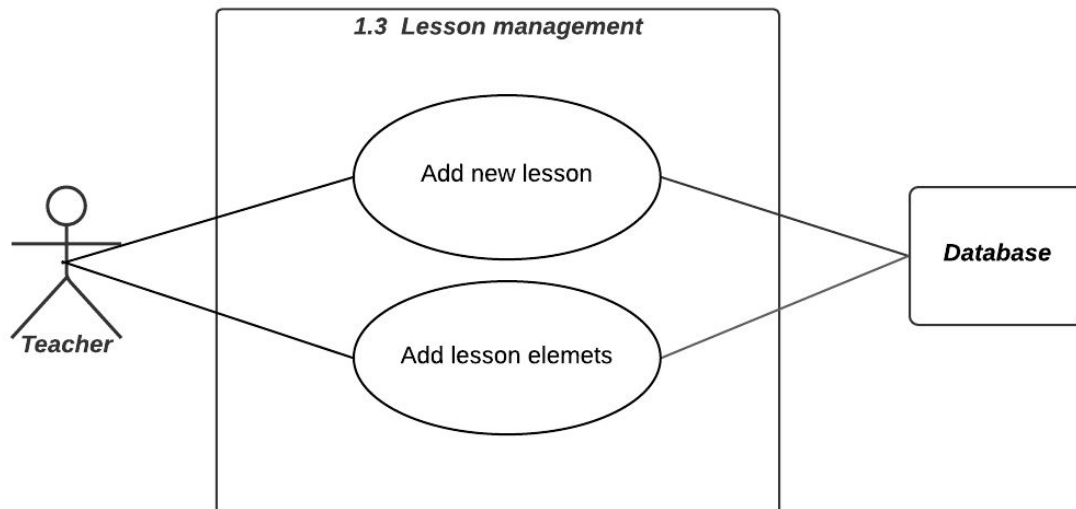


Figure 8: Level-1.3 Lesson management.

**Description of Use Case diagram level-1.3:**

Teachers can add new lessons in the system which is used in the learning of specially abled children. Teacher needs to give a unique lesson element's name to create a new lesson element. There are various types of lesson elements. Then teachers add lesson contents.

The two subsystems are further decomposed, in level 1.1.1 and 1.1.2 respectively.

**Action and Reply of Use Case diagram level-1.3:****Teacher:**

- A1: Teacher provides required information for lesson.  
R1: System will check validity of lesson's unique name. A valid information system will allow store information and create a new lesson.
- A2: Teacher provides existed lesson's name.  
R2: R2: System will show error messages and allow him/her to try again to provide a unique lesson's name.

#### 4.3.9 Level-1.3.1 Use Case Diagram - Add new lesson

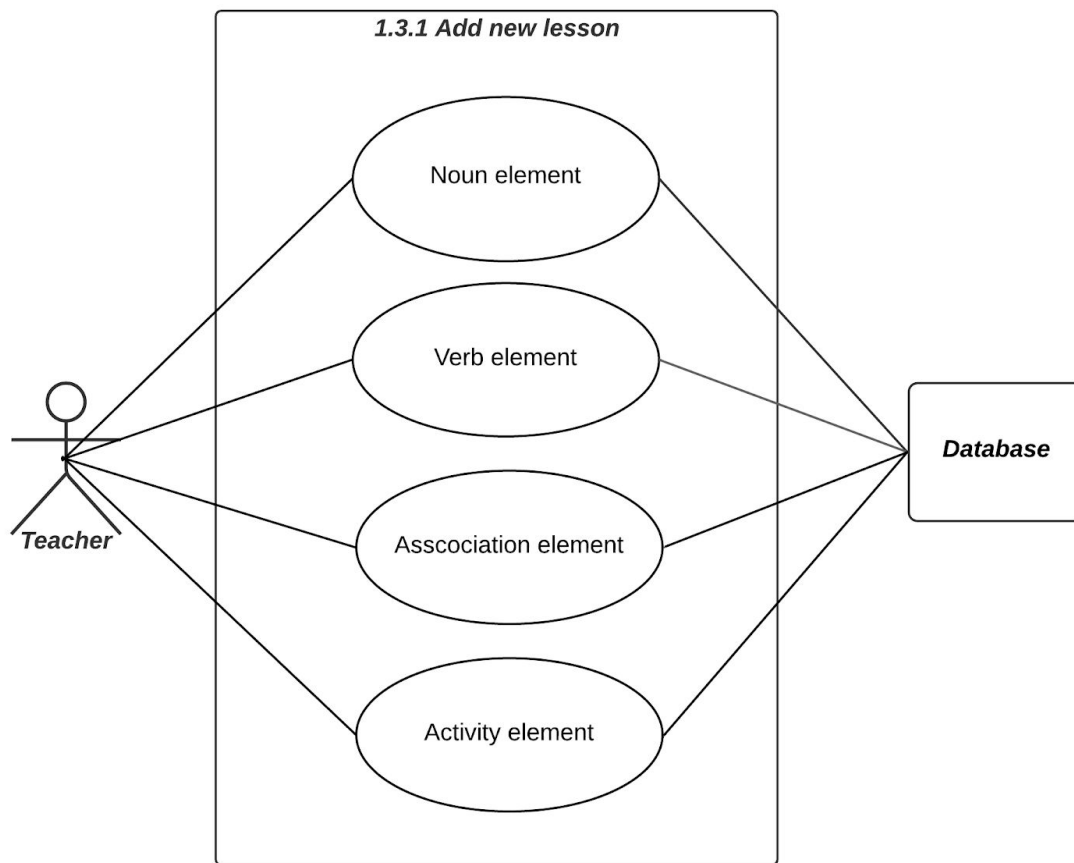


Figure 9: Level-1.3.1 Add new lesson.

### **Description of Use Case diagram level-1.3.1:**

Teachers can add new lessons in the system. There are four types of elements. They are:

1. Noun element
2. Verb element
3. Associate element
4. Activity element

Teacher creates noun and verb elements individually. But associate elements need two types of elements. Using verb and noun elements, the

teacher creates an associate element. Teachers also can create activity elements to teach the students how to do their daily activities.

### **Action and Reply of Use Case diagram level-1.3:**

#### **Teacher:**

- A1: Teacher provides unique lesson's name.  
R1: System will check validity of lesson's unique name. A valid information system will allow store information and create a new lesson.
  
- A2: Teacher chooses type of element.  
R2: System will create that type of lesson element.
  
- A3: Teacher gives commands to create associate elements.  
R3: System will create associate elements using noun and verb elements.

### 4.3.10 Level-1.3.2 Use Case Diagram - Add lesson elements

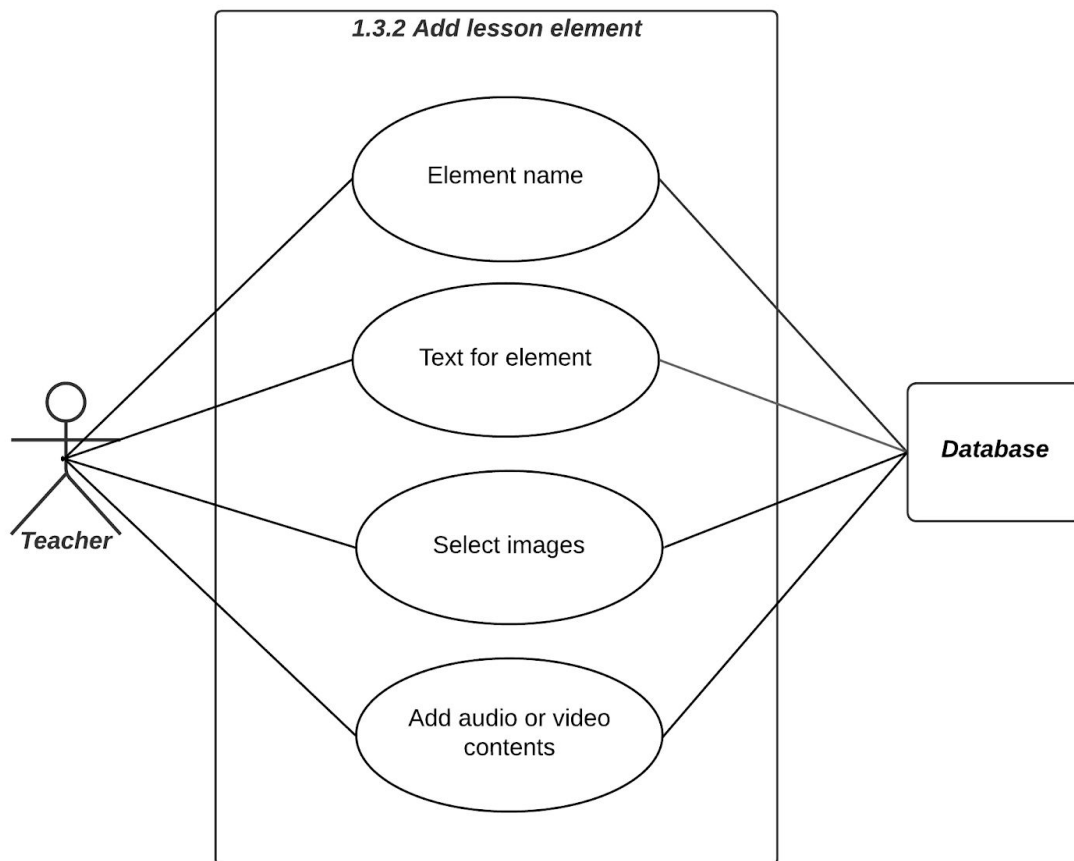


Figure 10: Level-1.3.2 Add lesson elements.

### **Description of Use Case diagram level-1.3.2:**

Teachers can add lesson elements in the new lesson. If a teacher wants to create an activity element, he/she can only add the lesson's name and video. In other elements, the teacher adds text, images, audio or video for the element.

### **Action and Reply of Use Case diagram level-1.3.2:**

#### **Teacher:**

- A1: Teacher provides unique lesson's name.  
R1: System will check validity of lesson's unique name. A valid information system will allow store information and create a new lesson.
  
- A2: Teacher chooses activity element.  
R2: System will give only video to select the user.
  
- A3: Teacher chooses a noun or verb element.  
R3: System will allow images, text , audio or video for the element to select.
  
- A4: Teacher chooses an associate element.  
R4: System will only allow a noun and a verb lesson element to select.



### 4.3.10 Level-1.4 Use Case Diagram - Show lesson module

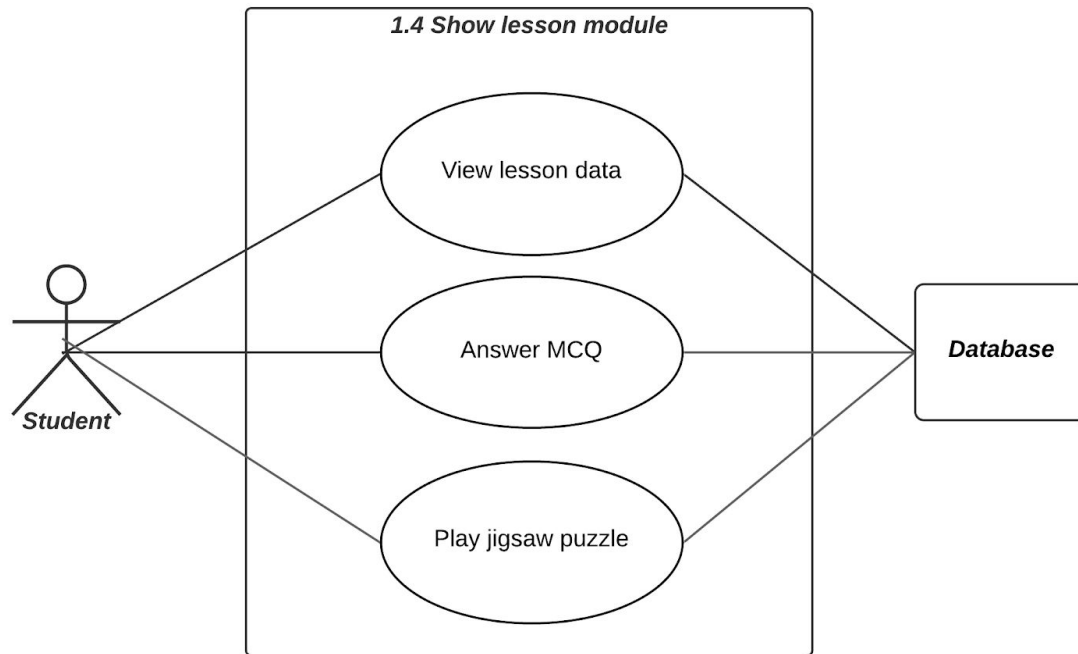


Figure 11: Level-1.4 Show lesson module.

#### **Description of Use Case diagram level-1.4:**

Students open the “.lesson” extension file. Then the student starts to view the lesson and learn from that lesson. After that the student is given some MCQ quiz and Jigsaw puzzle for testing. The response time and point of the student will be stored in the database.

#### **Action and Reply of Use Case diagram level-1.4:**

##### **Student:**

- A1: Student clicks the “.lesson” extension file.  
R1: System will open the lesson with the existing contents.

- A2: Student finishes the lesson.  
R2: System will give the student to answer some MCQ quiz and solve the Jigsaw puzzle.
- A3: Student gives the test about the lesson contents.  
R3: System will store the response time and point of the test which is taken.

## 4.4 Activity Diagrams of Study Learning Tool for Autistic Children

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

### 4.4.1 A Learning Tool for Autistic Children Activity Diagram

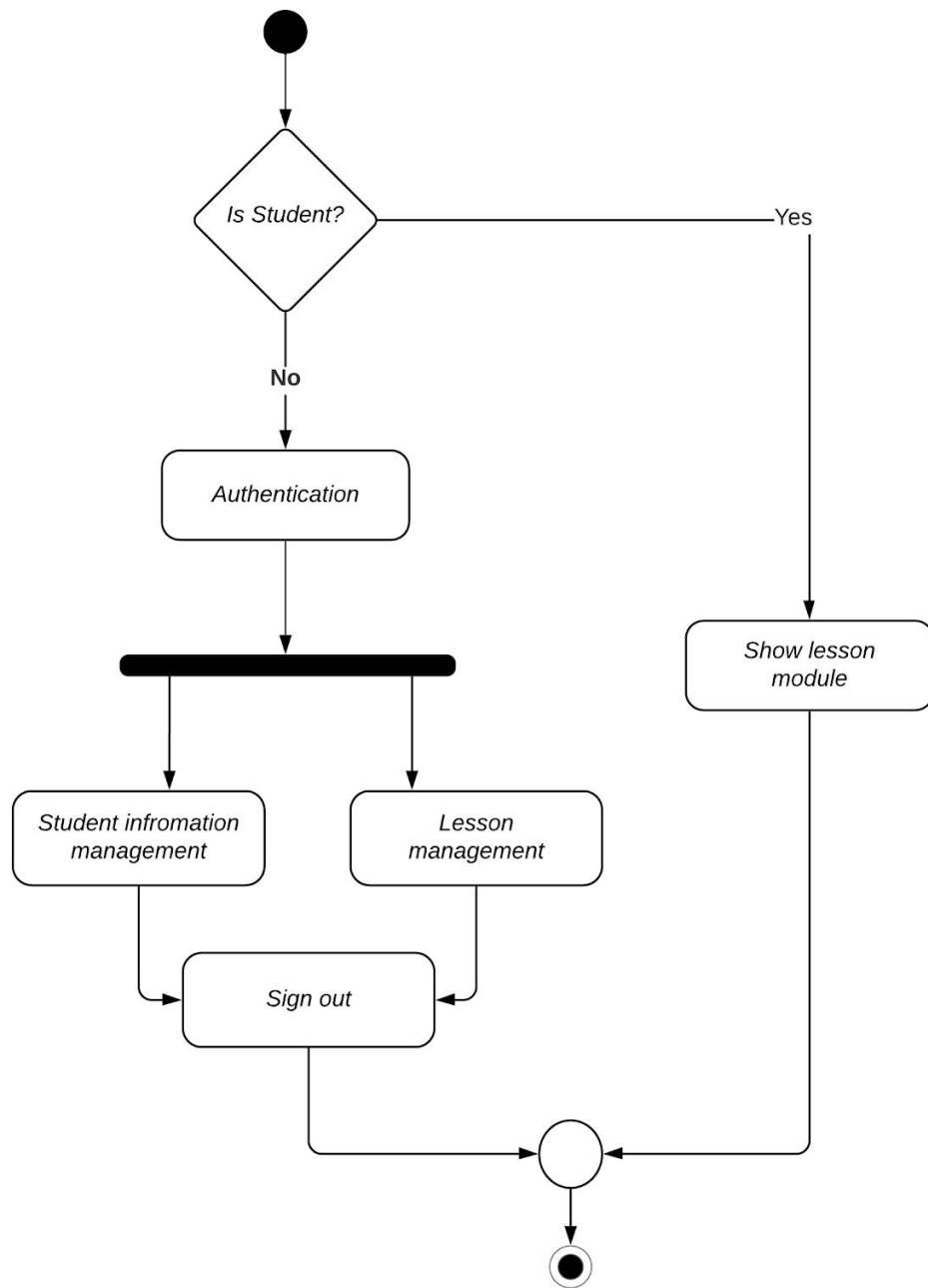


Figure 12: Activity diagram- Main.

#### 4.4.2 Sign up Activity Diagram

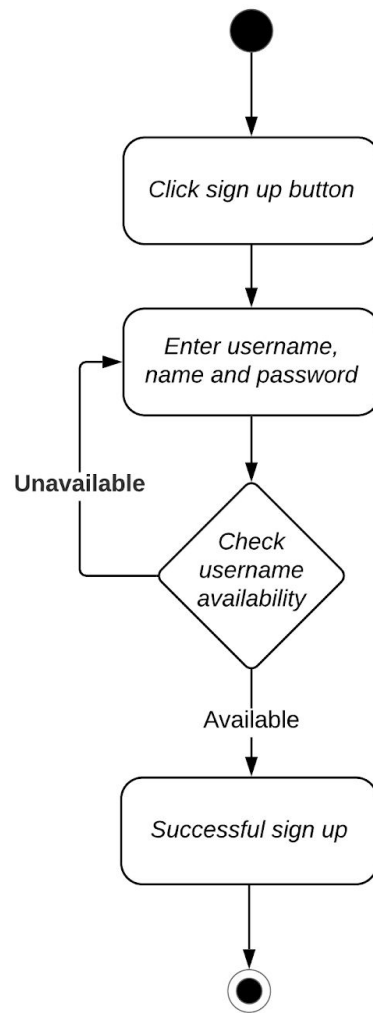


Figure 13: Activity diagram- Sign up.

### 4.4.3 Sign in Activity Diagram

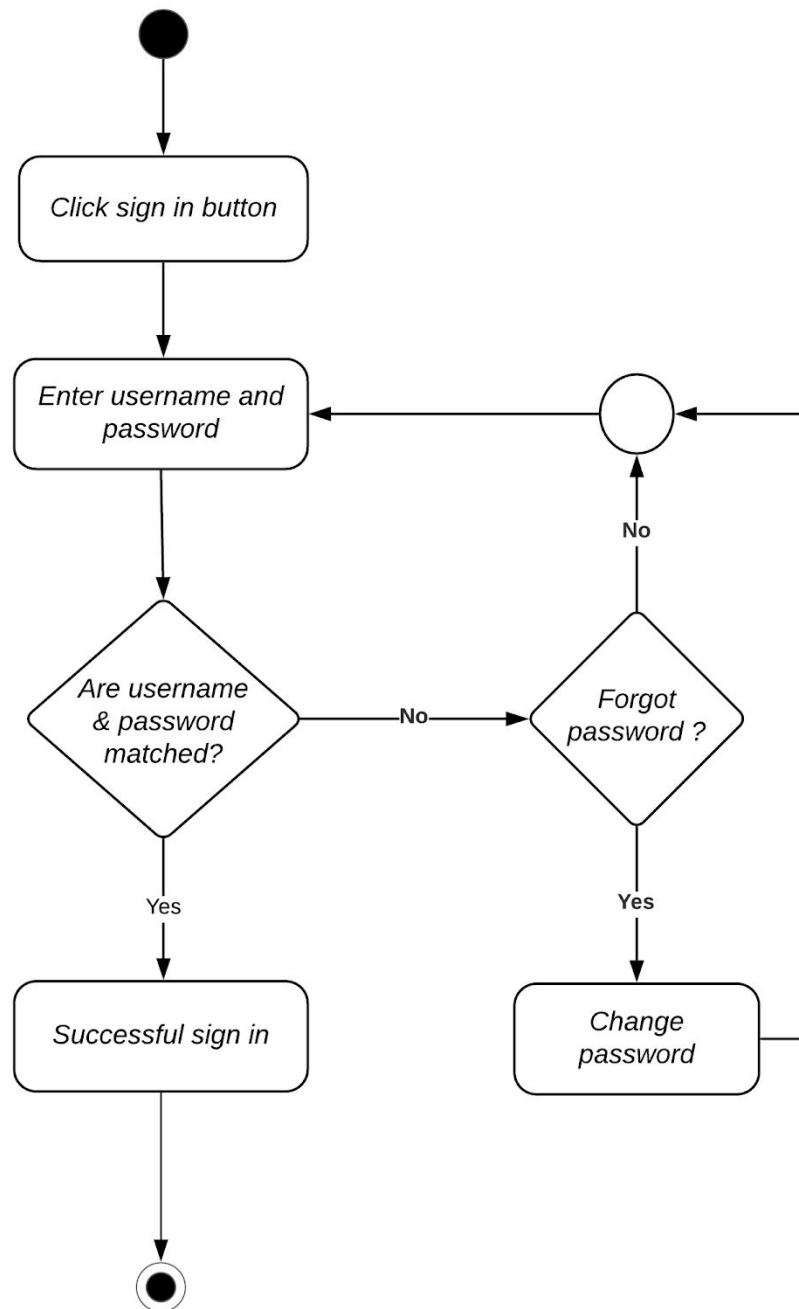


Figure 14: Activity diagram- Sign in.

#### 4.4.4 Change password Activity Diagram

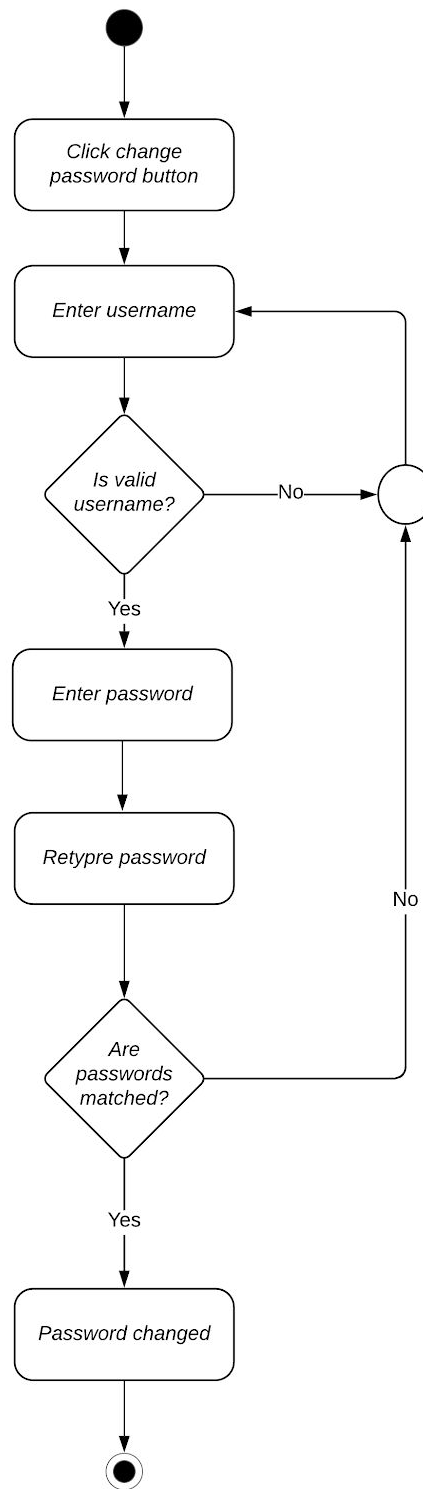


Figure 15: Activity diagram- Change password.

#### 4.4.5 Student info management Activity Diagram

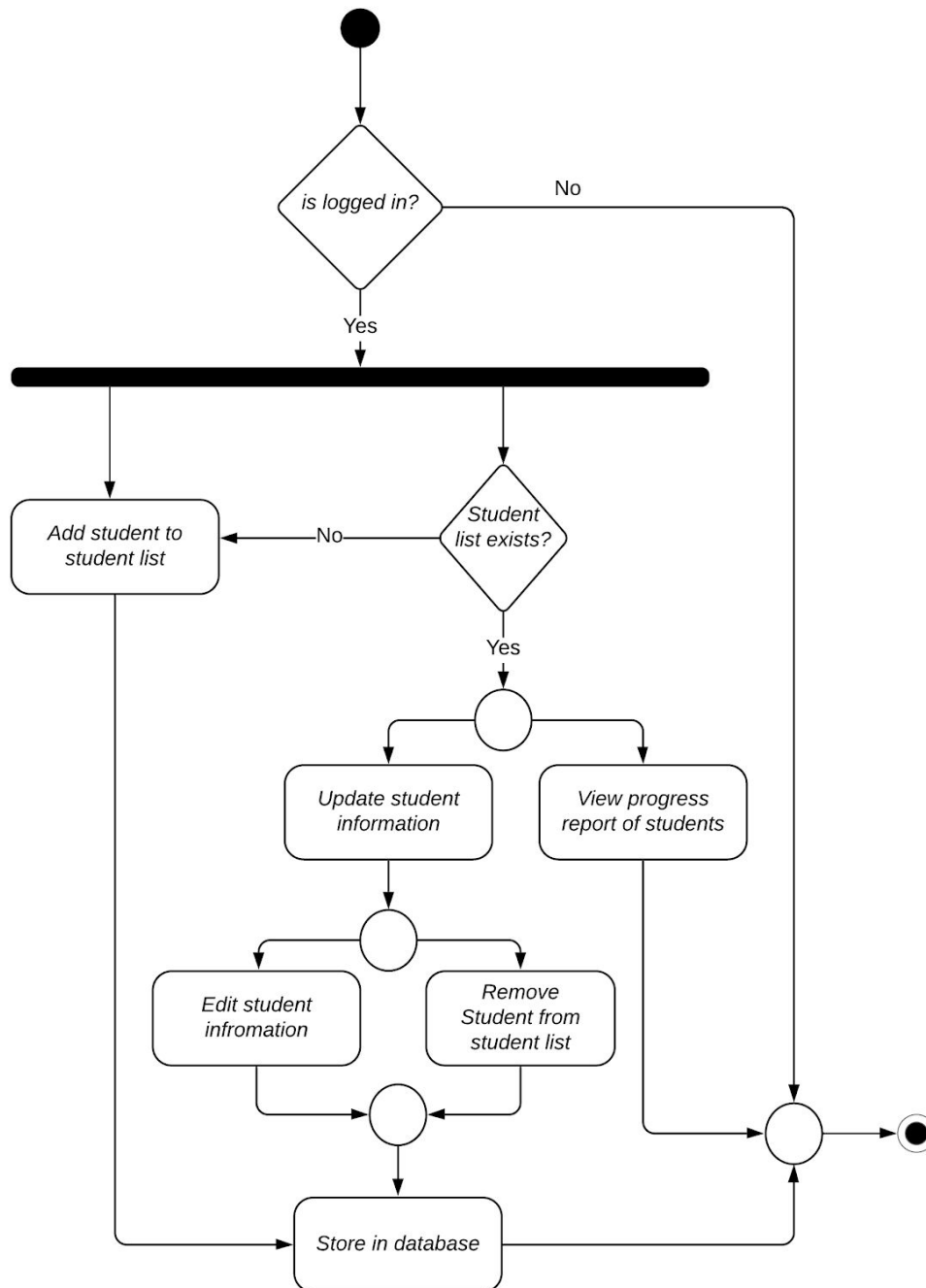


Figure 16: Activity diagram- Student info management.

#### 4.4.6 Add student info Activity Diagram

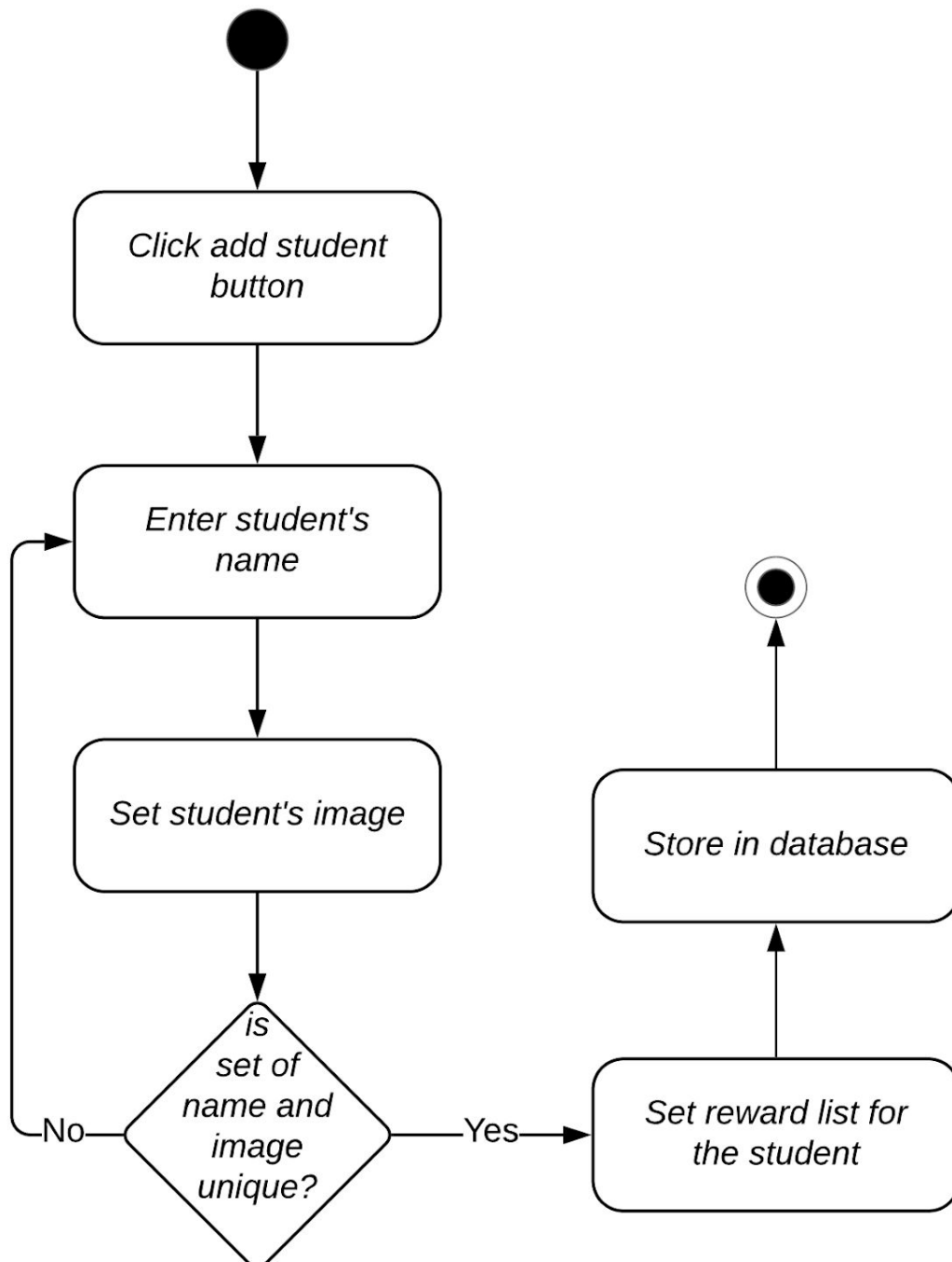


Figure 17: Activity diagram- Add student info.



#### 4.4.7 Lesson management Activity Diagram

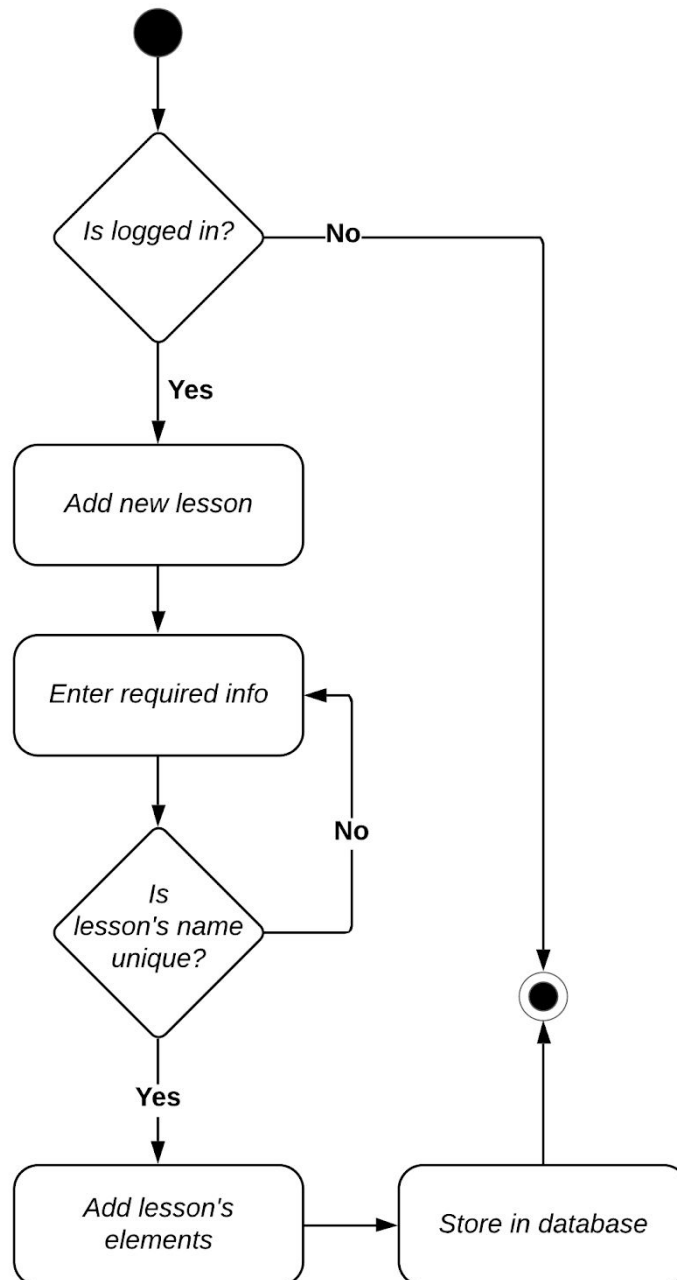


Figure 18: Activity diagram- Lesson management.

#### 4.4.8 Add new lesson Activity Diagram

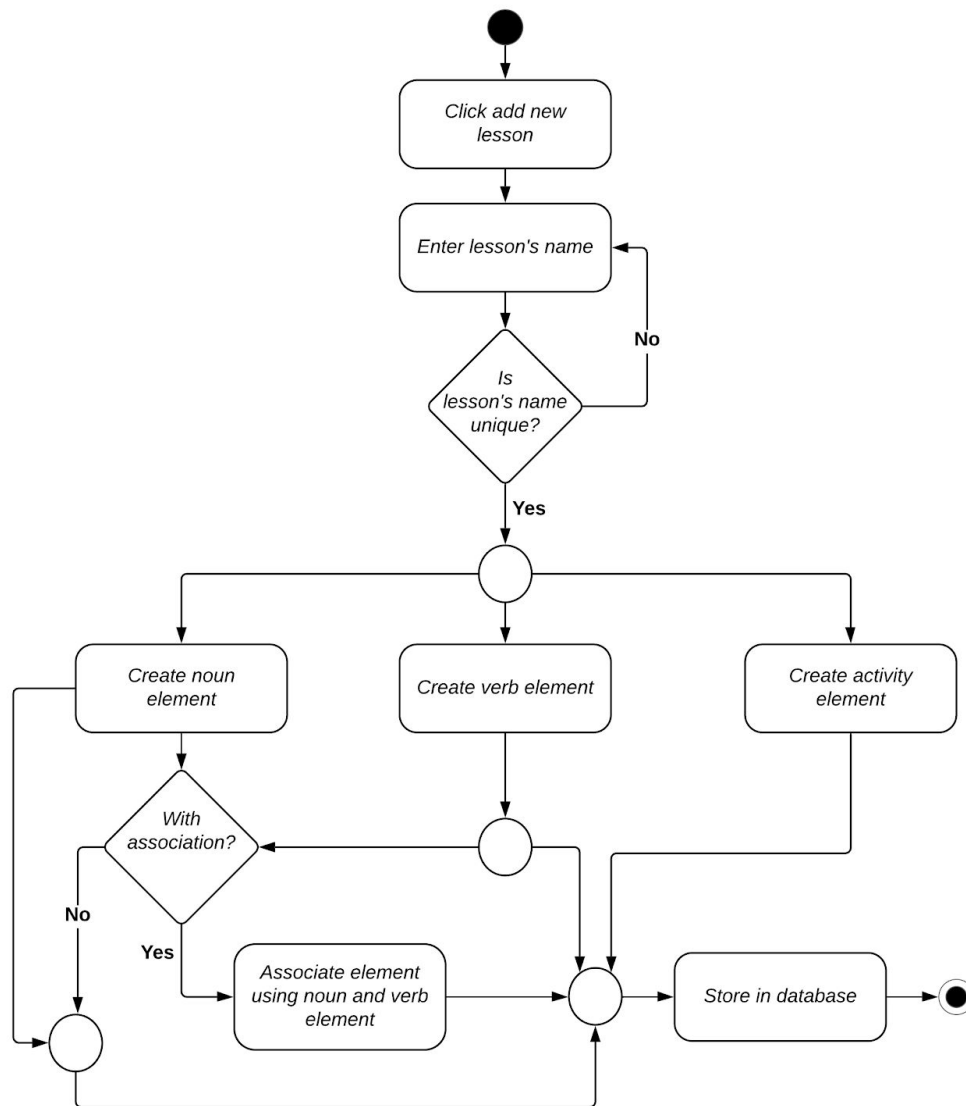


Figure 19: Activity diagram- Add new lesson.

#### 4.4.9 Add lesson elements Activity Diagram

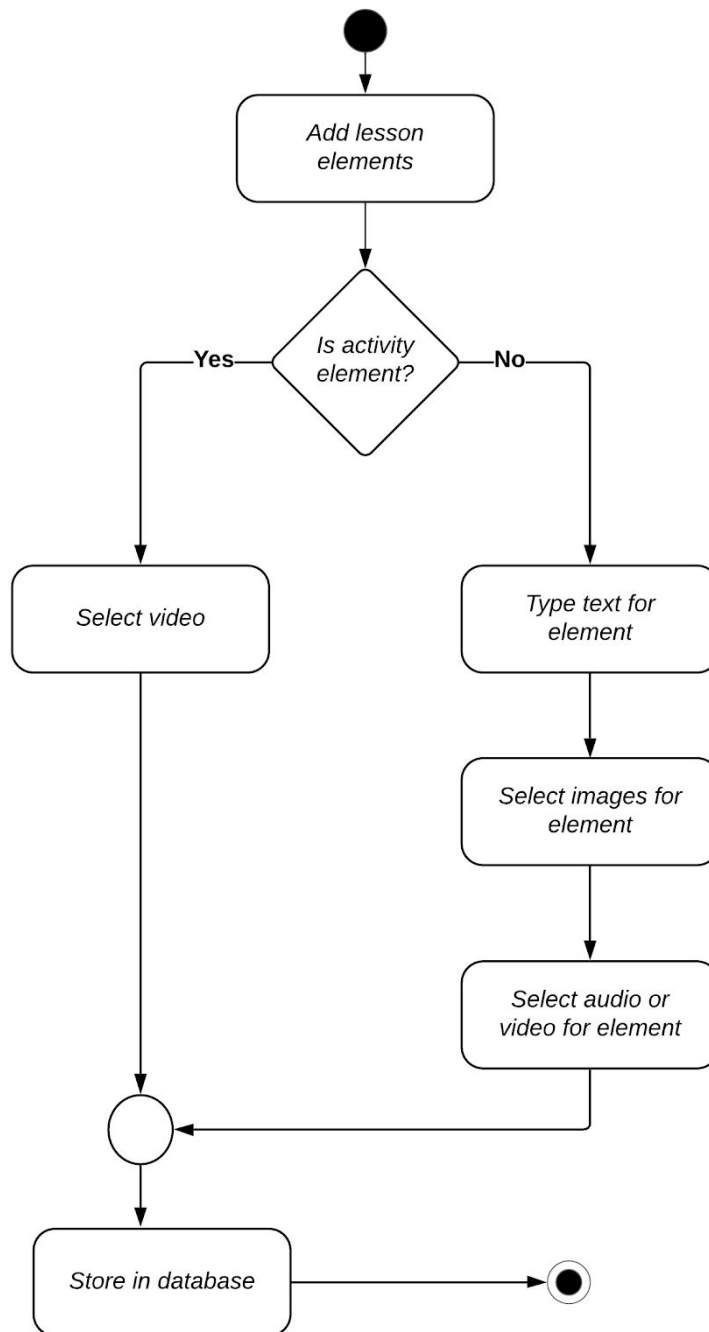


Figure 20: Activity diagram- Add lesson elements.

#### 4.4.10 Show lesson module Activity Diagram

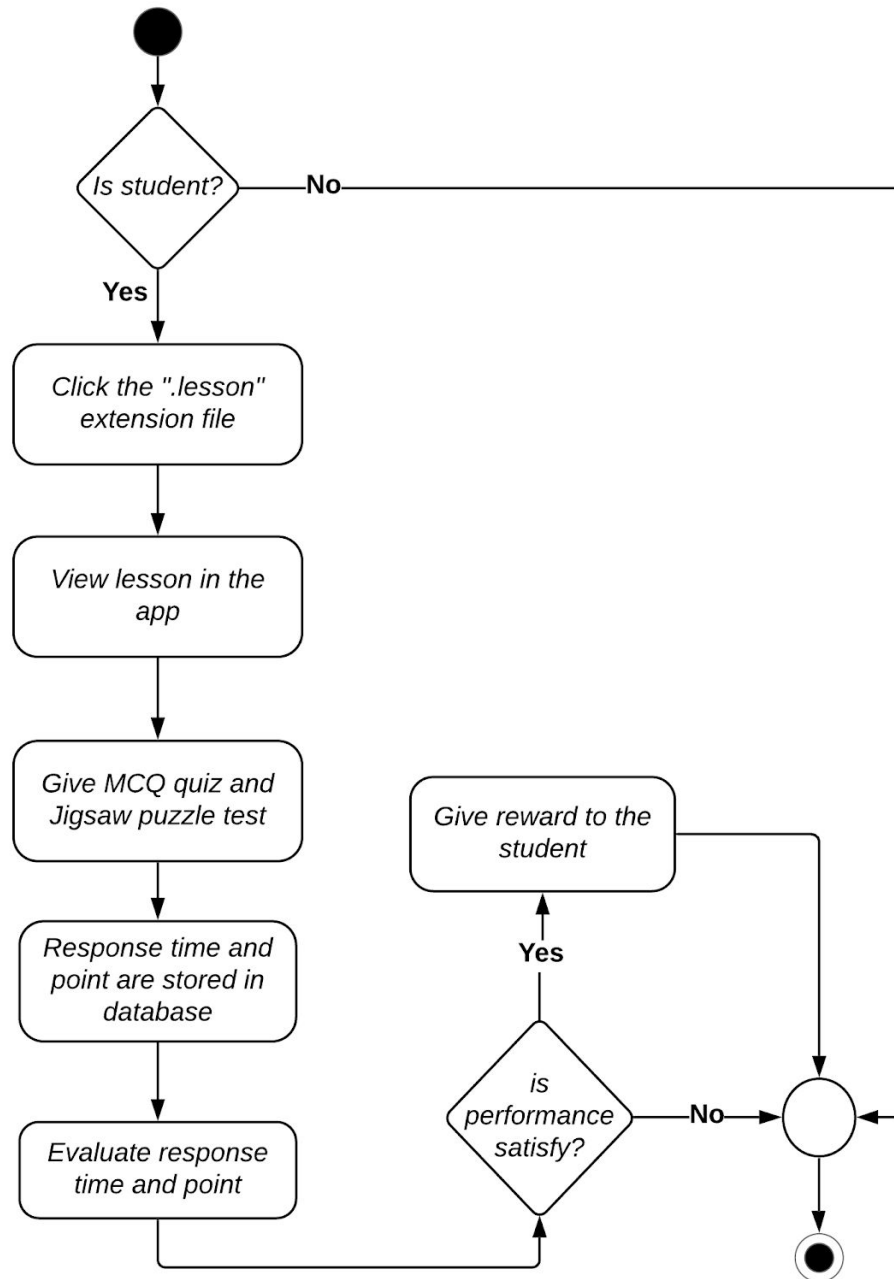


Figure 21: Activity diagram- Show lesson module.

## 4.5 Swim Lane Diagrams of Study Tool for Autistic Children

A swimlane (or swimlane diagram) is used in process flow diagrams, or flowcharts, that visually distinguishes job sharing and responsibilities for sub-processes of a business process.

## 4.5.1 Study Tool for Autistic Children Swimlane

### Diagram

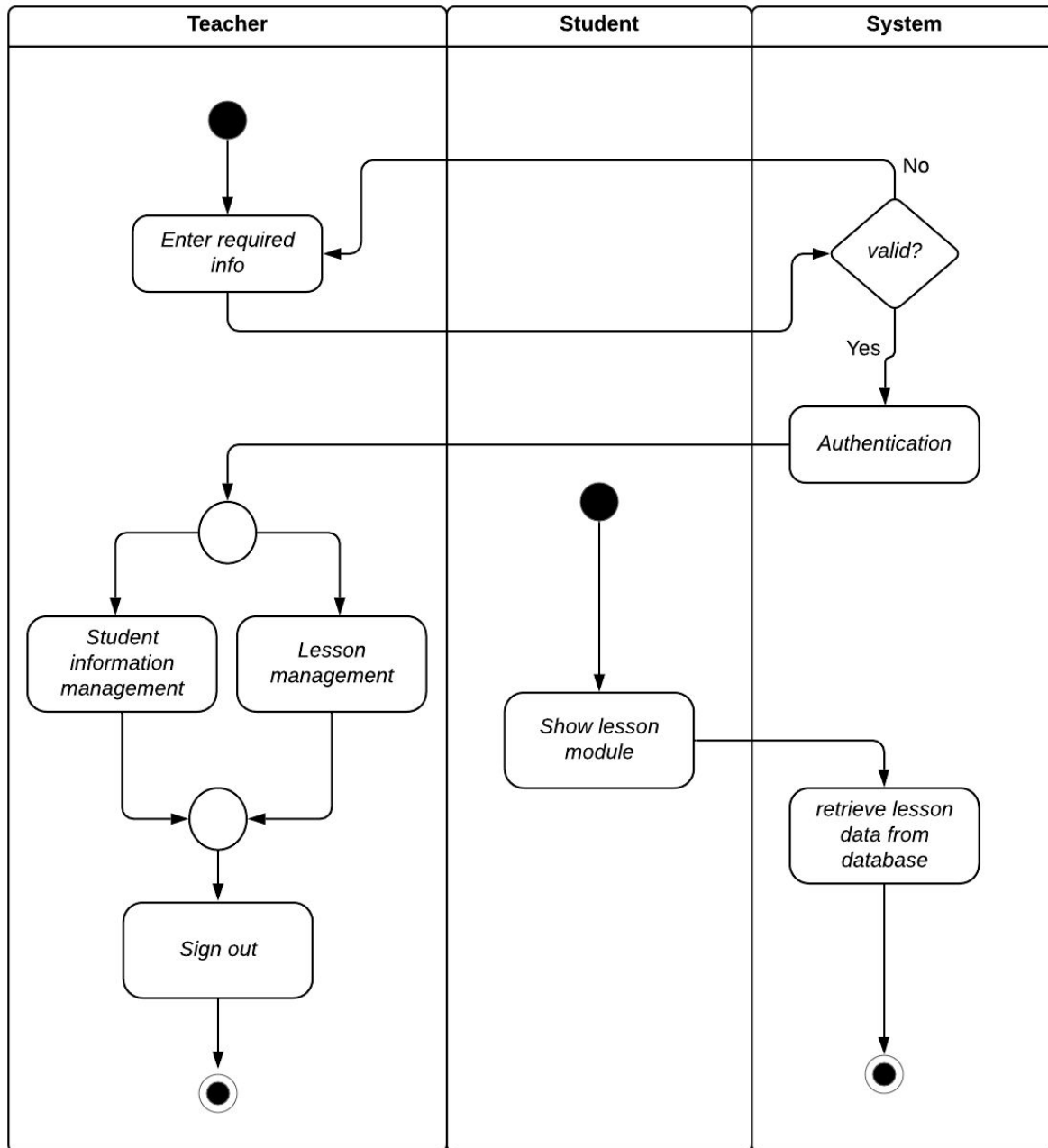


Figure 22: Swimlane diagram- Study Tool for Autistic Children

## 4.5.2 Sign up Swimlane Diagram

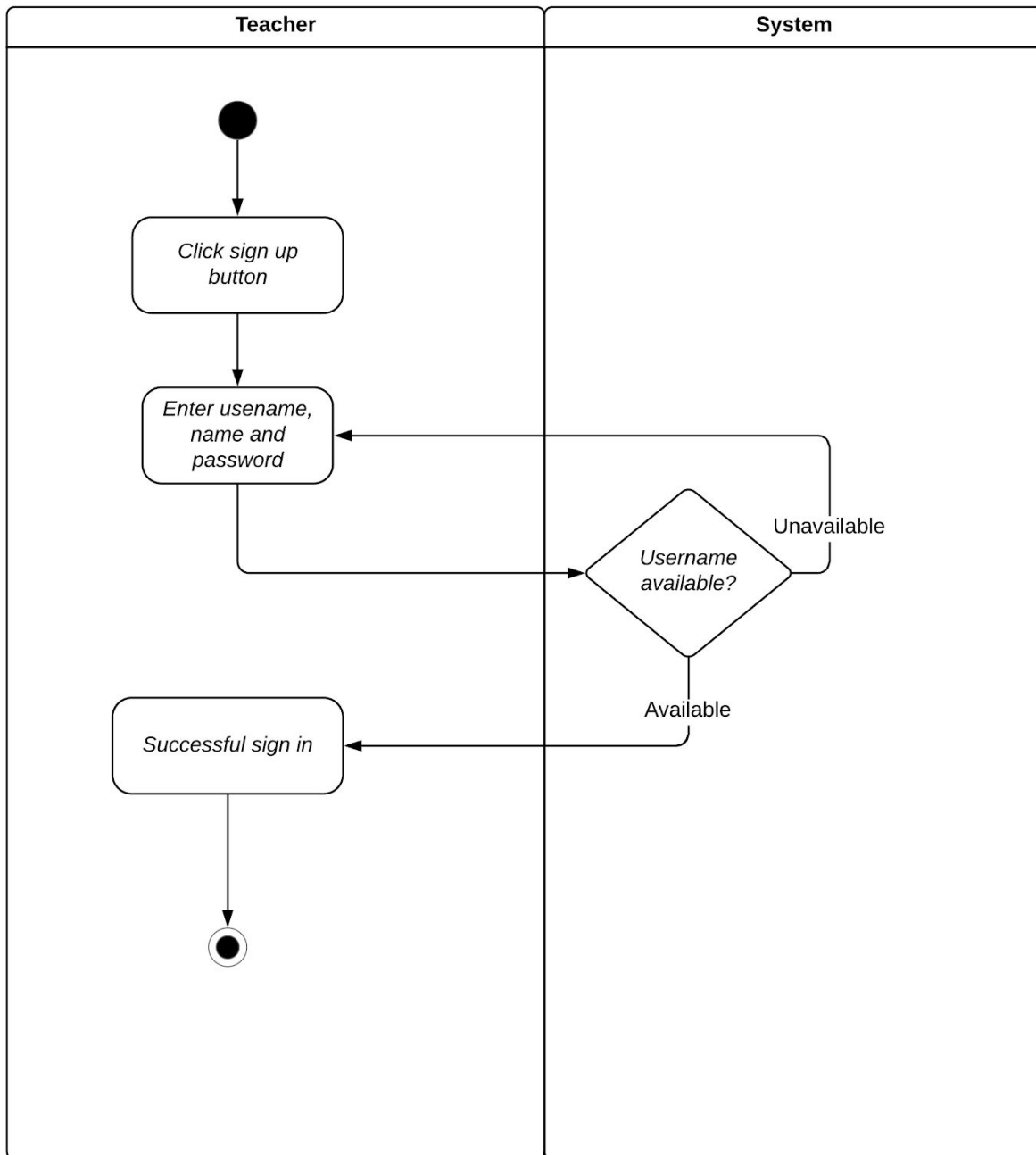


Figure 23: Swimlane diagram- Sign up.

### 4.5.3 Sign in Swimlane Diagram

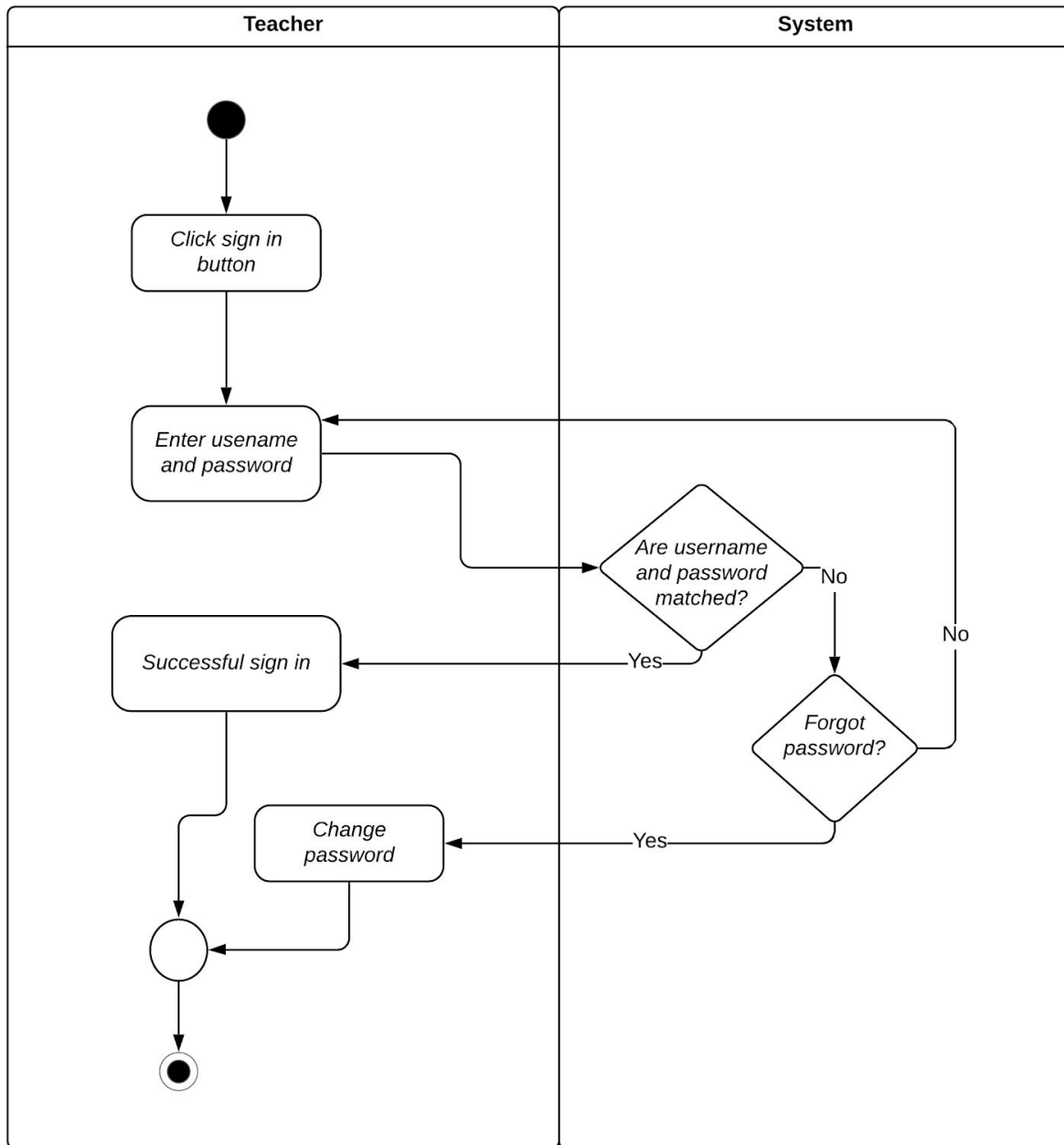


Figure 24: Swimlane diagram- Sign in.



## 4.5.4 Change password Swimlane Diagram

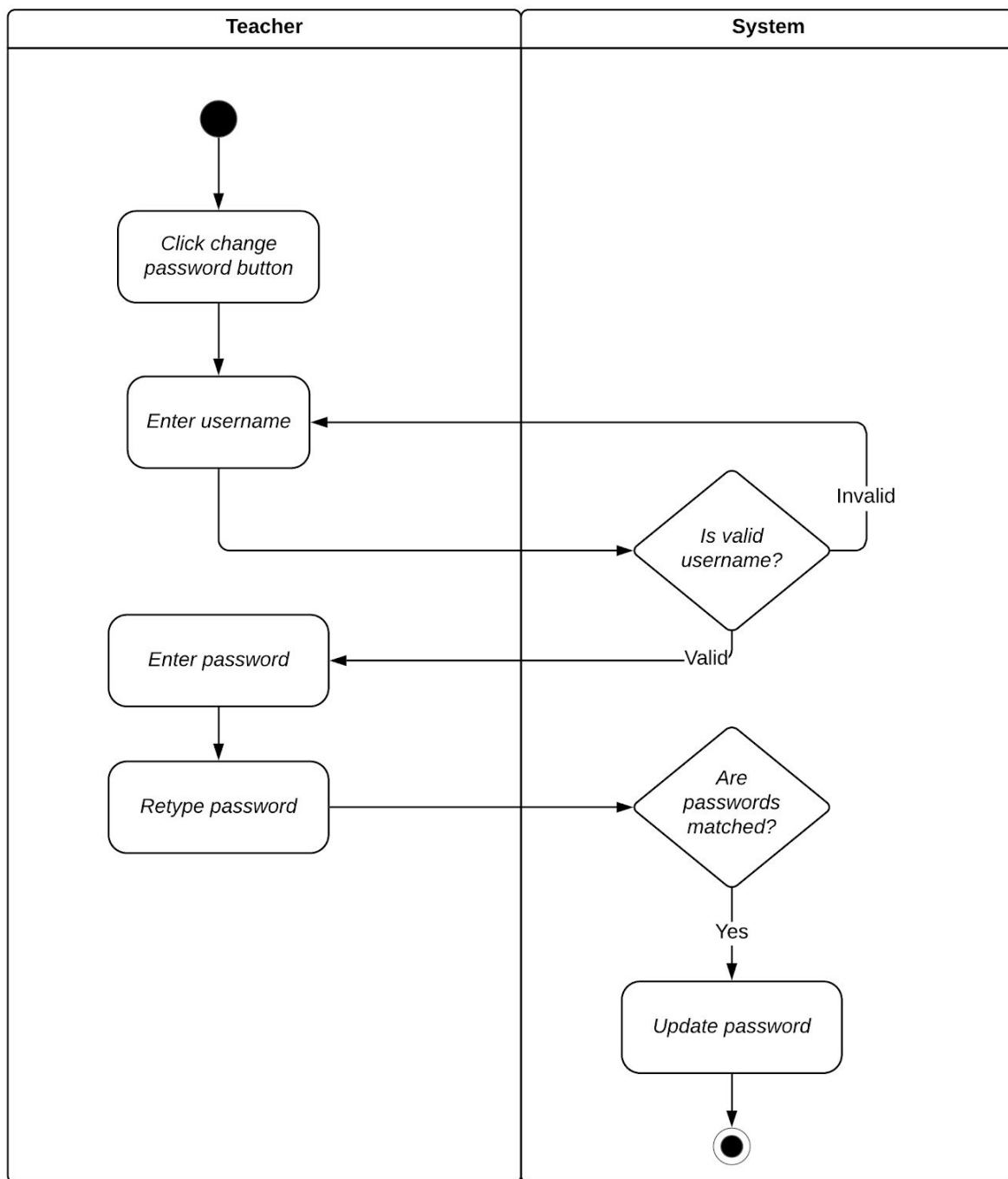


Figure 25: Swimlane diagram- Change password.

### 4.5.5 Student management Swimlane Diagram

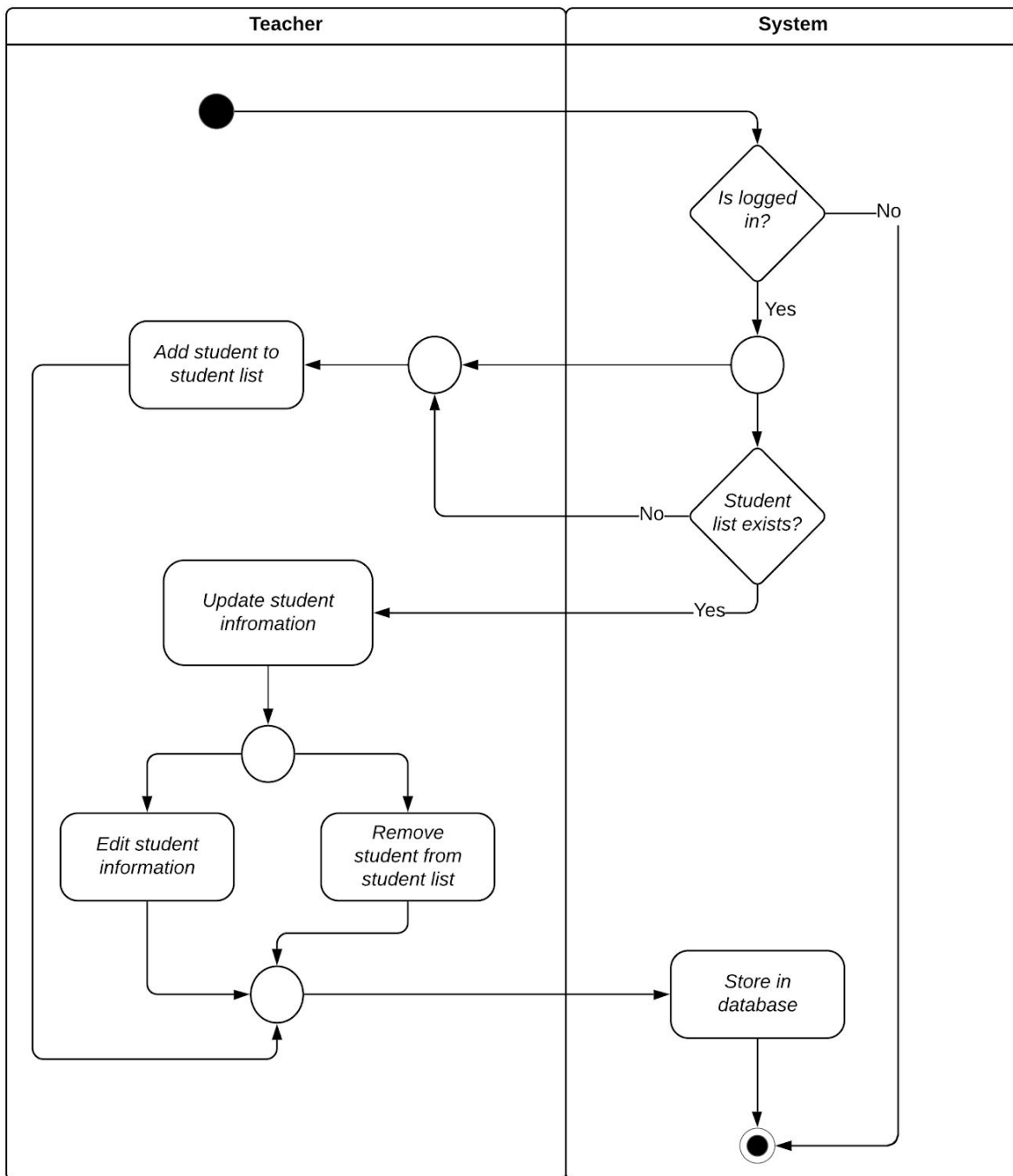


Figure 26: Swimlane diagram- Student management.

### 4.5.6 Add student info Swimlane Diagram

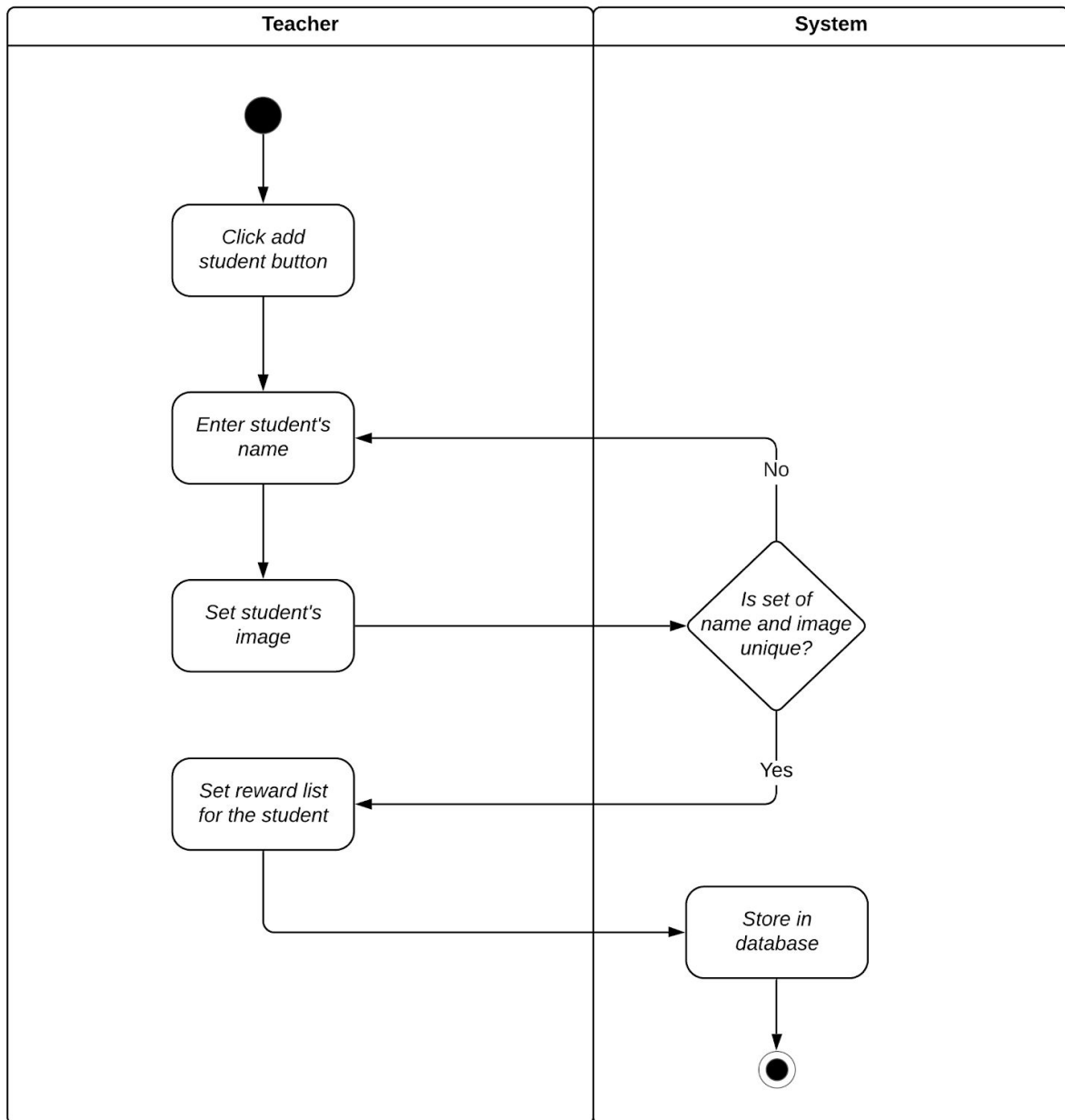


Figure 27: Swimlane diagram- Add student info.

### 4.5.7 Lesson management Swimlane Diagram

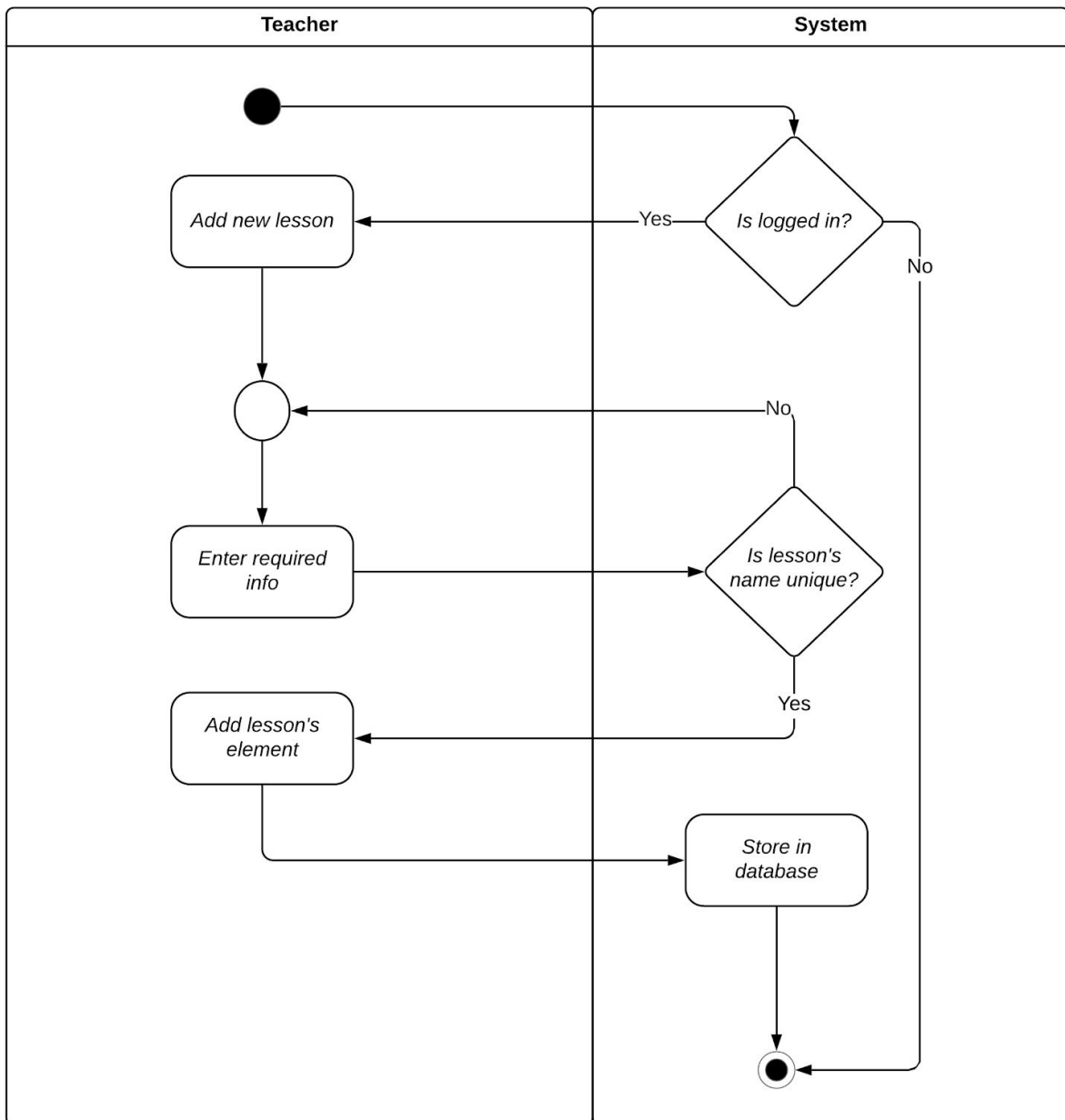


Figure 28: Swimlane diagram- Lesson management.

### 4.5.8 Add new lesson Swimlane Diagram

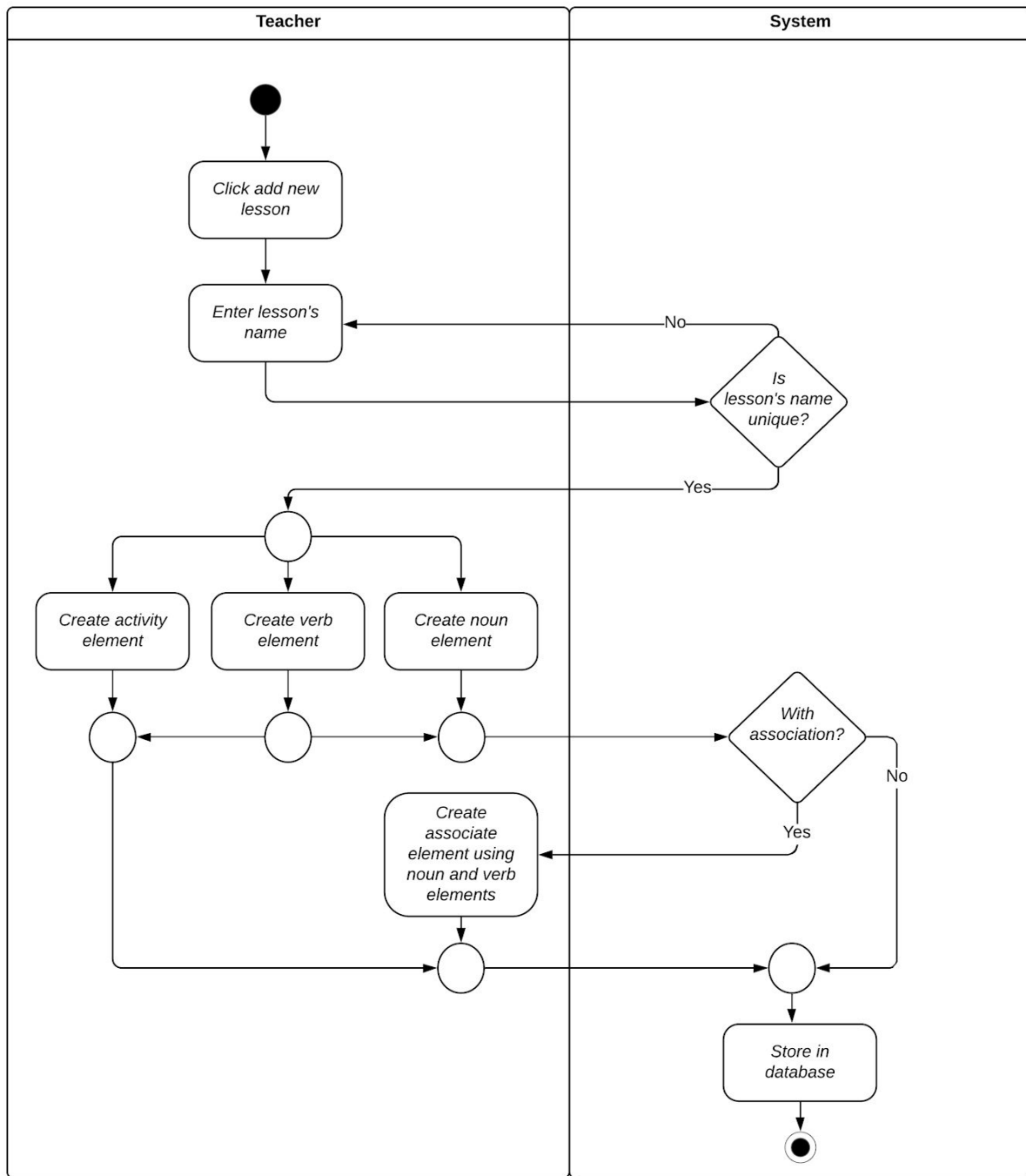


Figure 29: Swimlane diagram- Add new lesson.

#### 4.5.9 Add lesson elements Swimlane Diagram

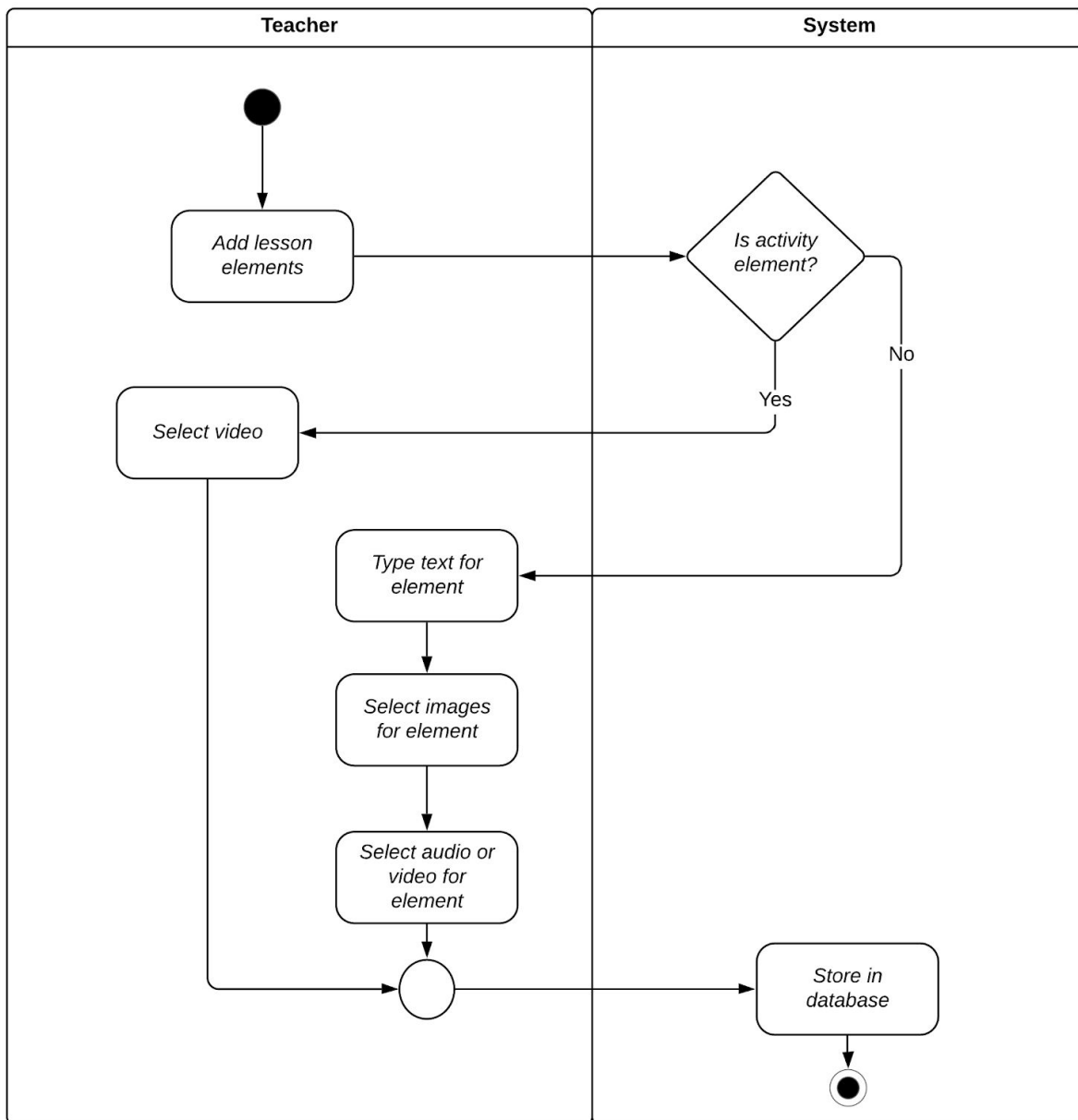


Figure 30: Swimlane diagram- Add lesson elements.

#### 4.5.10 Show lesson module Swimlane Diagram

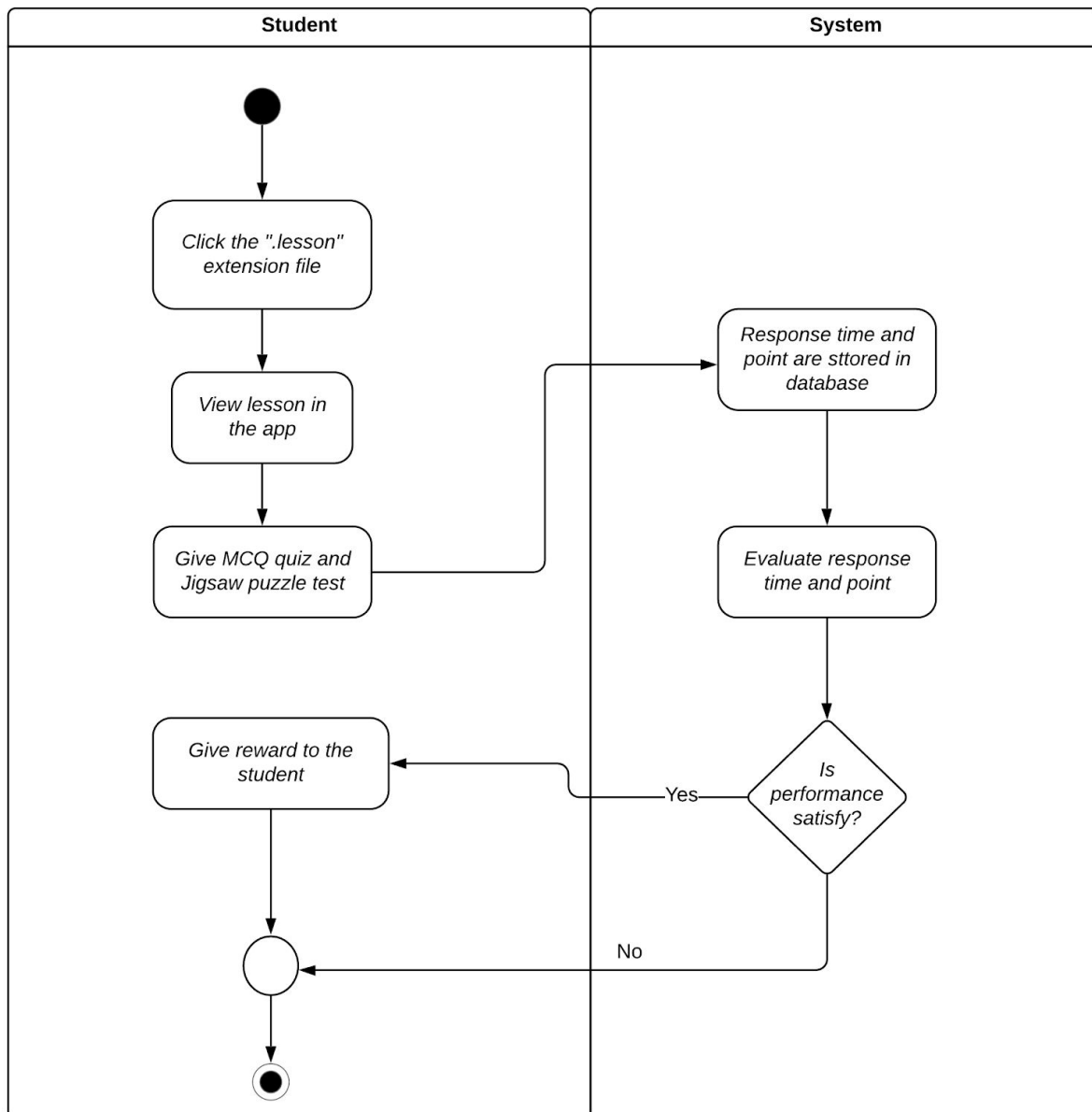


Figure 31: Swimlane diagram- Show lesson module.



# Chapter 5

## DATA MODELLING

### 5.1 Data Modelling Concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) identifies all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

#### 5.1.1 Data Objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

##### 5.1.1.1 Noun Identification

We identified all the nouns whether they are in problem space or in solution space from our story:

Table 1:Noun List

Serial no	Noun	Problem/solution space	Attributes
1	Application	p	
2	Childs	p	
3	Lesson	s	15,16
4	Vocabulary	p	
5	Authentication	p	
6	Teacher	s	8,9,10
7	Information	p	
8	Fullname	s	
9	Username	s	
10	password	s	
11	Student Name	s	
12	Image	s	23,17
13	Reward List	s	11,23
14	Object	p	
15	Lesson Name	s	
16	Thumbnail	s	
17	Element	s	
18	Noun	p	
19	Verb	p	
20	Measurement	p	

21	Cartoon	p	
22	Quiz section	p	
23	File(path)	s	
24	Association	p	
25	Activity	p	
26	Lesson element	s	27,28,29
27	Category	s	
28	Type	s	
29	Word	s	
30	Video	s	23,17
31	Audio	s	23,17
32	Check box	p	
33	Database	p	
34	Show Lesson Module	p	
35	Quiz	p	,
36	Mcq	p	
37	Jigsaw puzzle	p	
38	Extension	p	
39	Student	s	42,11
40	Button	p	
41	Question Window	p	
42	Response	s	
43	Music	p	

44	Animation	p	
45	Decision	p	
46	Computer	p	

### 5.1.1.2 Potential Data Objects

Table 2: potential data objects

No	Data Object	Attributes
1	Teacher	8,9,10
2	Student	11,42
3	Lesson	15,16
4	Lesson Element	27,28,29
5	video	17,23
6	Audio	17,23
7	Image	17, 23
8	Reward	11,23

### 5.1.1.3 Analysis for finalizing Data Objects

- Auto generated id numbers will be selected as the primary key for all data objects except teacher . The object teacher has an attribute named username which will perform as the primary key .

- Video , audio , image file paths will be stored with the lesson element id .
- Reward which is a file path type will be stored with student id .

#### 5.1.1.4 Final Data Objects

In the following table we finalize the data objects with their attributes Most of the attributes of the data objects are selected from the usage scenario and some of the attributes are selected to complete the system which are not in the usage scenario but important for the data objects .

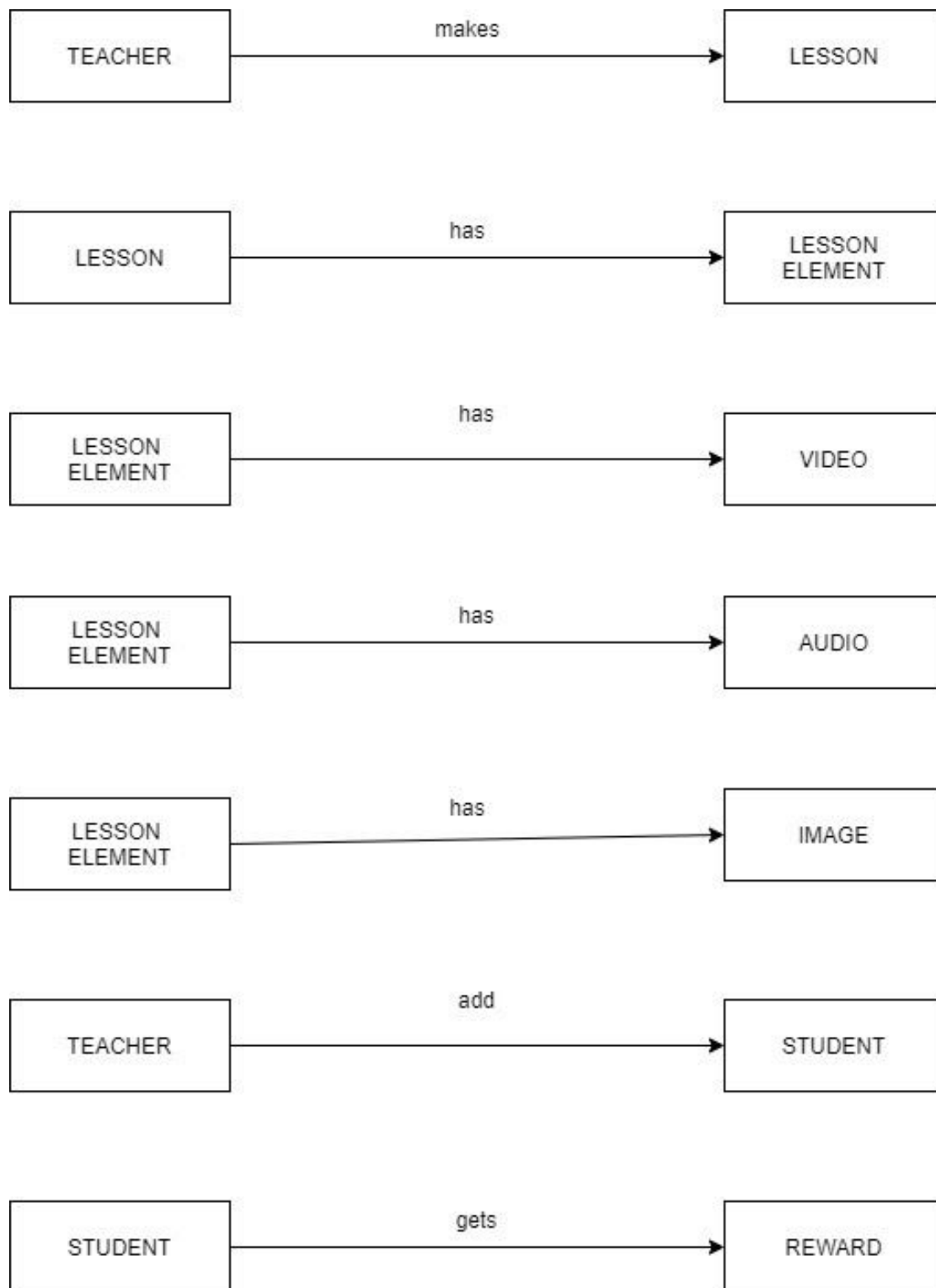
Table 3: data objects

No	Data Object	Attributes
1	Teacher	Full Name , <u>username</u> , password
2	Student	<u>Student id</u> , Student name , image ,Response
3	Lesson	<u>Lesson id</u> , Lesson name ,Thumbnail Path
4	Lesson Element	Lesson id, <u>Element id</u> , Word , type
5	video	<u>Video id</u> , Element id , path
6	Audio	<u>Audio id</u> , Element id , path
7	Image	<u>Image id</u> , Element id , path
8	Reward	<u>Reward id</u> , Reward (File path)

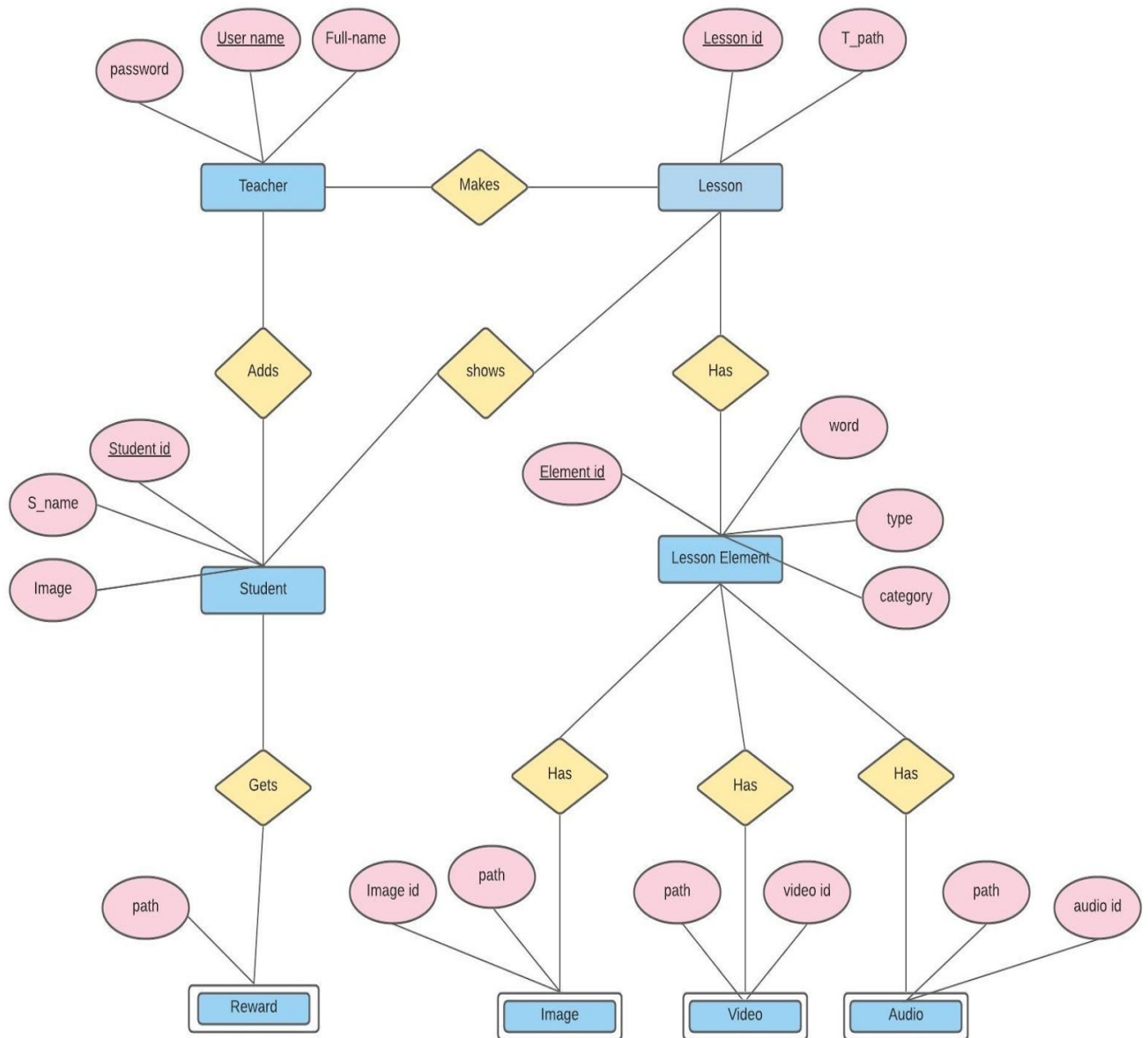
## 5.2 DATA OBJECT RELATIONSHIPS

Data objects are connected to one another in different ways.





## 5.3 ENTITY RELATIONSHIP DIAGRAM





## 5.4 SCHEMA DIAGRAM

Table 04 : Schema for Teacher

Attribute	Type	Size
Full Name	Varchar	40
<u>Username</u>	Varchar	40
Password	Varchar	40

Table 05 : Schema for Student

Attribute	Type	Size
Student_name	Varchar	40
<u>Student_id</u>	Varchar	40
Image_path	Varchar	40
Response(poin in quiz)	Number	10

Table 06: Schema for Lesson

Attribute	Type	Size
<u>Lesson id</u>	Varchar	40
Lesson_name	Varchar	40
Thumbnail Path	Varchar	40

Table 07: Schema for Lesson Element

Attribute	Type	Size
<u>element_id</u>	Varchar	40
Lesson_id	Varchar	40

Word	Varchar	40
Type	Varchar	40

Table 08 : Schema for Video

Attribute	Type	Size
Element_id	Varchar	40
Video_id	Varchar	40
Path	Varchar	40

Table 09: Schema for Audio

Attribute	Type	Size
Element_id	Varchar	40
Audio_id	Varchar	40
Path	Varchar	40

Table 10 : Schema for Image

Attribute	Type	Size
Element_id	Varchar	40
Image_id	Varchar	40
Path	Varchar	40

Table 11 : Schema for Reward

Attribute	Type	Size
-----------	------	------

Student_id	Varchar	40
Reward(File_path)	Varchar	40

# Chapter 6

## CLASS-BASED MODELING

This Chapter is intended to describe class based modeling of “”.

### 6.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

### 6.2 Noun List

Table 12:noun list

NO	Noun	No	Noun
1	Application	24	Association
2	Childs	25	Activity
3	Lesson	26	Lesson element
4	Vocabulary	27	Category
5	Authentication	28	Type
6	Teacher	29	Word
7	Information	30	Video
8	Fullname	31	Audio
9	Username	32	Check box
10	password	33	Database
11	Student Name	34	Show Lesson Module

12	Image	35	Quiz
13	Reward List	36	Mcq
14	Object	37	Jigsaw puzzle
15	Lesson Name	38	Extension
16	Thumbnail	39	Student
17	Element	40	Button
18	Noun	41	Question Window
19	Verb	42	Response time
20	Measurement	43	Music
21	Cartoon	44	Animation
22	Quiz section	45	Decision
23	Database File	46	Computer

## 6.3 General Classification

To identify the potential classes, we have first selected the nouns from the solution space of the story. These were then characterized in seven general classification. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units

## 6. Places

## 7. Structures

Following are the specifications of the nouns according to the general Classifications:

Table 13: gc

No	Noun	General Classification
1	Application	3,6
2	Child	4
3	Lesson	2,3
4	Vocabulary	
5	Authentication	
6	Teacher	4
7	Information	
8	Fullname	
9	Username	
10	password	
11	Student Name	
12	Image	
13	Reward List	
14	Object	
15	Lesson Name	

16	Thumbnail	
17	Element	
18	Noun	
19	Verb	
20	Measurement	
21	Cartoon	
22	Quiz section	
23	Database File	
24	Association	
25	Activity	
26	Lesson element	2,3
27	Category	
28	Type	
29	Word	
30	Video	
31	Audio	
32	Check box	
33	Database	1,3,6
34	Show Lesson Module	2,3,6
35	Quiz	
36	Mcq	
37	Jigsaw puzzle	
38	Extension	
39	Student	4
40	Button	
41	Question Window	
42	Response time	

43	Music	
44	Animation	
45	Decision	
46	Computer	

## 6.4 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills most of the characteristics for these six.

1.Retain information

2.Needed services

3.Multiple attributes

4.Common attributes

5.Common operations

6.Essential requirements

Table 14:sc

No	Noun	Selection Criteria
1	Application	3
2	Child	3
3	Lesson	1,6



4	Teacher	1,6
5	Lesson element	1,2,6
6	Database	1,2,6
7	Show Lesson Module	2,6
8	Student	1,6
9		
10		

## 6.4 Attribute Selection

After identifying the classes, we have specified their attributes and methods.

Table 15: attribute selection

No	Class Name	Attribute
1	Teacher	Full Name Username Password StudentList
2	Student	Name Image Reward
3	Lesson	Lesson Name Thambnail
4	Lesson Element	Category Type Word Video Image Audio
5	Show Lesson Module	Category Type Word

		Video Image Audio
6	Database	Database File(sqlite file path)

## 6.5 Method Identification

After identifying the classes, we have specified their methods.

Table 16: method identification

No	Class Name	Method
1	Teacher	<ul style="list-style-type: none"> <li>• Sign in()</li> <li>• Sign up()</li> <li>• storeTeacherInfo()</li> <li>• LessonManagement()</li> <li>• StudentListManagement()</li> <li>• addNewStudentToStudentList()</li> <li>• deleteStudentFromStudentList()</li> </ul>
2	Student	<ul style="list-style-type: none"> <li>• showLesson()</li> </ul>
3	Lesson	<ul style="list-style-type: none"> <li>• addNewLesson()</li> <li>• storeLessonInfo()</li> <li>• deleteLesson()</li> <li>• addLessonElement() <ul style="list-style-type: none"> <li>○</li> </ul> </li> </ul>
4	Lesson Element	<ul style="list-style-type: none"> <li>• AddNewLessonElement()</li> <li>• storeLessonElement()</li> <li>• deleteLessonElement()</li> </ul>
5	Show Lesson Module	<ul style="list-style-type: none"> <li>• retrievingLessonContent()</li> <li>• slideShowingWithLessonContent()</li> <li>• autoGeneratingMcqWithImage()</li> <li>• autoGeneratingJigsawPuzzleWithImage()</li> <li>• storeResponseOfQuiz()</li> </ul>
6	Database	<ul style="list-style-type: none"> <li>• saveNewLesson()</li> <li>• saveNewLessonElement()</li> </ul>

		<ul style="list-style-type: none"> <li>• retrieveLesson()</li> <li>• retrieveLessonElement()</li> <li>• deleteLesson()</li> <li>• deleteLessonElement()</li> <li>• saveNewStudentInfo()</li> <li>• deleteStudentInfo()</li> <li>• saveNewTeacherInfo()</li> <li>• deleteTeacherInfo()</li> <li>• checkInfo()</li> </ul>
--	--	---

## 6.6 Analyzing Classes

It is required to store every data to the database . So , every class must connect to the database class which is the controller of the main database .For this a common method database connection() is added to all other classes .

## 6.7 Class Card

After identifying our final classes we have generated the following class cards.

Table 17: class card for teacher

Teacher	
Attribute	Method
<ul style="list-style-type: none"> <li>• Username</li> <li>• Fullname</li> <li>• Password</li> <li>• StudentList</li> </ul>	<ul style="list-style-type: none"> <li>• Sign in()</li> <li>• Sign up()</li> <li>• databaseConnection()</li> <li>• LessonManagement()</li> <li>• StudentListManagement()</li> <li>• addNewStudentToStudentList()</li> <li>• deleteStudentFromStudentList()</li> </ul>

Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Creating new teacher account</li> <li>• Creating teacher's student List</li> <li>• Store all info to database</li> </ul>	<ul style="list-style-type: none"> <li>• addNewLesson()</li> <li>• StudentListManagement()</li> </ul>

Table 18: class card for lesson

Lesson	
Attribute	Method
Lesson name Thumbnail	<ul style="list-style-type: none"> <li>• addNewLesson()</li> <li>• addLessonElement()</li> <li>• databaseConnection()</li> <li>• deleteLessonElement()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Making new lesson</li> <li>• Adding new lesson content to lesson</li> </ul>	<ul style="list-style-type: none"> <li>• LessonManagement()</li> <li>• AddNewLessonElement()</li> </ul>

Table 19: class card for lesson element

Lesson Element	
Attribute	Method
category Word Type Image Video Audio	<ul style="list-style-type: none"> <li>• AddNewLessonElement()</li> <li>• deleteLessonElement()</li> <li>• databaseConnection()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>• Adding new element to lesson</li> <li>• Adding text to lesson</li> <li>• Adding video to lesson</li> <li>• Add audio to lesson</li> <li>• Add image to lesson</li> <li>• Deleting a lesson element</li> </ul>	<ul style="list-style-type: none"> <li>• addLessonElement()</li> <li>• deleteLessonElement()</li> </ul>

Table 20: class card for student

Student	
Attribute	Method
Student name Image reward(File path)	<ul style="list-style-type: none"> <li>• showLesson()</li> <li>• databaseConnection()</li> <li>• studentListManagement()</li> </ul>
Responsibilities	Collaboration

<ul style="list-style-type: none"> <li>Students can show lesson</li> </ul>	<ul style="list-style-type: none"> <li>StudentListManagement()</li> <li>slideShowingWithLessonContent()</li> </ul>
--	--

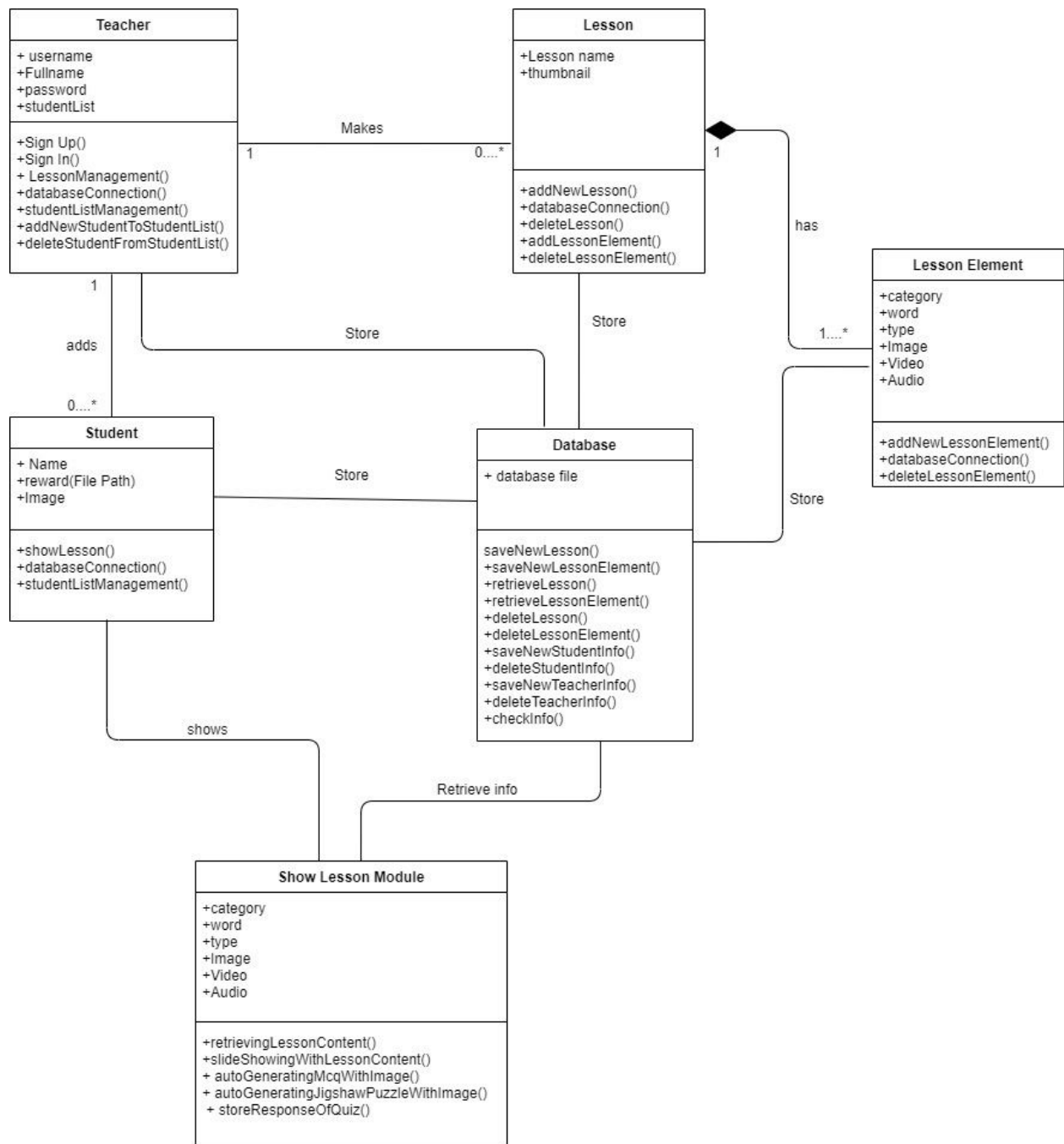
Table 21: class card for database

Database	
Attribute	Method
Database file	<ul style="list-style-type: none"> <li>saveNewLesson()</li> <li>saveNewLessonElement()</li> <li>retrieveLesson()</li> <li>retrieveLessonElement()</li> <li>deleteLesson()</li> <li>deleteLessonElement()</li> <li>saveNewStudentInfo()</li> <li>deleteStudentInfo()</li> <li>saveNewTeacherInfo()</li> <li>deleteTeacherInfo()</li> <li>checkInfo()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>Store all information in database</li> <li>Retrieve information from database</li> <li>Delete information</li> <li>Check any information</li> </ul>	<ul style="list-style-type: none"> <li>databaseConnection()(from every class except show lesson module )</li> <li>retrievingLessonContent()</li> </ul>

Table 22: class card for Show Lesson Module

Show Lesson Module	
Attribute	Method
category Word Type Image Video Audio	<ul style="list-style-type: none"> <li>retrievingLessonContent()</li> <li>slideShowingWithLessonContent()</li> <li>autoGeneratingMcqWithImage()</li> <li>autoGeneratingJigsawPuzzleWithImage()</li> </ul>
Responsibilities	Collaboration
<ul style="list-style-type: none"> <li>Showing lesson elements</li> <li>Appearing jigsaw puzzle window</li> <li>Appearing mcq quiz window</li> </ul>	<ul style="list-style-type: none"> <li>showLesson()</li> </ul>

## 6.8 CRC Diagram





## Chapter 7

### Behavioral Modeling

#### 7.1 State Transition Diagram

State diagram represents active states for each class the events (triggers). For this we identified all the events, their initiators and collaborators.

##### 7.1.1 Event Identification

Table23 : Event Identification

Serial no	Events	Primary Object	Collaborator	Invoke Method
1.	Make lesson with contents	Teacher	Lesson, Database	lessonManagement(), addNewLesson() , saveNewLesson() )
2.	Teach vocabulary	Teacher	Lesson Element	lessonManagement(), addNewLessonElement()
3.	Provide required information for sign up	Teacher	Database	signUp(), checkInfo()

4.	Save sign up info		Database	saveNewTeacherInfo()
5.	Provide required information for sign in	Teacher	Database	signIn(), checkInfo()
6.	Create student list providing student information	Teacher	Database	StudentListManagement(), saveNewStudentInfo()
7.	Update student information in the student list	Teacher	Database	addNewStudentToStudentList(),  deleteStudentFromStudentList(), saveNewStudentInfo()
8.	Create new lesson	Teacher	Lesson, Database	lessonManagement(), addNewLesson(), , saveNewLesson(), , storeLessonInfo()
9.	Edit lesson info	Teacher	Lesson, Lesson Element, Database	lessonManagement(), addLessonElement(), addNewLessonElement(), deleteLesson(), storeLessonInfo()
10.	Update lesson elements	Teacher	Lesson Element,	lessonManagement(),

			Database	addNewLessonElement(), storeLessonElement(), deleteLessonElement(), saveNewLessonElement()
11.	Lesson elements stored in database		Lesson Element, Database	storeLessonElement(), saveNewLessonElement()
12.	Select lesson contents	Teacher	Lesson, Lesson Element, Database	lessonManagement(), addLessonElement(), addNewLessonElement(), storeLessonElement(), saveNewLessonElement()
13.	View lesson contents	Student	Show Lesson Module, Database	showLesson(), retrievingLessonContent(), slideShowingWithLessonContent(), retrieveLesson(), retrieveLessonElement()
14.	Quizzes appear	Student	Show Lesson Module, Database	showLesson(), autoGeneratingMcqWithImage(), autoGeneratingJigsawPuzzleWith

				hImage(), retrieveLesson()
15.	Complete the lesson	Student	Show Lesson Module	showLesson(), slideShowingWithLessonContent() )
16.	Point stored against the student id		Show Lesson Module	storeResponseOfQuiz()
17.	Response time stored		Show Lesson Module	storeResponseOfQuiz()
18.	Point and response time measured	Teacher	Database	StudentListManagement(), checkInfo()
19.	MCQ quiz will be made	Database	Show Lesson Module	autoGeneratingMcqWithImage()
20.	Jigsaw puzzle will be generated	Database	Show Lesson Module	autoGeneratingJigsawPuzzleWithImage()
21.	Measure students outcome	Teacher	Database	StudentListManagement(), checkInfo()
22.	Update teacher account info	Teacher	Database	storeTeacherInfo(), saveNewTeacherInfo()

## 7.1.2 Events After Analysis

Table 24: Merged Events

Serial no	Events	Primary Object	Collaborator	Invoke Method
1.	Provide required information for sign up	Teacher	Database	signUp(), checkInfo()
2.	Save sign up info	Database		saveNewTeacherInfo()
3.	Provide required information for sign in	Teacher	Database	signIn(), checkInfo()
4.	Create student list	Teacher	Database	StudentListManagement(), saveNewStudentInfo()
5.	Update student information	Teacher	Database	addNewStudentToStudentList(), deleteStudentFromStudentList(), saveNewStudentInfo()

6.	Create new lesson	Teacher, Lesson	Database	lessonManagement(), addNewLesson() , saveNewLesson() , storeLessonInfo() )
7.	Edit lesson info	Teacher, Lesson, Lesson Element	Database	lessonManagement(), addLessonElement(), addNewLessonElement(), deleteLesson(), storeLessonInfo() )
8.	Update lesson elements	Teacher, Lesson Element	Database	lessonManagement(), addNewLessonElement(), storeLessonElement(), deleteLessonElement(), saveNewLessonElement(), storeLessonElement() )
9.	View lesson contents	Student, Show Lesson Module	Database	showLesson(), retrievingLessonContent(), slideShowingWithLessonContent() , retrieveLesson(), retrieveLessonEl

				ement()
10.	MCQ quiz, Jigsaw puzzle will be appeared	Student, Show Lesson Module	Database	showLesson(), autoGenerating McqWithImage(), autoGeneratingJi gshawPuzzleWit hImage(), retrieveLesson()
11.	Point stored and response time against the student id	Show Lesson Module	Database	storeResponseO fQuiz()
12.	Remove information	Database		deleteLesson(), deleteLessonEle ment(), deleteStudentInf o(), deleteTeacherInf o()
13.	Evaluate point and response time	Teacher	Database	StudentListMana gement(), checkInfo()
14.	Update information	Database		saveNewLesson( ) saveNewLesson Element(), saveNewStudent Info(), saveNewTeache rInfo()

## 7.1.3 State Transition Diagrams

### Teacher

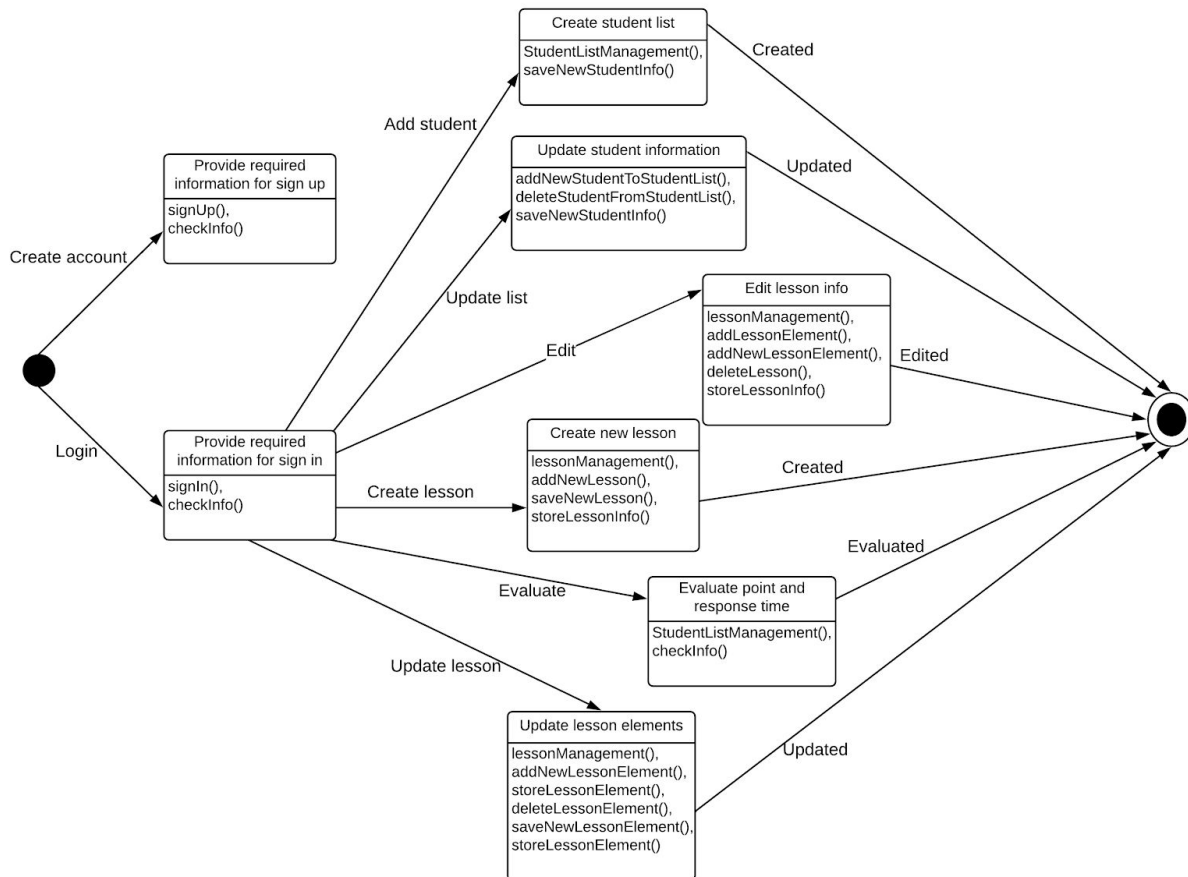


Figure : Teacher State Diagram.



## Student

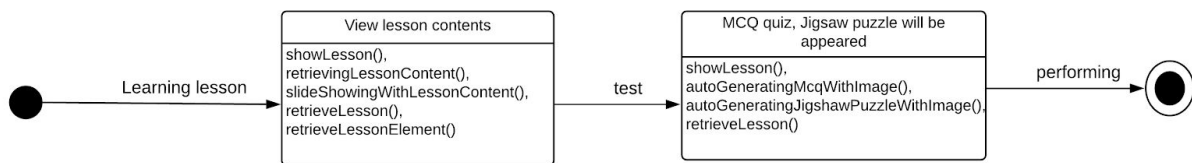


Figure : Student State Diagram.

## Lesson

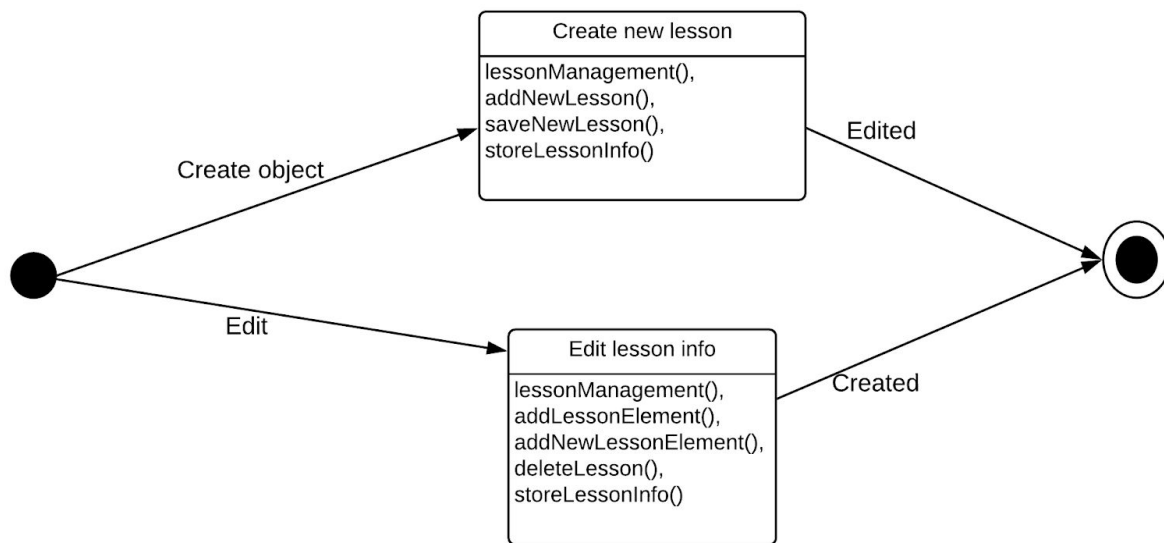


Figure : Lesson State Diagram.

## Lesson Element

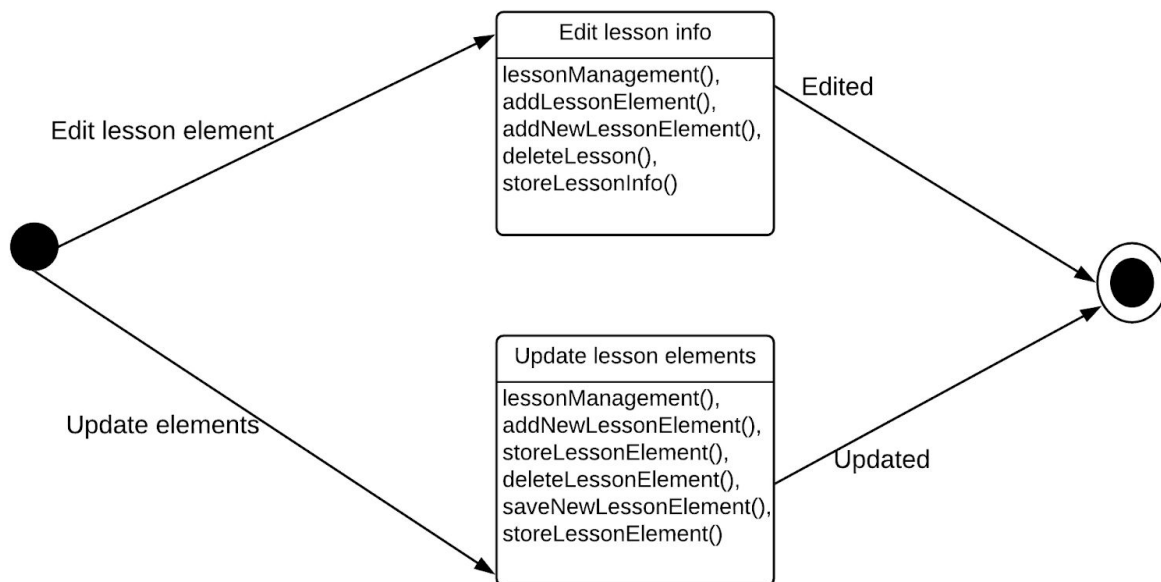


Figure : Lesson Element State Diagram.

## Show Lesson Module

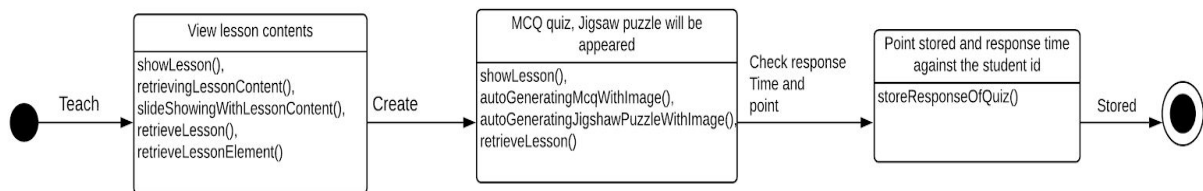


Figure : Show Lesson Module State Module.

## Database

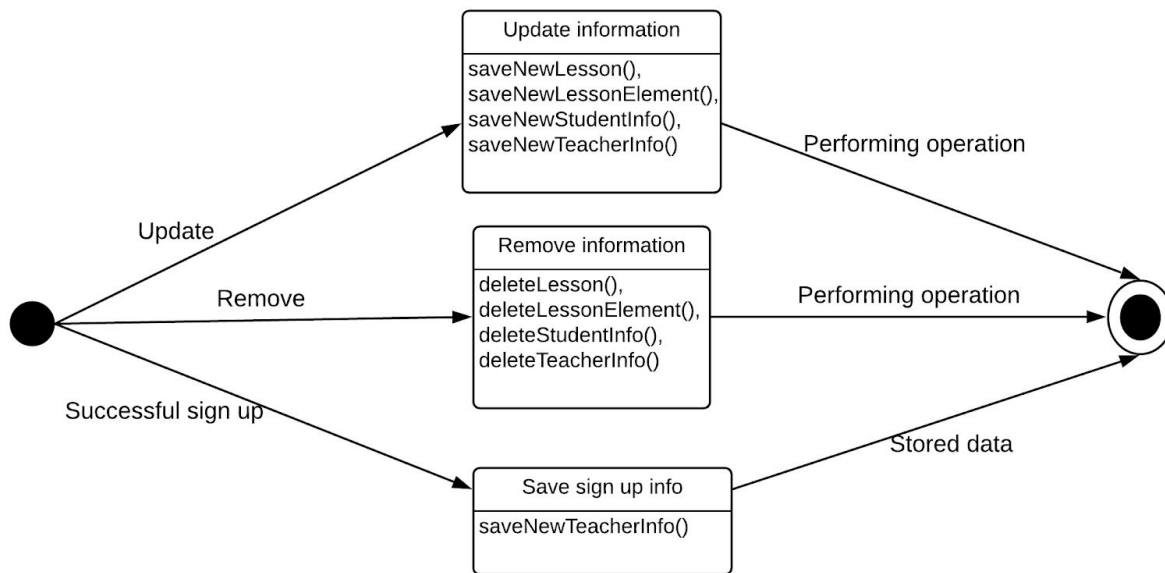
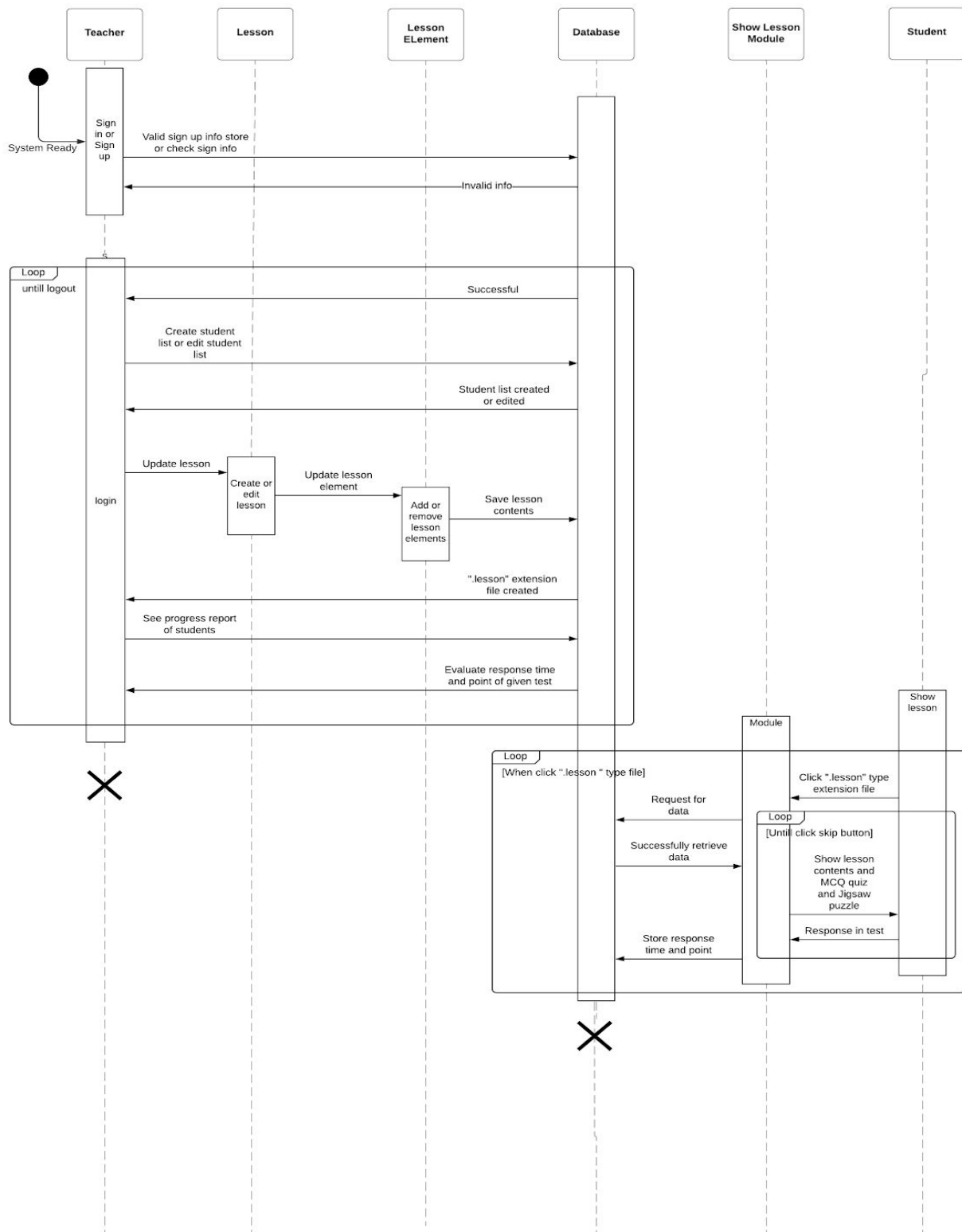


Figure : Database State Diagram.

## 7.2 Sequence Diagram



----- THE END-----