PLAGIARISM DETECTION SYSTEM

Documentation

MAY 3, 2016

INSTITUTE OF INFORMATION TECHNOLOGY
University of Dhaka

Submitted to

Asif Imran
Lecturer
Institute of Information Technology
University of Dhaka

Submitted by

Md. Feroz Ahmmed

BSSE 0618

Supervisor

Md. Saeed Siddik

Lecturer

Institute of Information Technology

University of Dhaka

Submission date

May 3, 2016

Letter of Transmittal

May 3, 2016
Asif Imran
Lecturer
Institute of Information Technology
University of Dhaka

Dear Sir,
I have prepared the report on Software Requirements Specification of **'Plagiarism Detection 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

Figures

Tables

Acronyms

Acronym	Definition
SRS	Software Requirement Specification
PDS	Plagiarism Detection 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 Plagiarism Detection System (PDS). 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 two key themes in terms of the Plagiarism Detection System - A submission system of authenticated users and plagiarism detection of submitted documents. This system is only for academic purpose. Any academic activity which is not related to document submission and viewing documents is considered as out of scope, and will not be covered in this document.

1.4 Rational

Manual document judging have some problems. Some of these problems are-

- Students try to submit same documents changing some contents or may be just changing the title which is an illegal activity.
- It is very hard for teachers to check similarity among documents and sometimes it causes misjudgment.
- We also notice that students try to take solution of assignments from seniors. Teachers can't fix it because of course teacher replacement.

This online submission and plagiarism detection system will make the task easy for teachers to stop these illegal activities.

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 I consulted with my Honorable teacher Md. Saeed Siddik 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 Plagiarism Detection System**:**

1. Student The largest user group of the system. The whole project become useless without them. They are the main beneficiaries of this project. They will fill the form, pay submit documents as well as they will be judged through their submissions.
2. Teacher: Faculty members are an essential part of this project. They have direct influence over Plagiarism Detection System. They view the documents submitted by students and check the reports of the submissions.

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

2.1.2 Asking the First Question

I 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 me to identify all stakeholders, measurable benefit of the successful implementation and possible alternatives to custom software development. Next set of question helped me 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

I collected these viewpoints by discussing with some students, teachers and officials from different departments and institutes.

1. University's viewpoint:
    - No disruption of rules and regulation
2. Student's viewpoint:
    - User friendly
    - Web-Based Interfaces
    - Option to see other's activity
3. Teacher's viewpoint:
    - Web-Based Interfaces
    - Maintain a database of all items in this system
    - Strong Authentication
    - User friendly
    - 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. I 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
- The application can be accessed from any computer that has Internet access
- Maintain a database of all information in this system

Conflicting Requirements:

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

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

Final Requirements:

I 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 me to establish basic understanding about Plagiarism Detection System, identify the people who will be ben**e**fited if Plagiarism Detection System establishes, define the nature of the Plagiarism Detection 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 I 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, I 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 I 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. I completed following steps to do it.

- The meetings were conducted with my honorable teacher. He was questioned about his requirements and expectations from the Plagiarism Detection System.
- Teachers were asked about the problems they are facing with the current manual system.
- At last I selected the final requirement list from the meetings with my 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 Plagiarism Detection 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.

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. The user interface should provide appropriate error messages for invalid input.
2. 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.

3.6 Usage Scenarios

User authentication of Plagiarism Detection System can be separated into three parts named as sign up, sign in and sign out. Sign up form contains student name, faculty name, department name, registration number, class, email address, session, password and class roll. When a student fills up the sign up form, a verification request is sent to admin. Verification request contains all the information about the student except the password. If admin verifies the student, the account becomes active. Admin has the right to create or remove any student. Students can sign in through email and password. Students can change their password, but other information are not changeable. There is some super users who are basically teachers. Teachers are added by admin with their information. Teachers' information contains name, faculty name, department name, email address and designation. Admin provides the initial password to every teacher. Teachers can reset their password at any time. Teachers can be removed by admin. Users can exit from the system through sign out.

Signed in students can submit any kind of document at any time. To submit a document, student has to mention submission type, title and course code. After submitting the document, student will get a report link about validity of the submitted document. If the document contains negligible amount of plagiarism the report contains the message that submission is successful. On the other hand, if the document contains significant amount of plagiarism, the report contains the message that submitted document contains plagiarism. The report also contains how similar previous files were with the submitted document.

Teachers can view any kind of submission. Teachers can also download any document. Each submission in the submission history is associated with the link of report of that submission. Admin and Teachers can view reports through these links.

3.7 Elicitation Work Product

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

My elicitation work product includes:

- Make a statement of my requirements for Plagiarism Detection System.
- Make a bounded statement of scope for my 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
Plagiarism Detection 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
	Submission		Student
	View Submission	View Document	Teacher
		Download	Teacher
	View Report		Teacher

4.3 Use Case Description

I shall 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 Plagiarism Detection System.

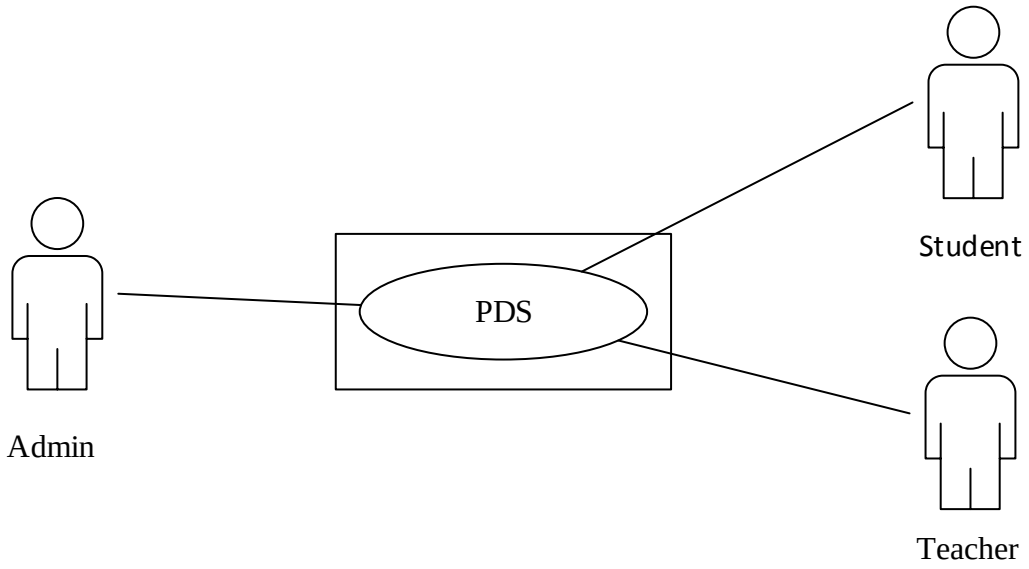


Figure 4.1: Use Case Diagram of PDS

(Level-0)

This is the elaborated form of level-0 for Plagiarism Detection System.

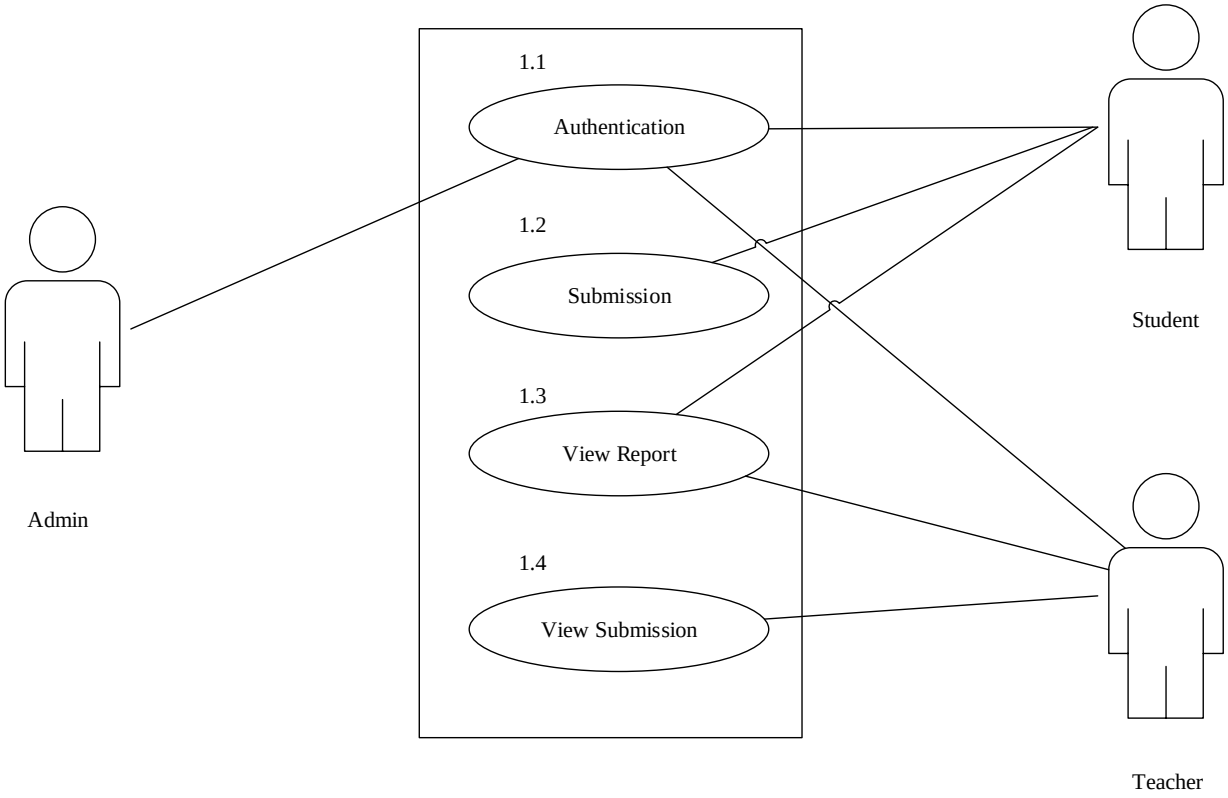


Figure 4.2: Use Case Diagram
of PDS (Level-1)

4.3.1 Authentication

We can further section Authentication system into sub-systems.

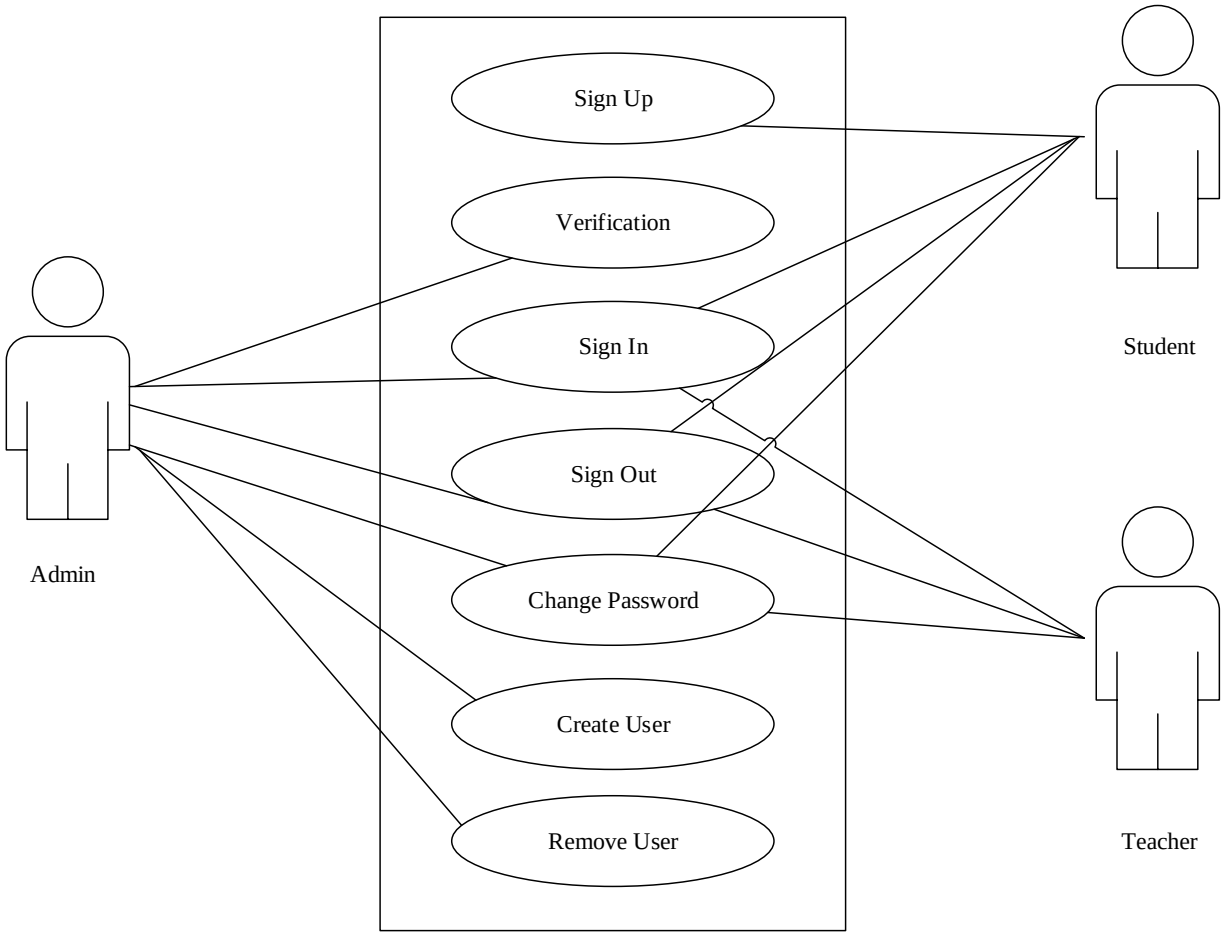


Figure 4.3: Use Case Diagram
of Authentication (Level-1.1)

4.3.1.1 Sign up

Primary Actor	:	Any person on the web
Goal in text	:	To become a Student in this system
Precondition	:	Must have some valid information
Scenario	:	Visit the Sign up page
		Click on ‘Sign up’ button
		Input all information and password
		Proceed to the next activity
Exceptions	:	Unrecognized user email
		Invalid information
		No internet connection
		Not all fields are filled
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

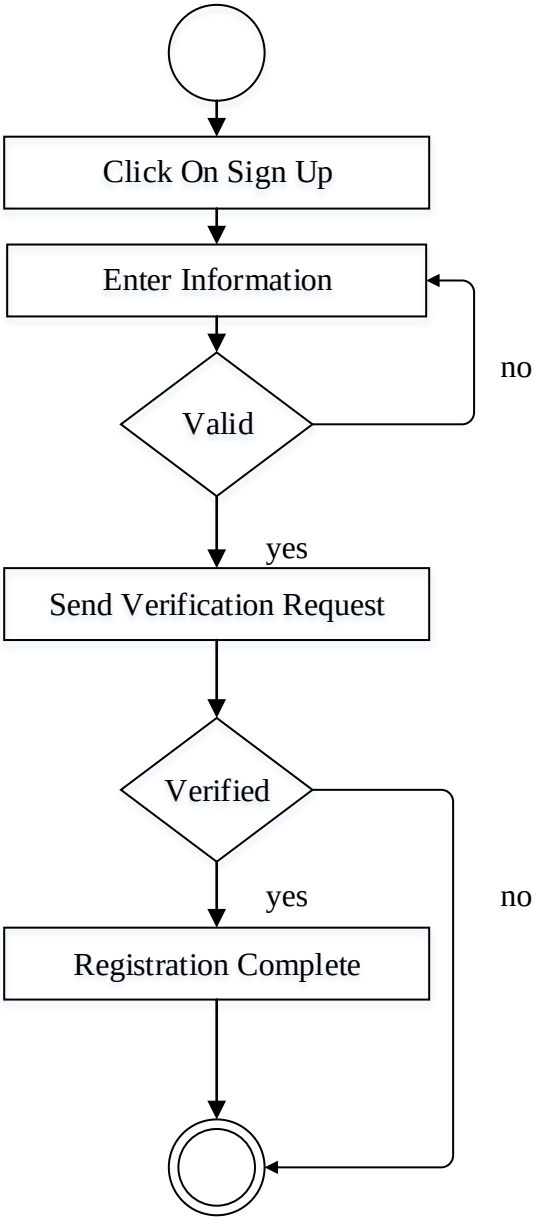


Figure 4.4: Activity Diagram – Sign up

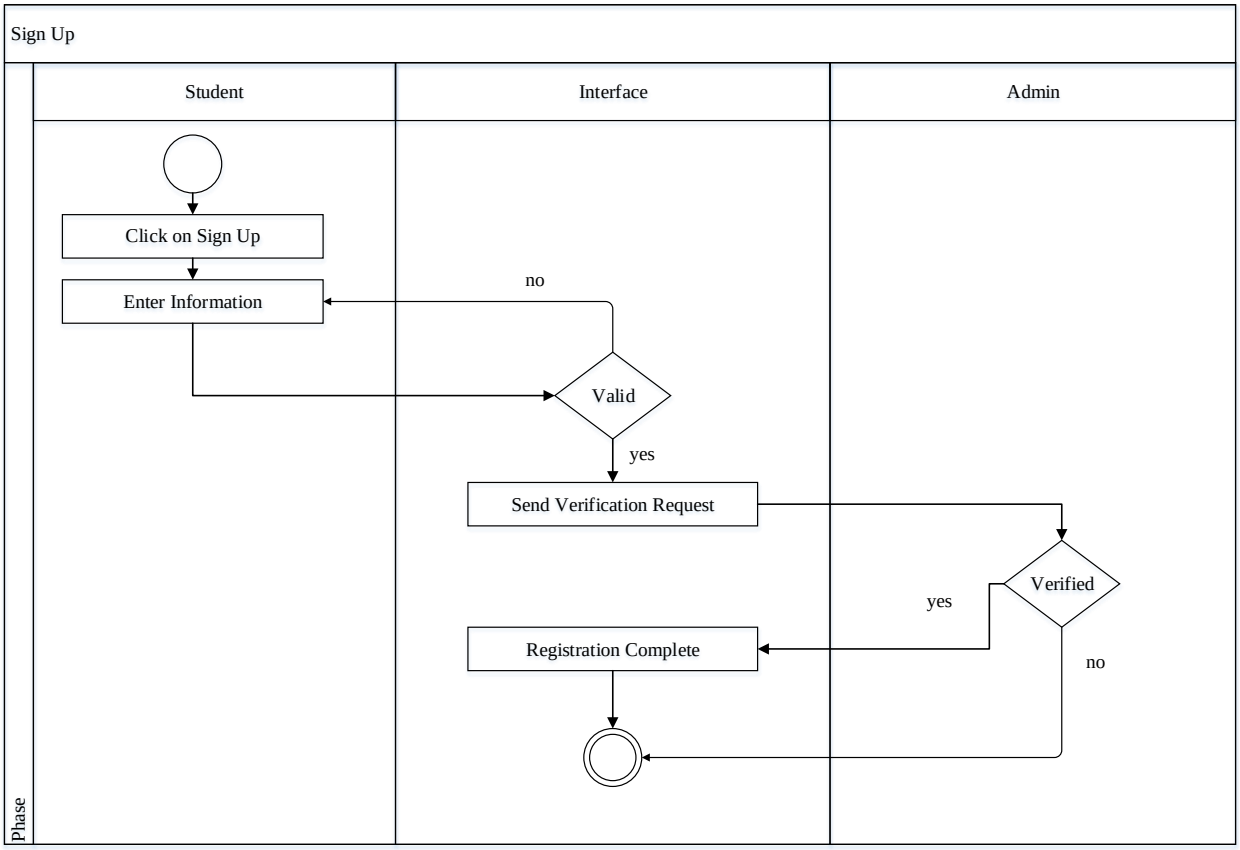


Figure 4.5: Swim-lane Diagram – Sign up

4.3.1.2 Verification

Primary Actor	:	Admin
Goal in text	:	To verify a user to become an active user
Precondition	:	Must have an account on this System
Scenario	:	Visit the Requests page
		Click on ‘Confirm’ or ‘Delete’ button
		Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

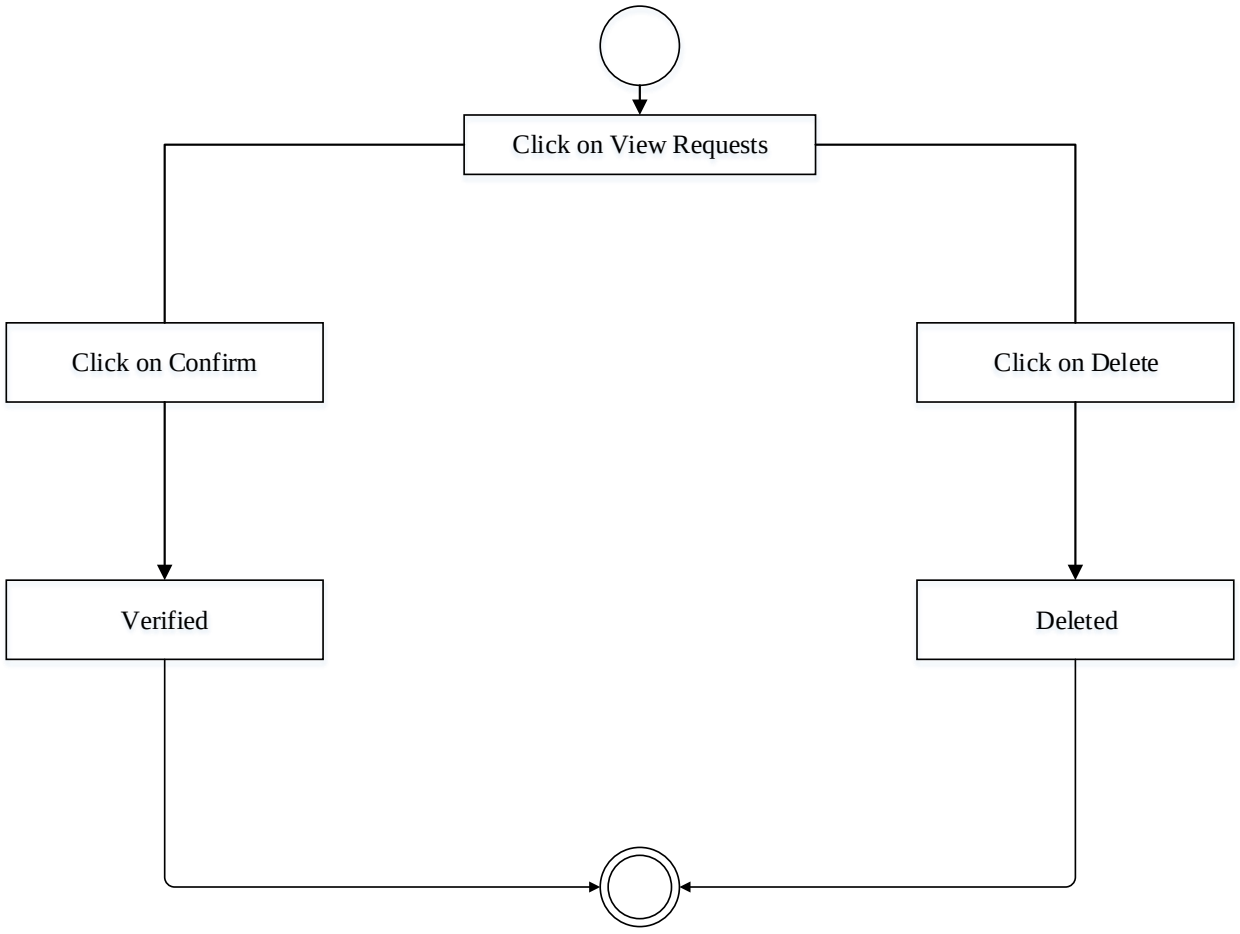


Figure 4.6: Activity Diagram – Verification

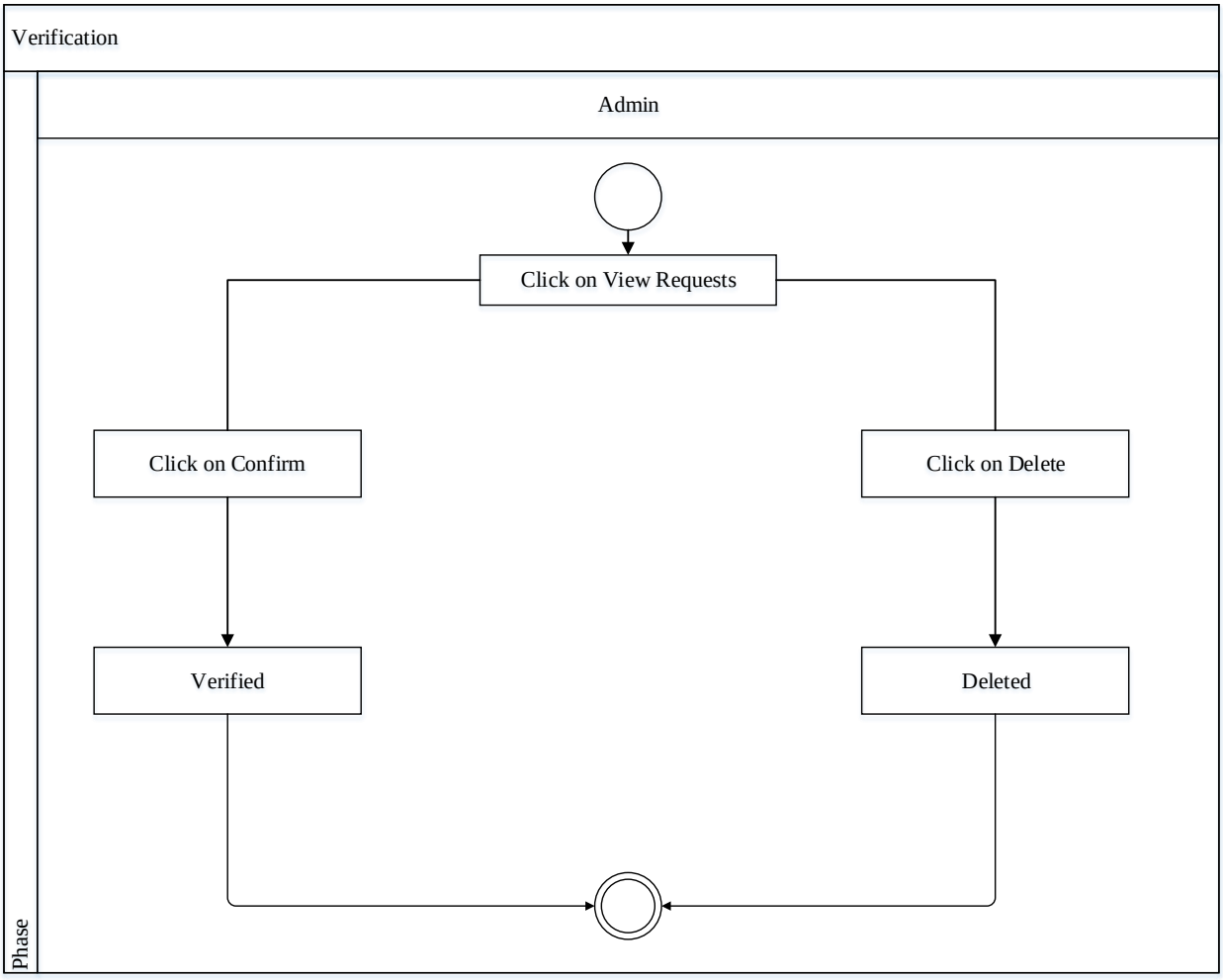


Figure 4.7: Swim-lane Diagram – Verification

4.3.1.3 Sign in

Primary Actor	:	Student Teacher Admin
Goal in text	:	To enter the System
Precondition	:	Must have an account on this System
Scenario	:	Visit the Sign in page Click on ‘Sign in’ button Input email and password Proceed to the next activity
Exceptions	:	Unrecognized user email Invalid information No internet connection Not all fields are filled
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

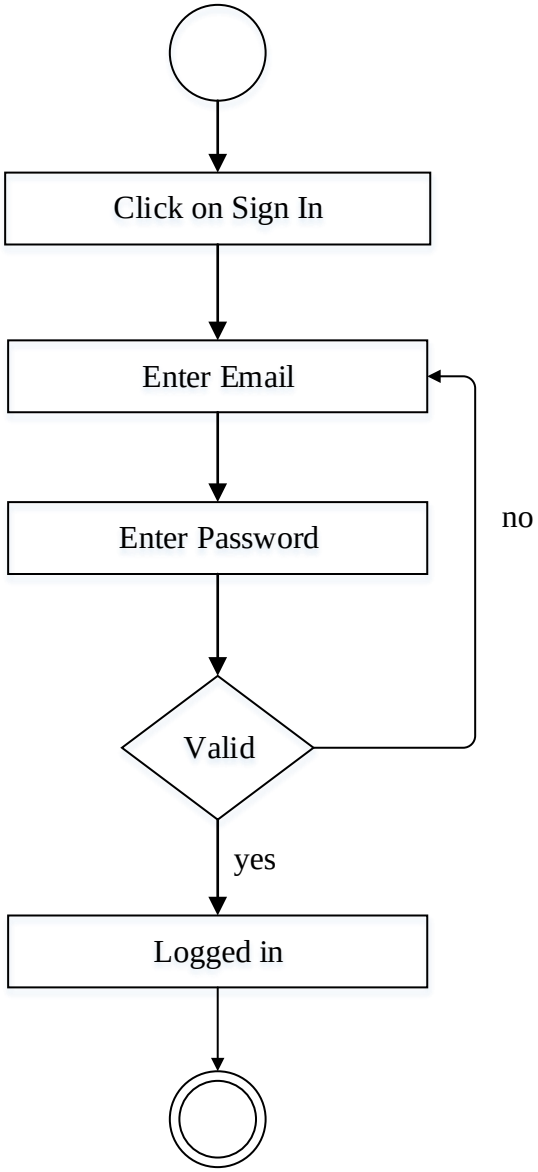


Figure 4.8: Activity Diagram – Sign in

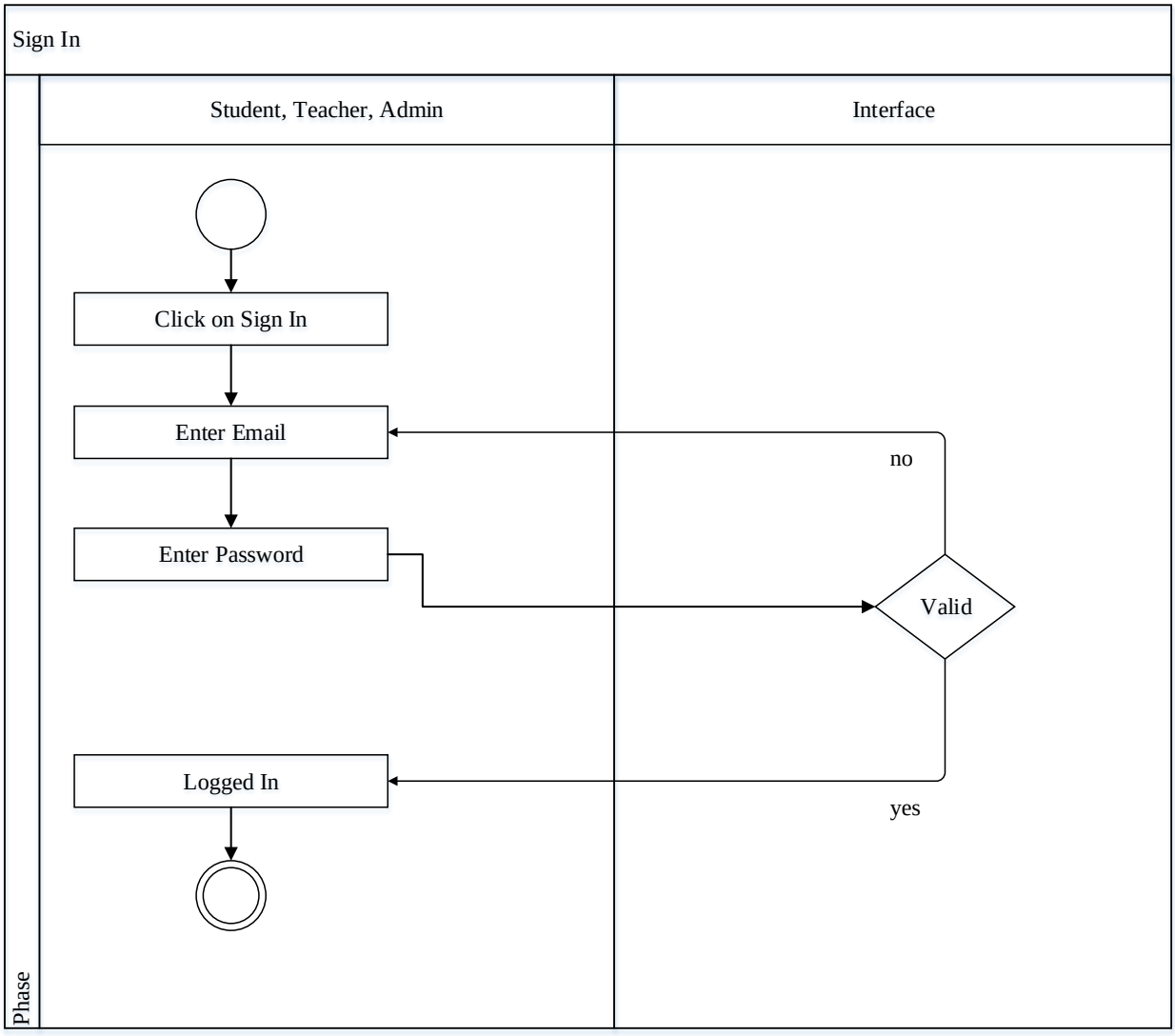


Figure 4.9: Swim-lane Diagram – Sign in

4.3.1.4 Sign out

Primary Actor	:	Student Teacher Admin
Goal in text	:	To exit from the System
Precondition	:	Must have logged in
Scenario	:	Click on ‘Sign out’ button Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

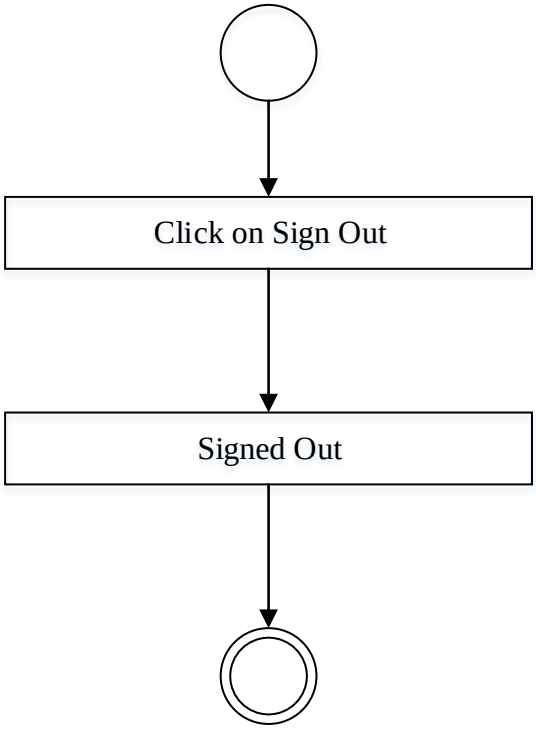


Figure 4.10: Activity Diagram – Sign out

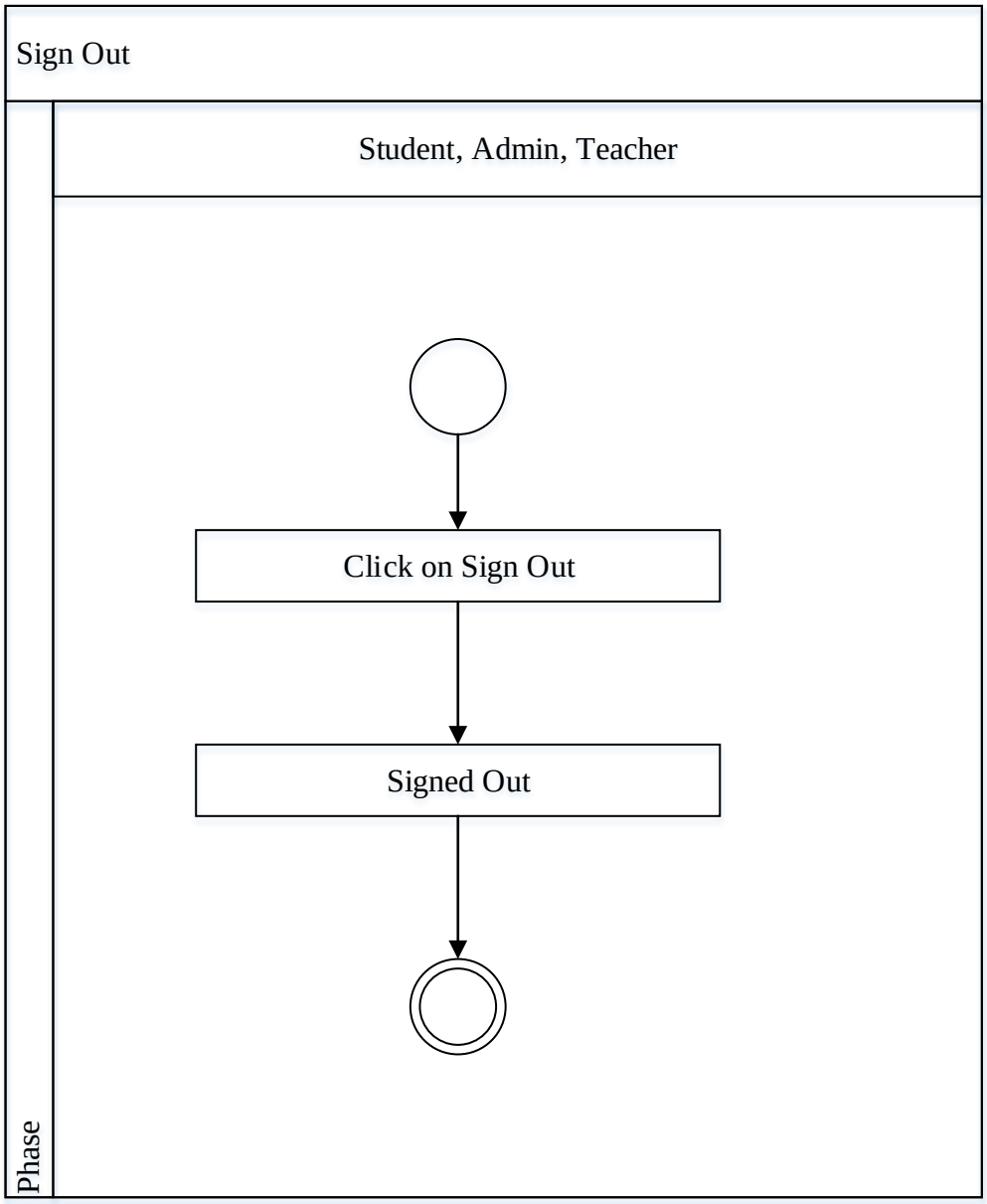


Figure 4.11: Swim-lane Diagram – Sign out

4.3.1.5 Change Password

Primary Actor	:	Student Teacher Admin
Goal in text	:	To change actor's password
Precondition	:	Must have logged in
Scenario	:	Click on 'change password' button Enter current and new password Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

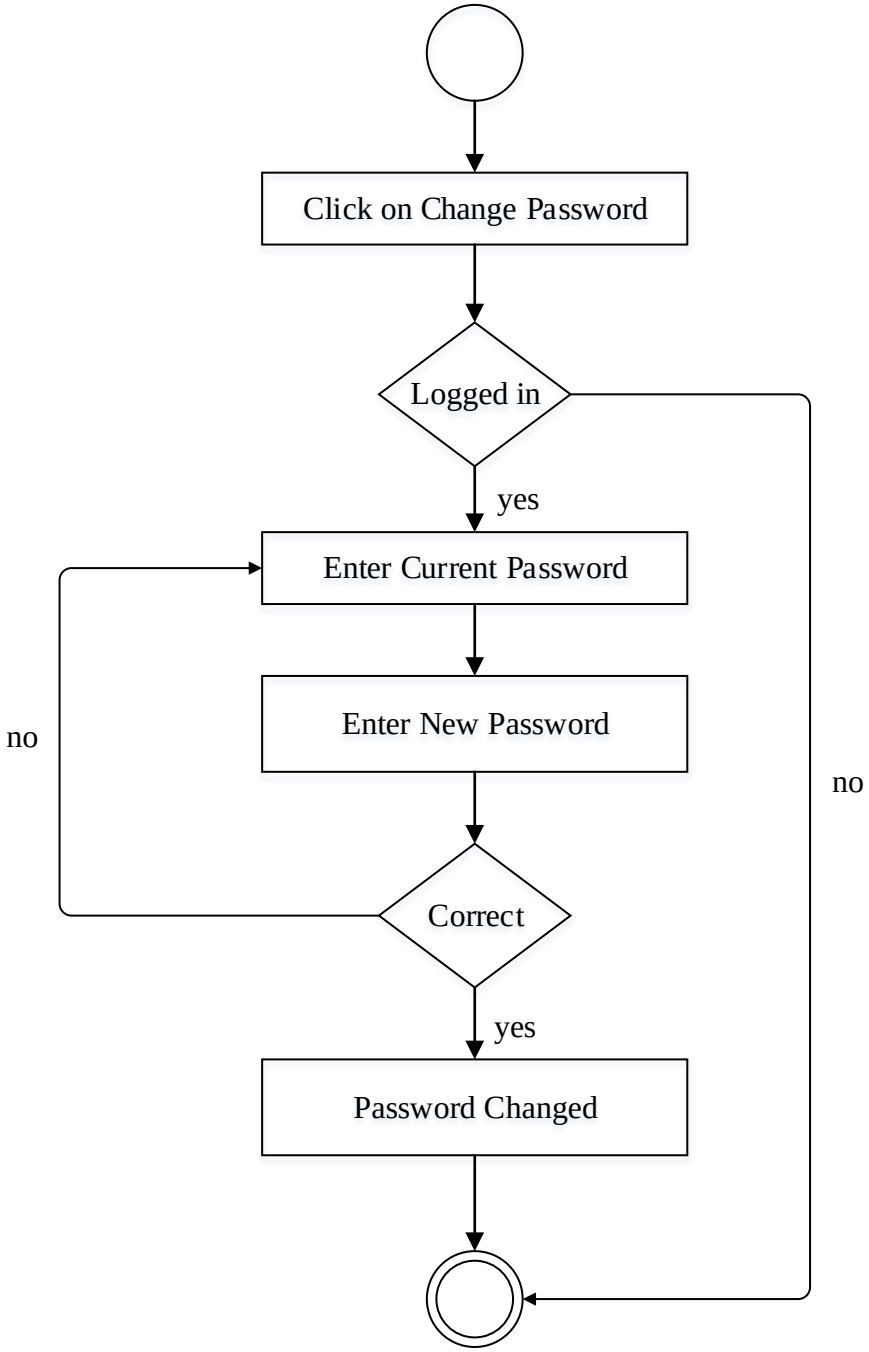


Figure 4.12: Activity Diagram – Change Password

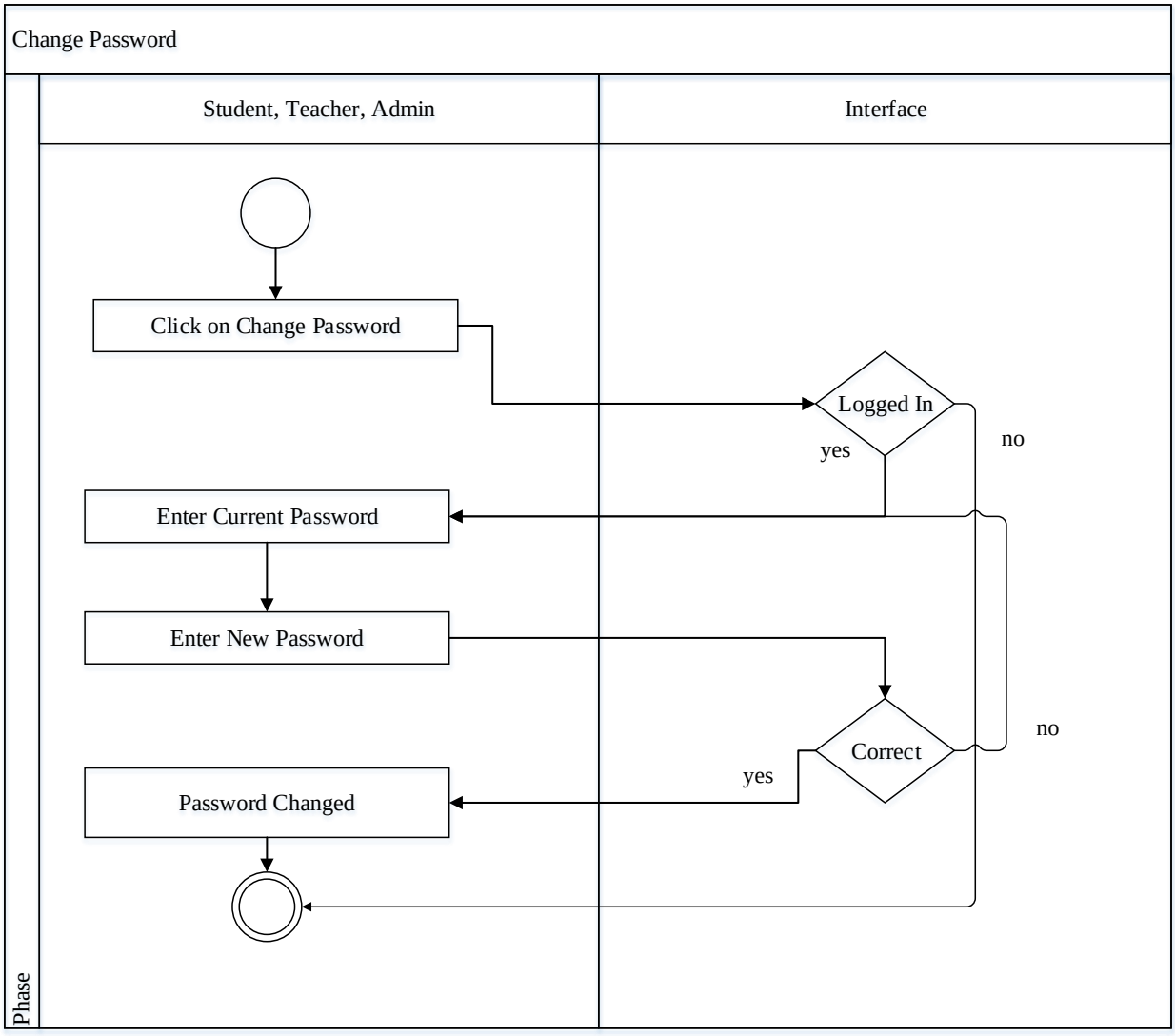


Figure 4.13: Swim-lane Diagram – Change Password

4.3.1.6 Create User

Primary Actor	:	Admin
Goal in text	:	To create a new Student / Teacher account
Precondition	:	Must have logged in
Scenario	:	Click on ‘Create User’ button
		Select user type
		Enter Information
		Proceed to the next activity
Exceptions	:	No internet connection
		Invalid information
		Email in use
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

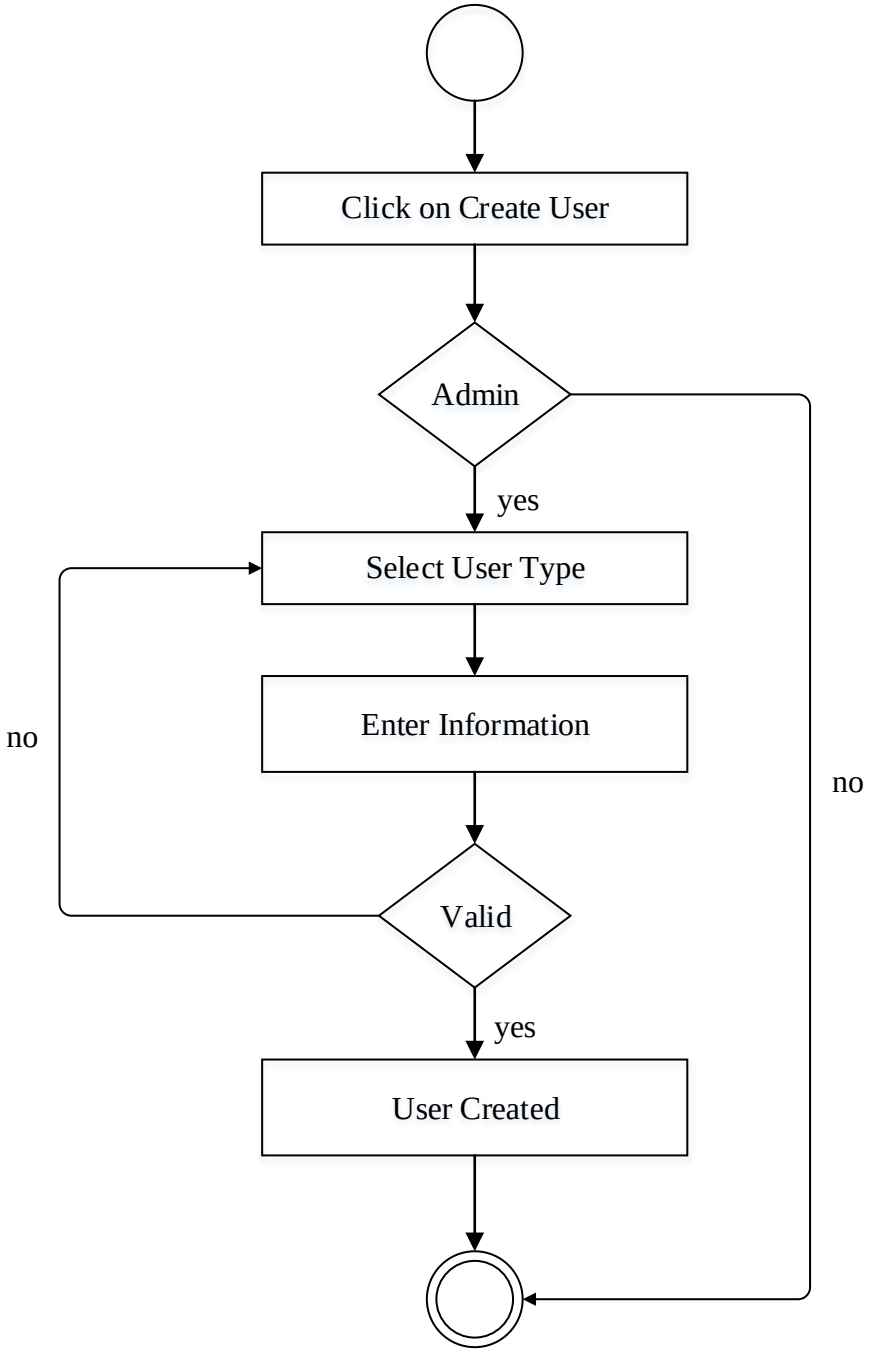


Figure 4.14: Activity Diagram – Create User

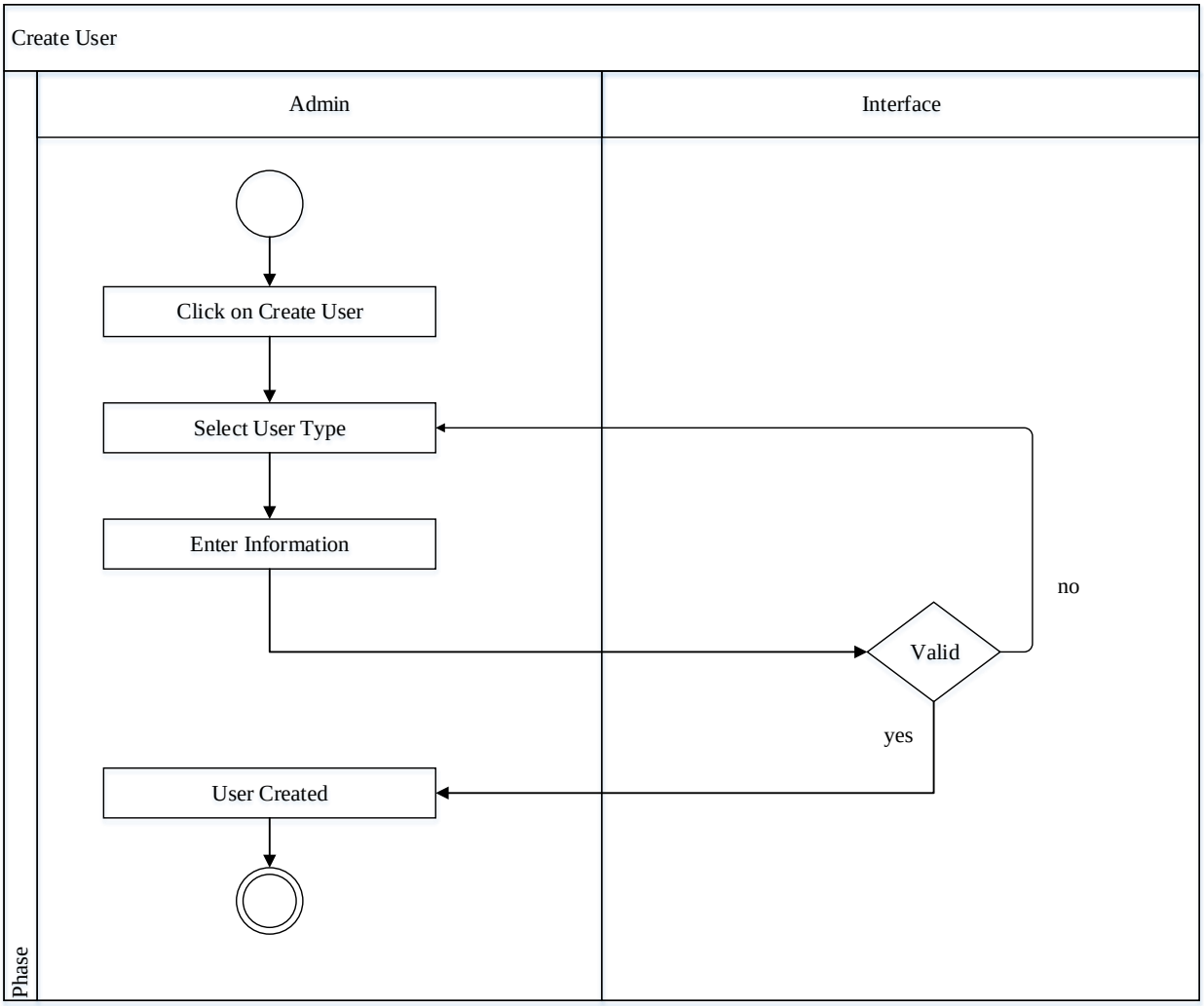


Figure 4.15: Swim-lane Diagram – Create User

4.3.1.7 Remove User

Primary Actor	:	Admin
Goal in text	:	To remove any Student / Teacher account
Precondition	:	Must have logged in
Scenario	:	Click on ‘Remove User’ button
		Select User
		Proceed to the next activity
Exceptions	:	No internet connection
		Remove admin
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

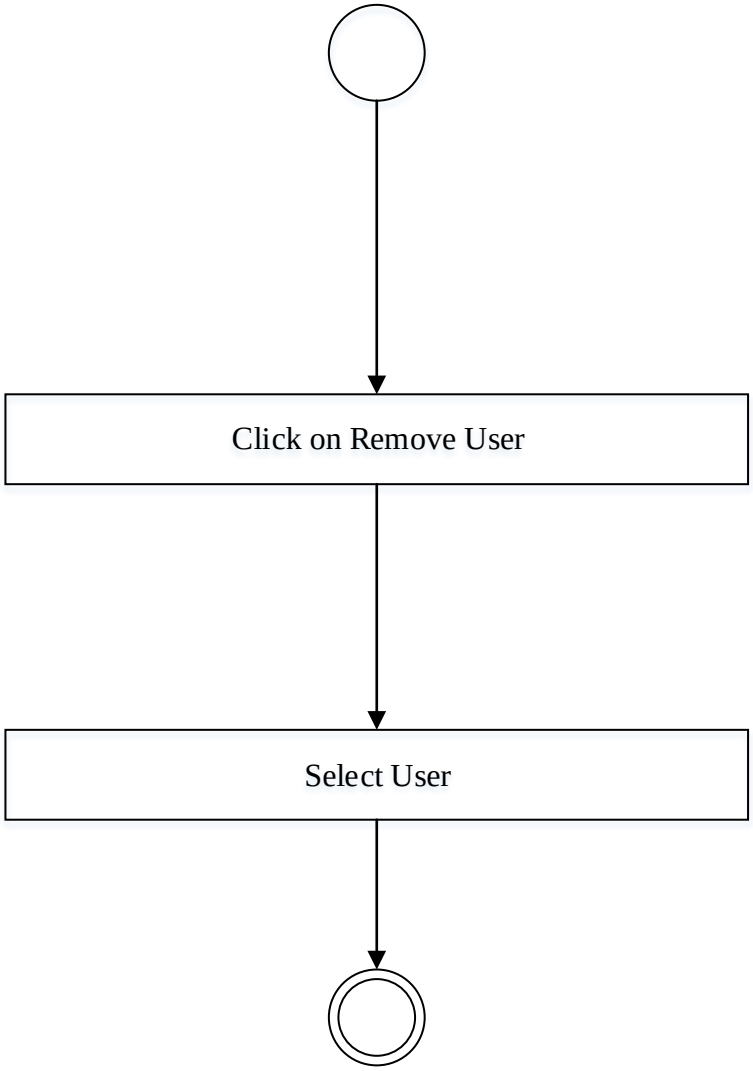


Figure 4.16: Activity Diagram – Remove User

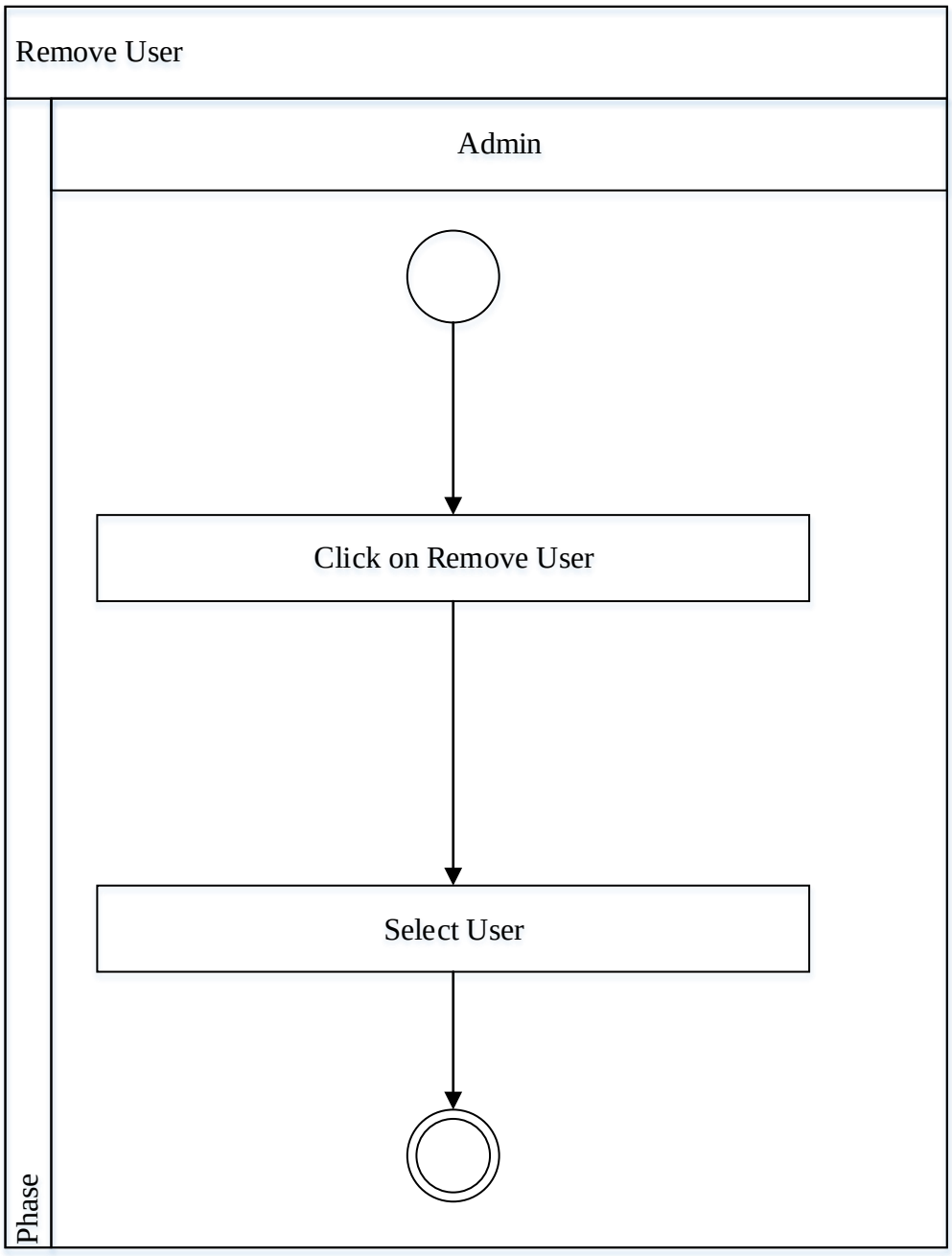


Figure 4.17: Swim-lane Diagram – Remove User

4.3.2 Submission

Primary Actor	:	Student
Goal in text	:	To submit any document
Precondition	:	Must have logged in
Scenario	:	Click on ‘Submit’ button
		Select Submission type
		Select Course code
		Browse document
		Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

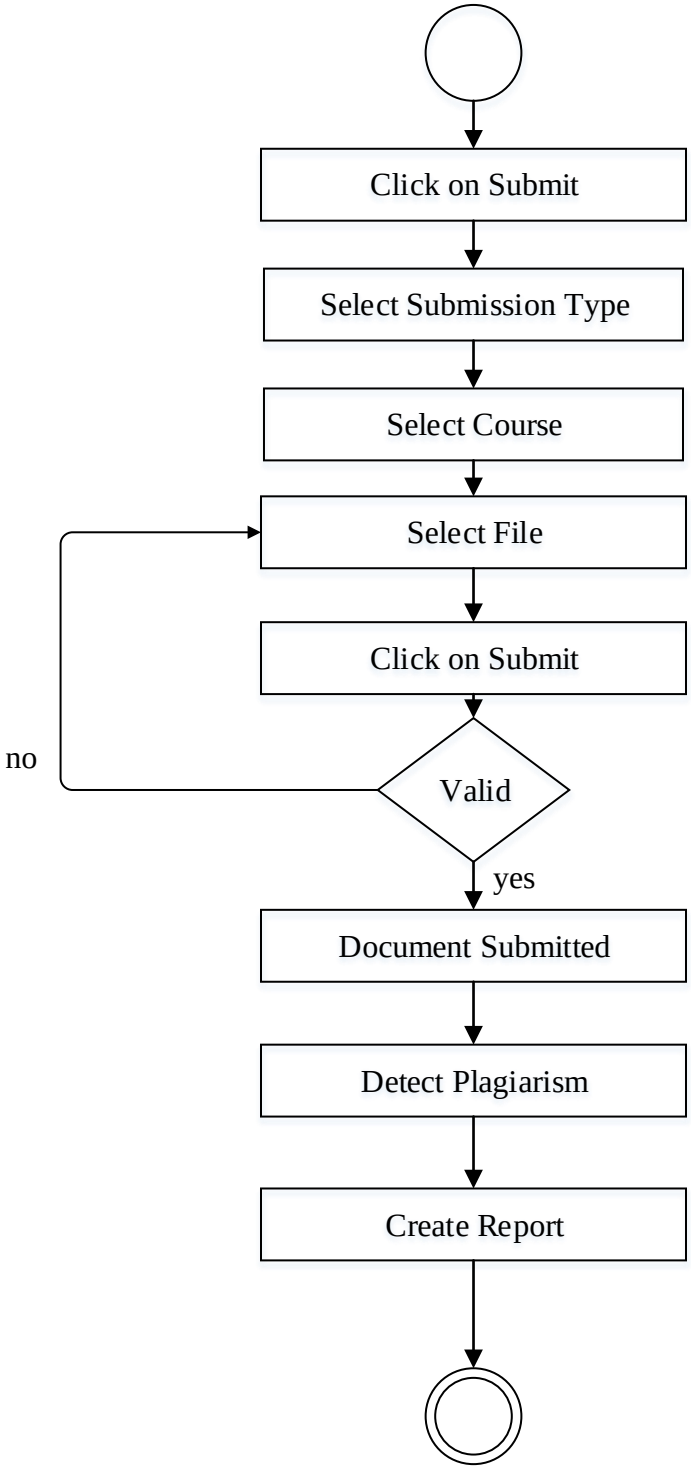


Figure 4.18: Activity Diagram – Submission

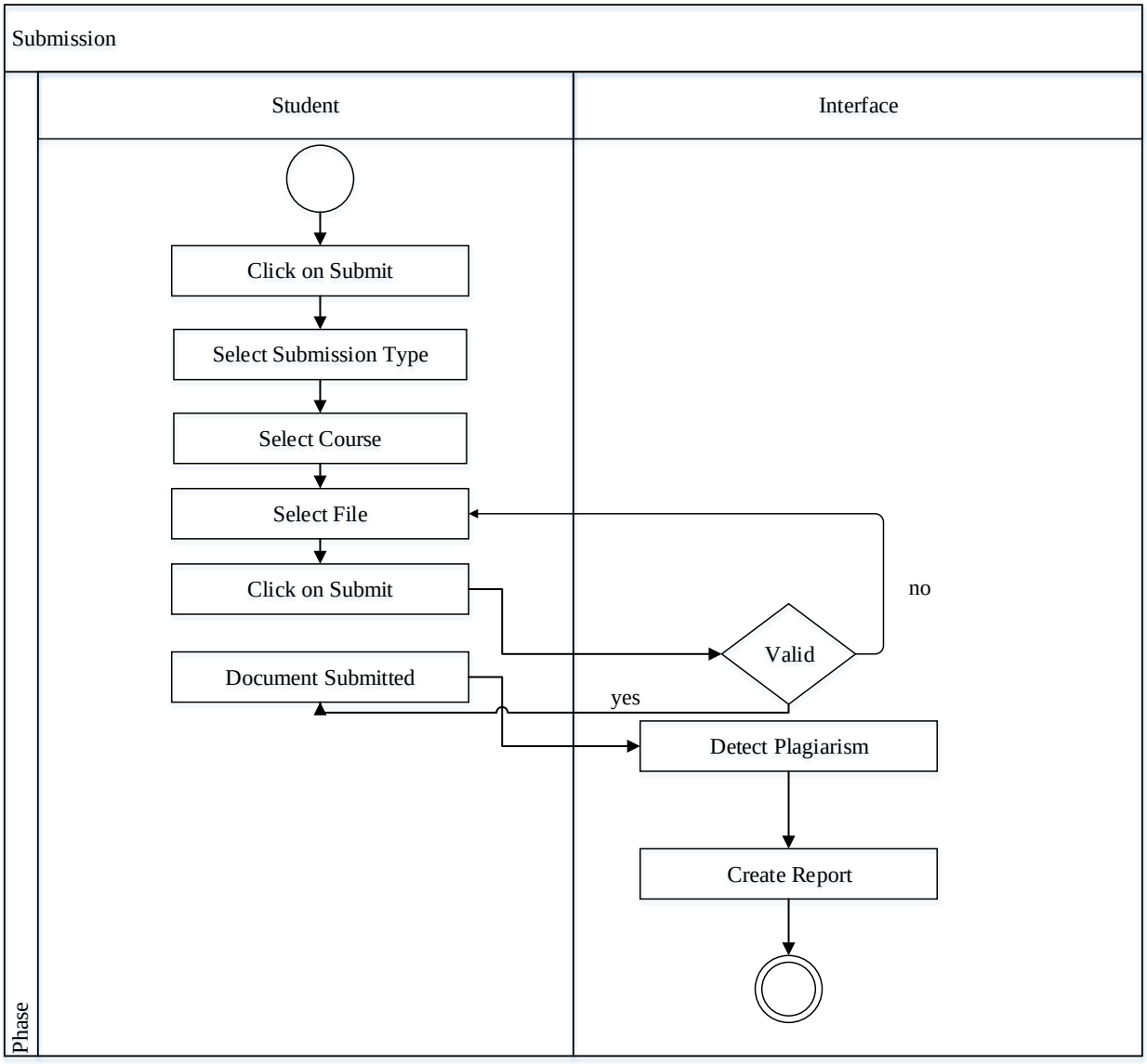


Figure 4.19: Swim-lane Diagram – Submission

4.3.3 View Report

Primary Actor	:	Teacher
Goal in text	:	To view any Report
Precondition	:	Must have logged in
Scenario	:	Click on ‘View Report’ button
		Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

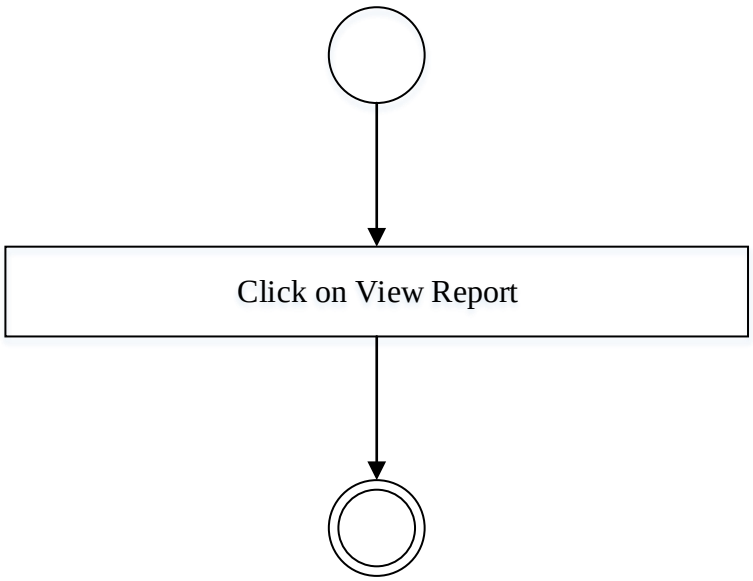


Figure 4.20: Activity Diagram – View Report

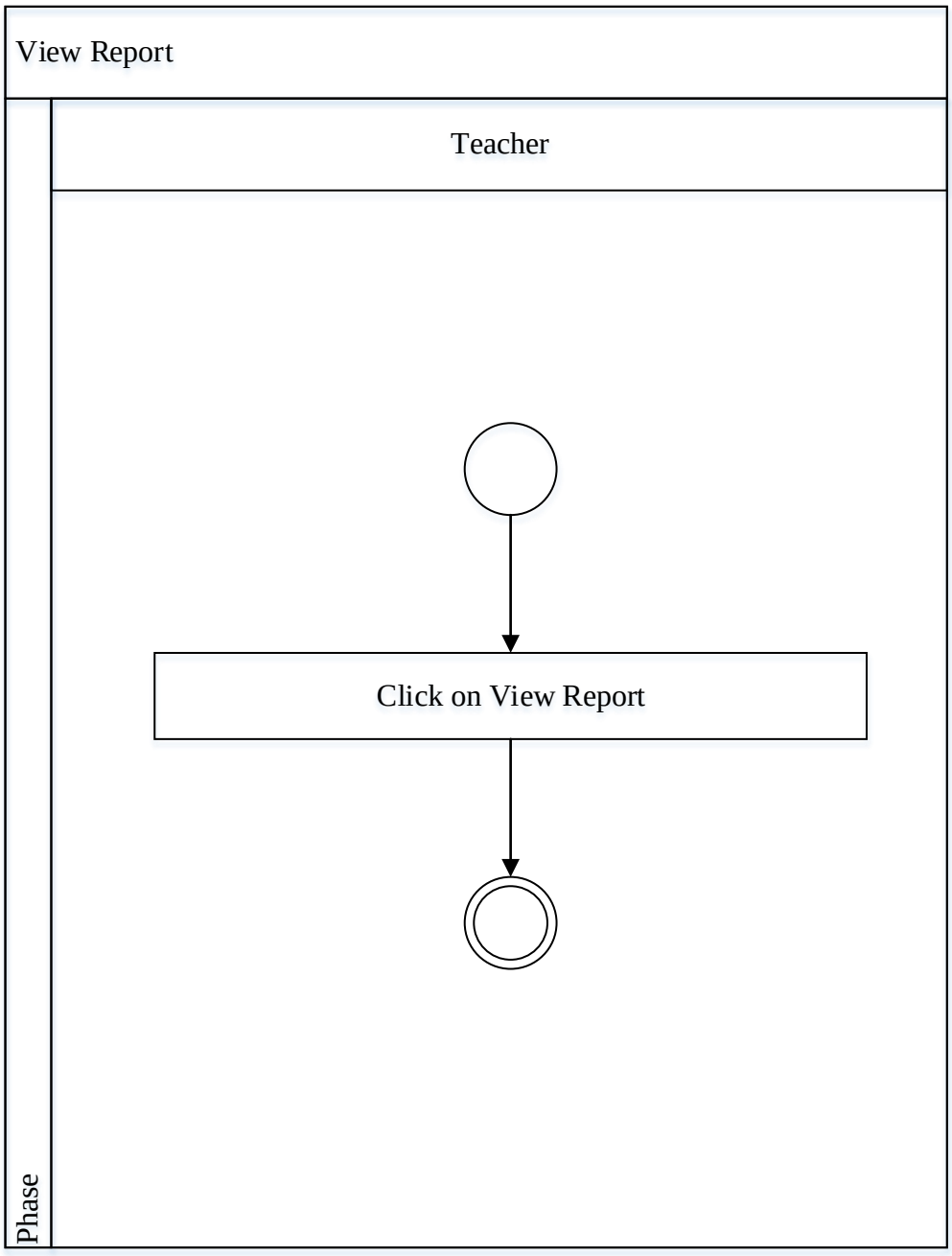


Figure 4.21: Swim-lane Diagram – View Report

4.3.4 View Submission

Primary Actor	:	Teacher
Goal in text	:	To view any Submission
Precondition	:	Must have logged in
Scenario	:	Click on ‘View Submission’ button
		Click on ‘View Document’ button
		Click on ‘Download’ button
		Proceed to the next activity
Exceptions	:	No internet connection
Priority	:	Essential, must be implemented
When available	:	First increment
Frequency	:	Many times a day

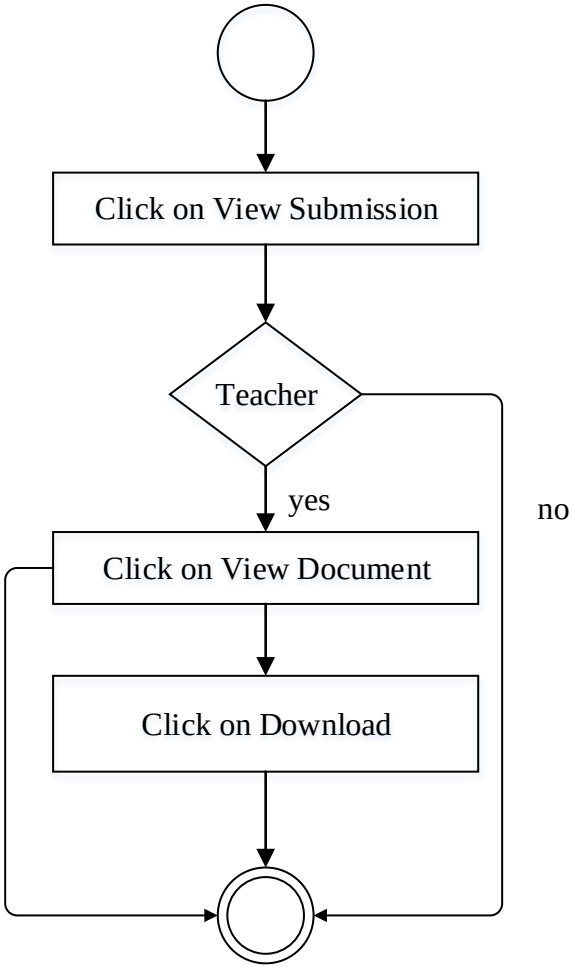


Figure 4.22: Activity Diagram –
View Submission

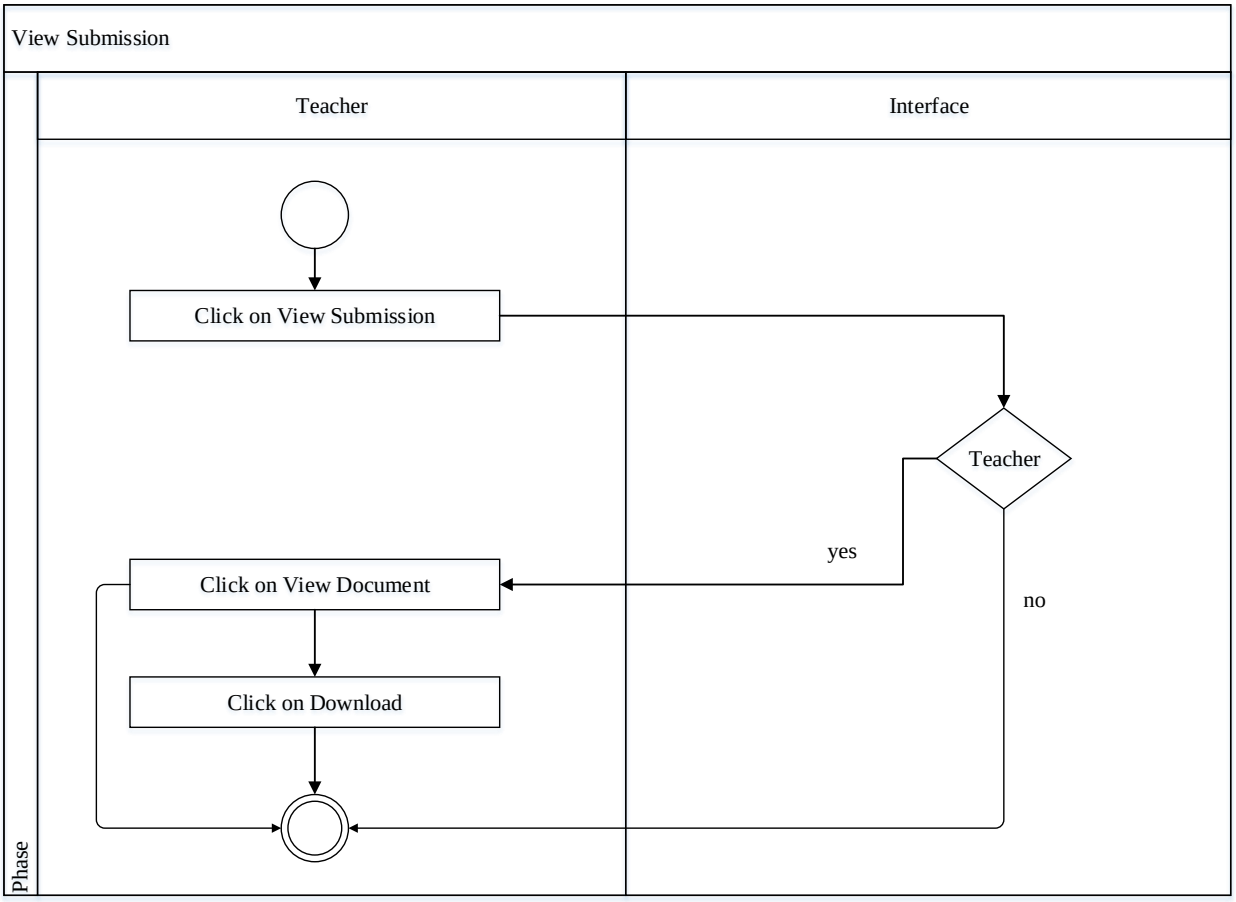


Figure 4.23: Swim-lane Diagram – View Submission

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, faculty, department, reg. no., class, email, session, password, roll	Potential Data Object	Accepted
Name		An Attribute of Student, Teacher, Admin	Rejected
Faculty		An Attribute of Student, Teacher, Admin	Rejected
Department		An Attribute of Student	Rejected
Reg. no.		An Attribute of Student	Rejected
Class		An Attribute of Student	Rejected
Email		An Attribute of Student, Teacher, Admin	Rejected
Session		An Attribute of Student	Rejected

Password		An Attribute of Student, Teacher, Admin	Rejected
Roll		An Attribute of Student	Rejected
Form	Name, faculty, department, reg. no., class, email, session, password, roll	Potential Data Object	Accepted
Admin	Name, email, password	Potential Data Object	Accepted
Information	Name, faculty, department, reg. no., class, email, session, roll, password	Duplicate of Form	Rejected
Teacher	Name, faculty, department, email, password, designation	Potential Data Object	Accepted
Designation		An Attribute of Teacher	Rejected
Document		An Attribute of Submission	Rejected
Submission Type		An Attribute of Submission	Rejected
Course Code		An Attribute of Submission	Rejected
Report		An Attribute of Submission	Rejected
Submission	Document, submission type, course code, report	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 + Faculty + Department + Reg. no. + Class + Email + Session + Password + Roll

Form = Name + Faculty + Department + Reg. no. + Class + Email + Session + Password + Roll

Admin = Name + Email + Password

Teacher = Name + Faculty + Department + Email + Password + Designation

Submission = Submission type + Course code + Document + Report

- A unique submission id is needed to identify each submission uniquely, this id will be auto incremented starting from 0. Besides this system will keep track who submitted this document. So email of Student is also saved with each submission.

Submission = Submission id + Email + Submission type + Course code + Document + Report

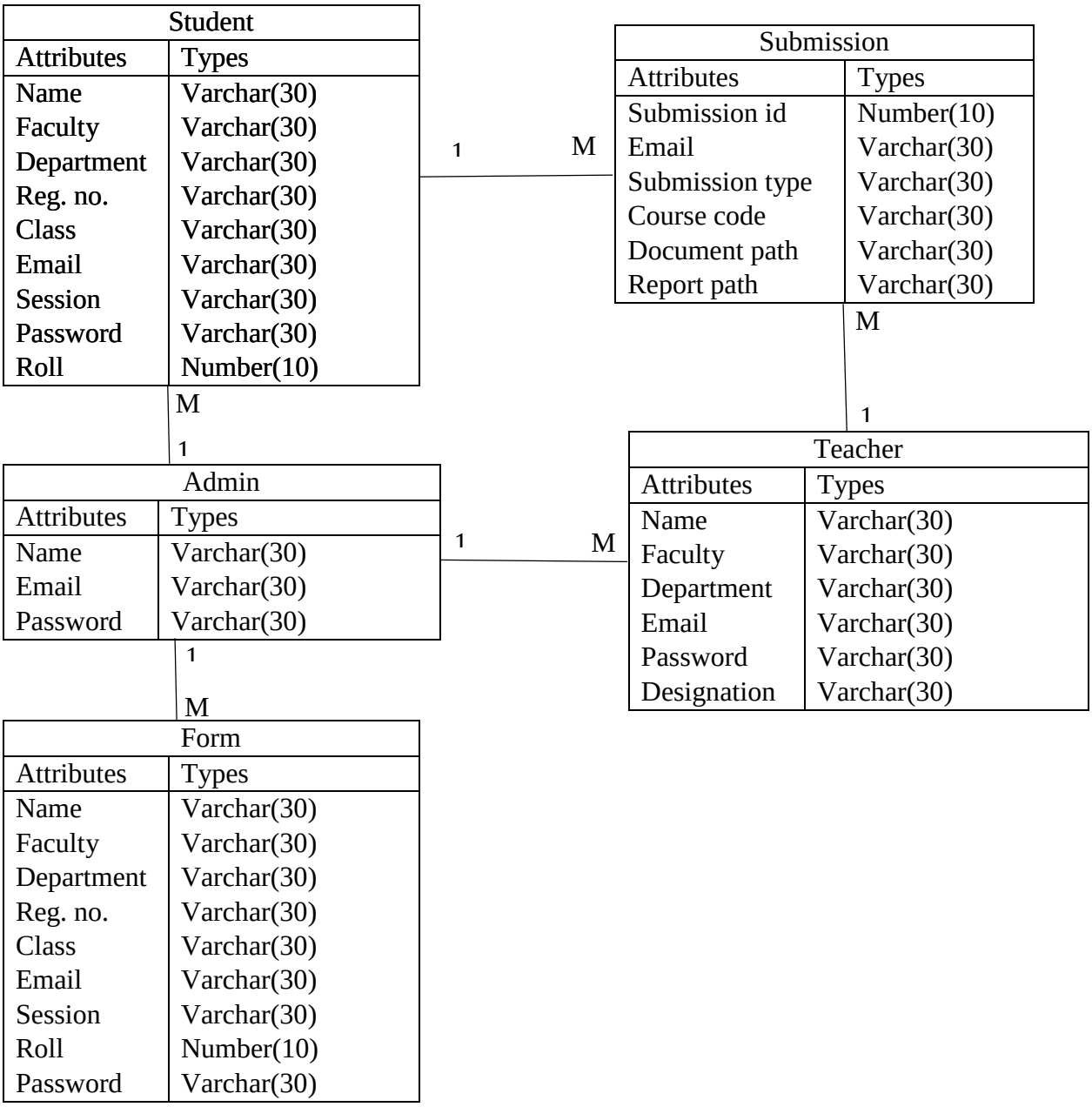
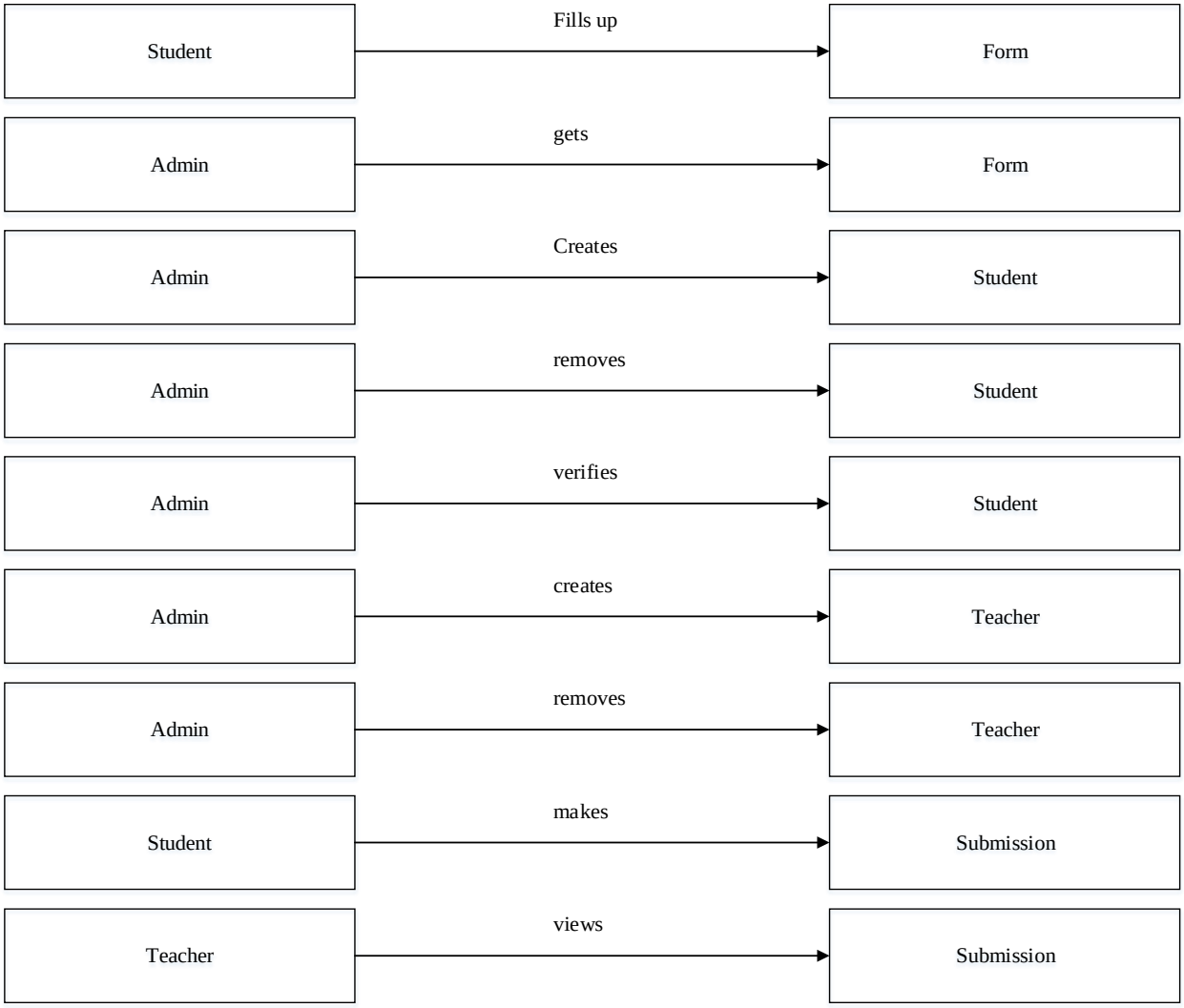


Fig 5.1: Database Schema

5.4 Relationship Between Data Objects

Here I have shown pair wise relation between two entities.



5.5 Entity Relationship(ER) Diagram

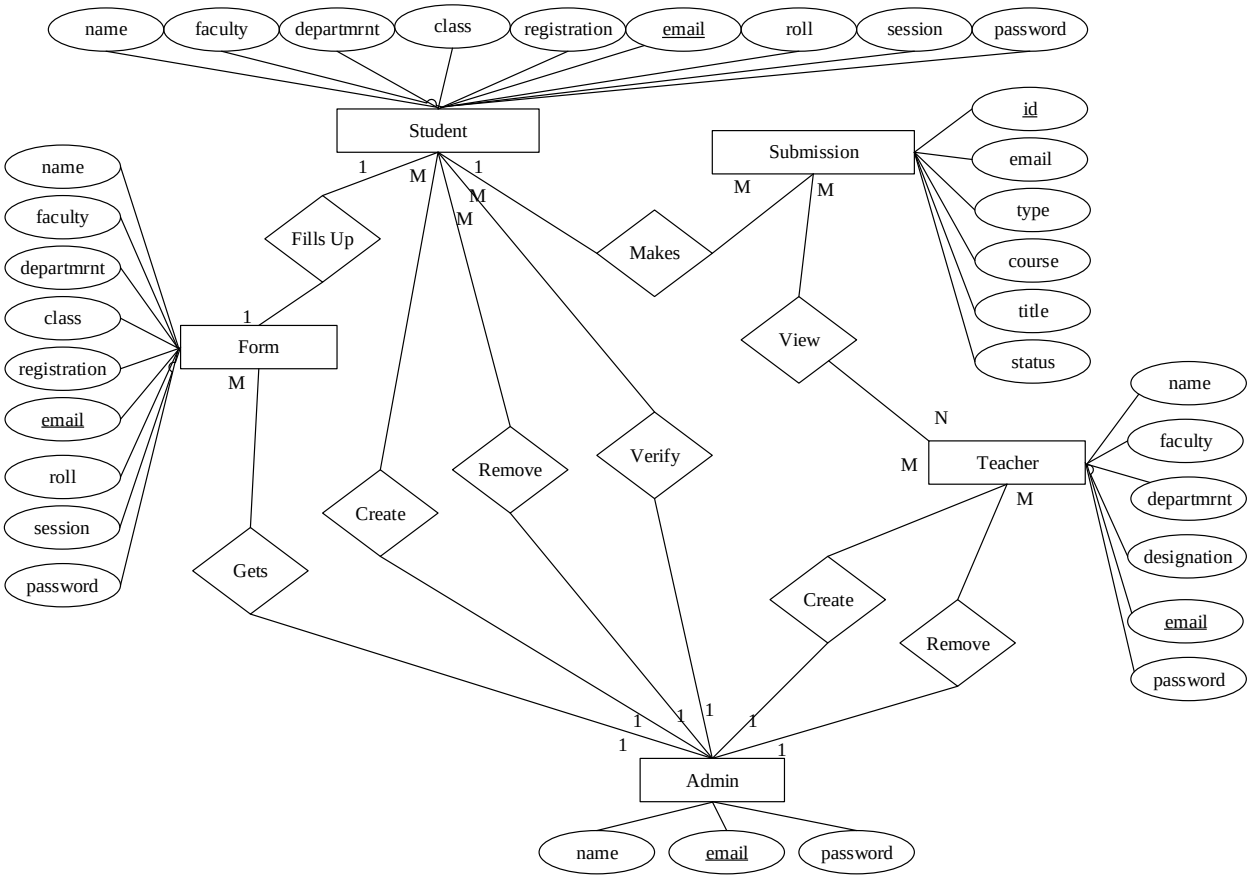


Figure 5.2: Entity Relationship Diagram

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
Faculty		No
Department		No
Reg. no.		No
Class		No
Email		No
Session		No
Password		No
Roll		No
Form	7	Yes
Admin	4, 5, 7	Yes
Information	7	Yes
Teacher	4, 5, 7	Yes
Designation		No
Document		No
Submission Type		No

Course Code		No
Report		No
Submission	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
Form	1, 3, 4	No
Admin	1, 2, 3, 6	Yes
Information	1, 3, 4	No
Teacher	1, 2, 3, 4, 5, 6	Yes
Submission	1, 3, 4	No

Primary classes:

1. Student
2. Admin
3. Teacher

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, faculty, department, reg. no., class, email, session, password, roll
Admin	Name, email, password
Teacher	Name, faculty, department, email, password, designation
Database	All attributes

6.5 Method Identification

Table 6.4: Method Identification

Class Name	Methods
Student	Get() Set() Sign_in() Sign_out() Change_password() Submit()
Admin	Get() Set() Sign_in() Sign_out() Change_password() Create_user() Remove_user() Verify_user_requesst() Delete_user_request()
Teacher	Get() Set() Sign_in() Sign_out() Change_password() View_document() View_report() Download_document()
Database	Insert() Delete() Update() View()

6.6 Class Card

Student	
Attributes	Methods
Name Faculty Department Reg. no. Class Email Session Password Roll	Get() Set() Sign_in() Sign_out() Change_password() Submit()
Responsibilities	Collaborative Class
Submission of documents	Database

Admin	
Attributes	Methods
Name Email Password	Get() Set() Sign_in() Sign_out() Change_password() Create_user() Remove_user() Verify_user_requesst() Delete_user_request()
Responsibilities	Collaborative Class
Creation of new user account	Database
Removal of existing user	Database
Verification of user request	Database
Deletion of user request	Database

Teacher	
Attributes	Methods
Name Faculty Department Email Password Designation	Get() Set() Sign_in() Sign_out() Change_password() View_document() View_report() Download_document()
Responsibilities	Collaborative Class
Viewing documents	Database
Viewing report	Database
Downloading	Database

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

After analyzing each classes I found some similar attributes and methods. So I can consider a super class named of them which will hold common attributes and common methods. Other classes will be sub class of this super class.

Common Attributes: Name, Email, Password
Common Methods: sign_in (), sign_out(), change_password()

6.7 CRC Diagram

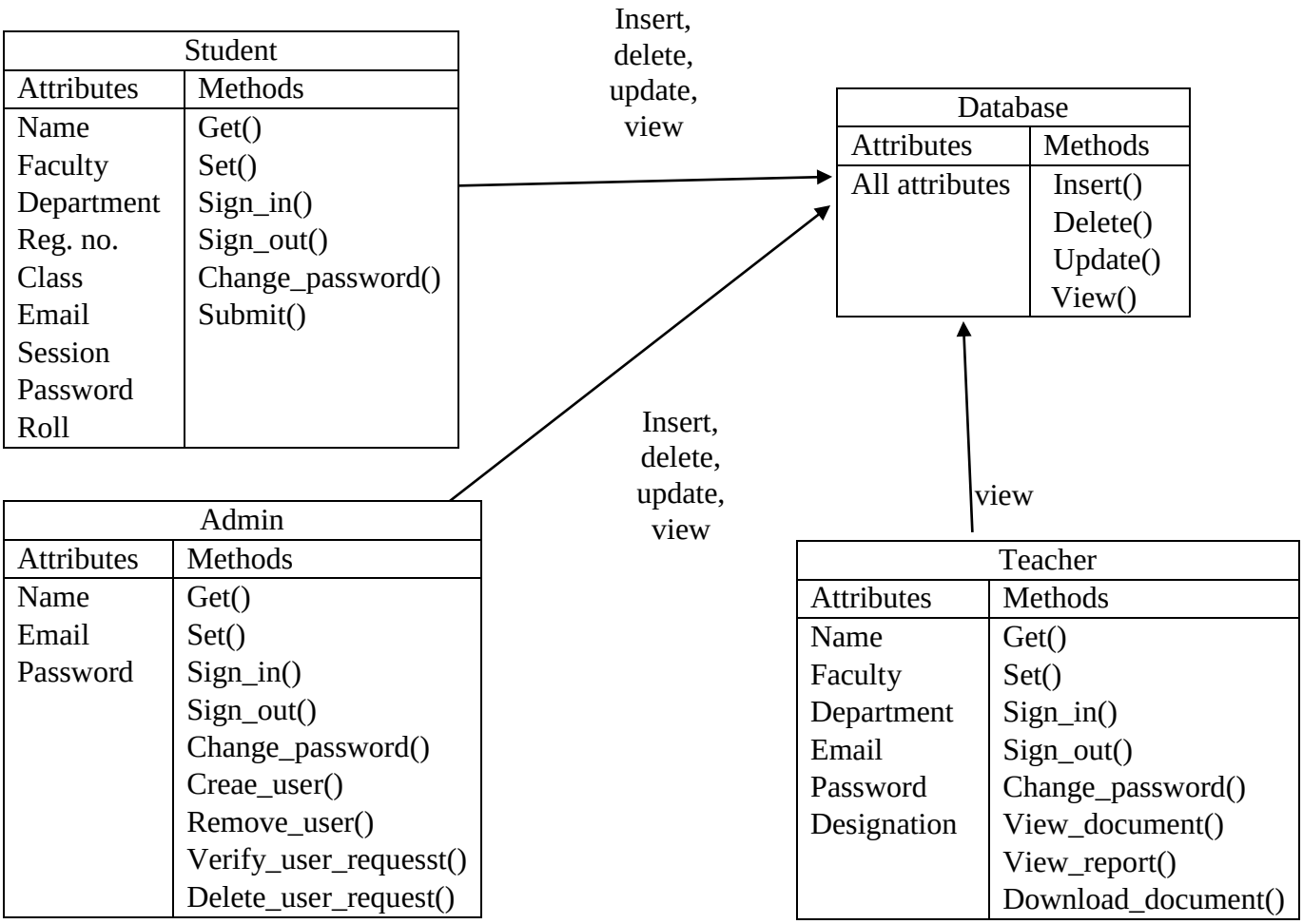


Fig 6.1: CRC diagram

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.

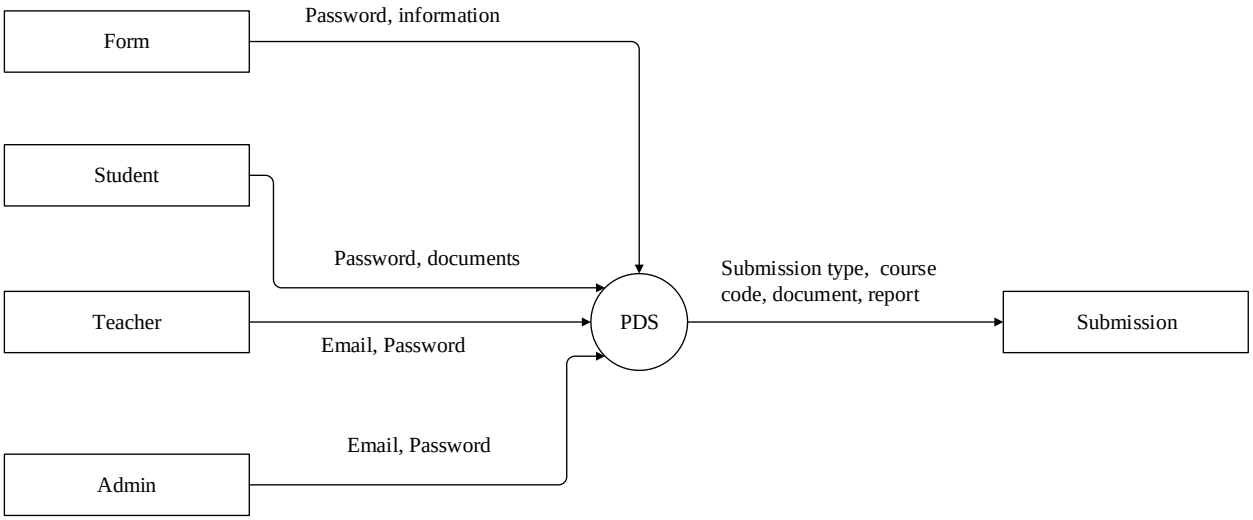


Fig 7.1: DFD Level 0

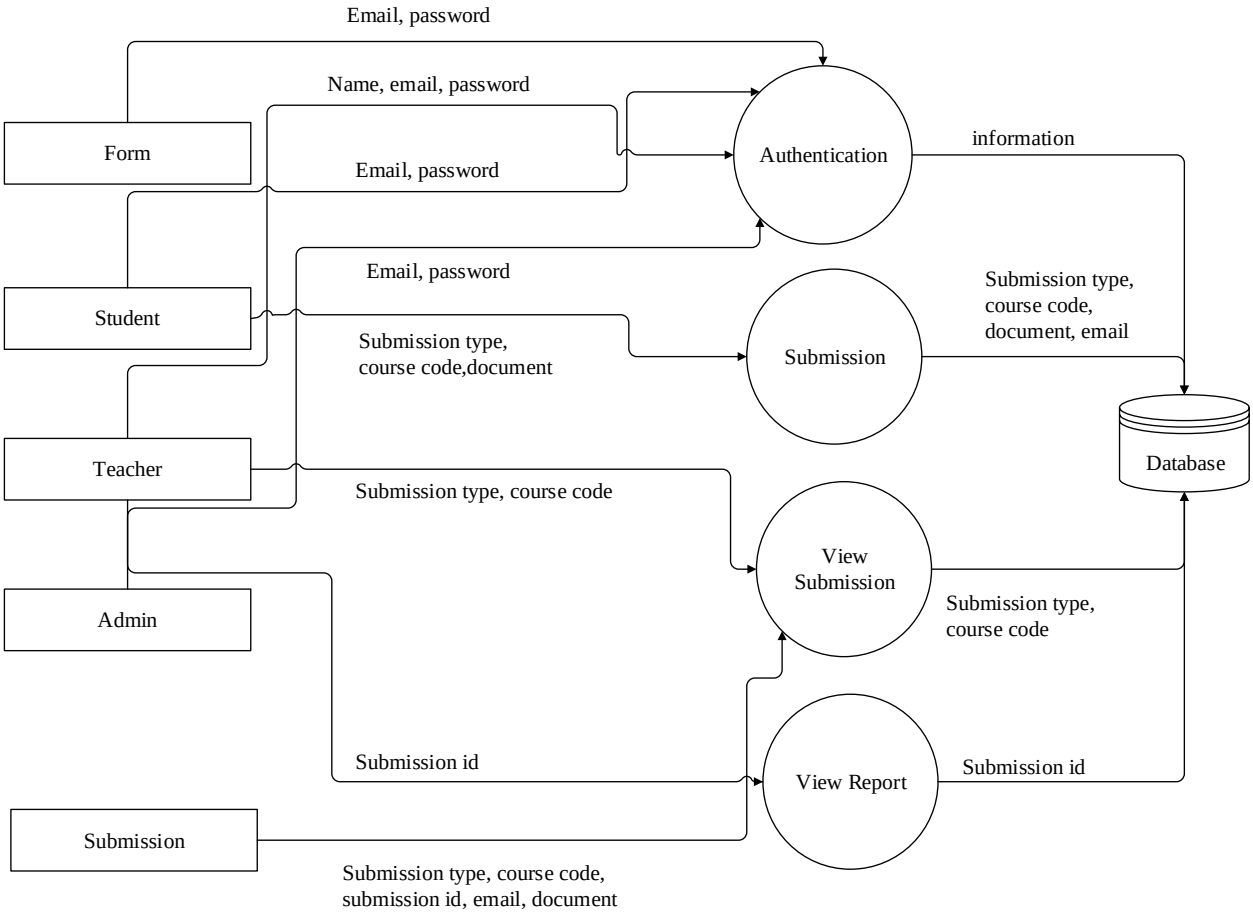


Fig 7.2: DFD Level 1

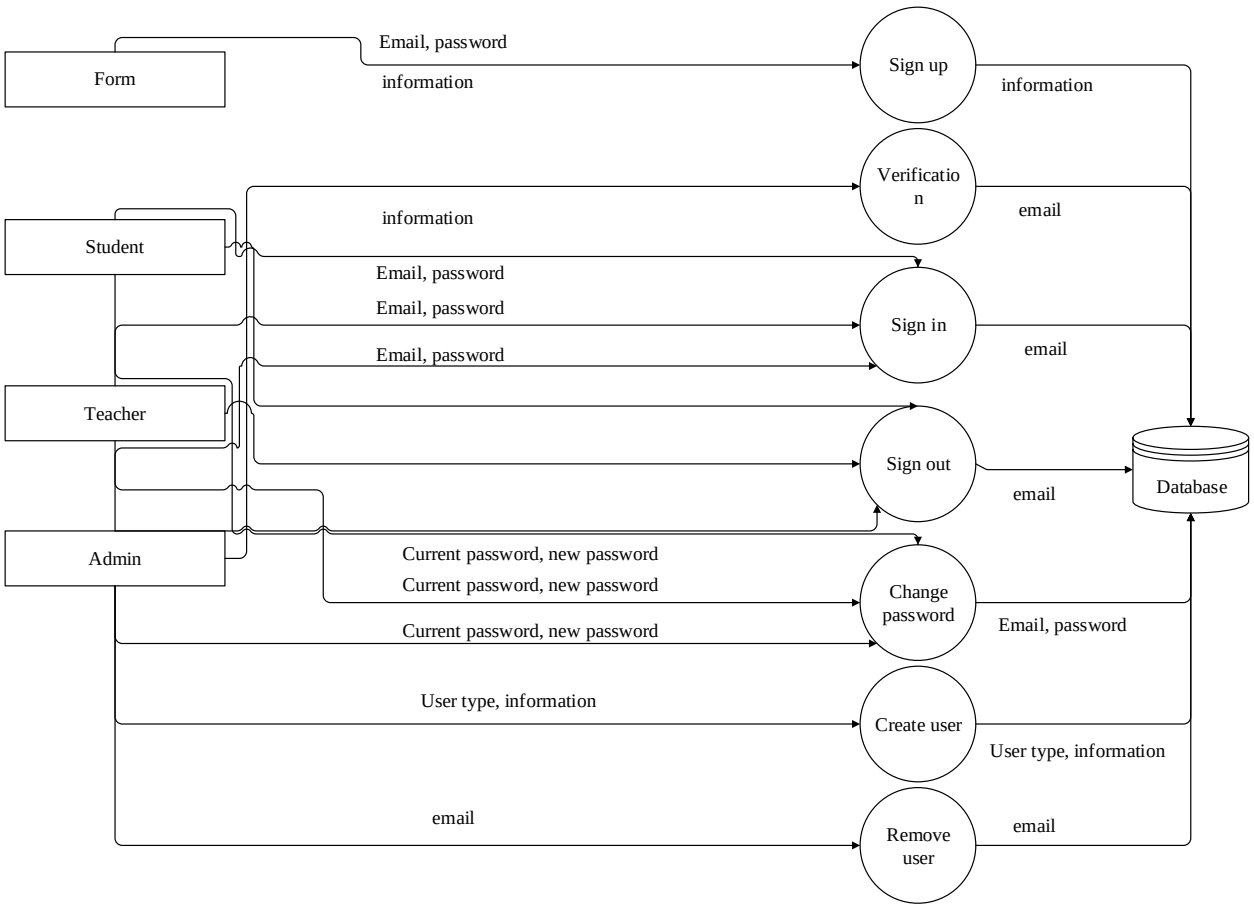


Fig 7.3: DFD Level 1.1 Authentication

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 I 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
Verify	Admin	Student, Database
Create User	Admin	Student, Teacher, Database
Remove User	Admin	Student, Teacher, Database
Change password	Student, Teacher, Admin	Database
Sign in	Student, Teacher, Admin	Database
Sign out	Student, Teacher, Admin	Database
Submit	Student	Database
View Document	Teacher	Database
Download Document	Teacher	Database
View Report	Teacher	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.

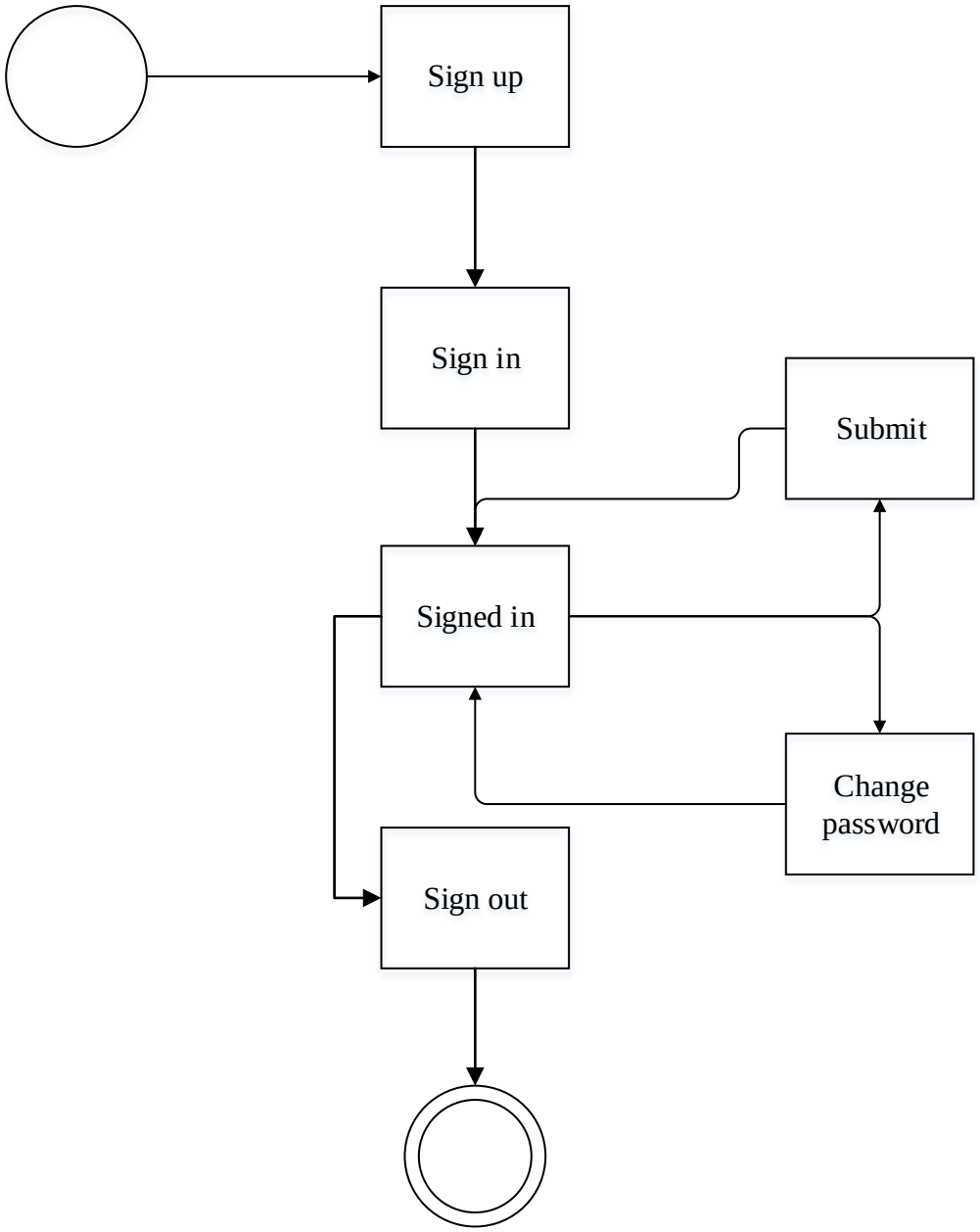


Fig 8.1: State Transition Diagram-Student

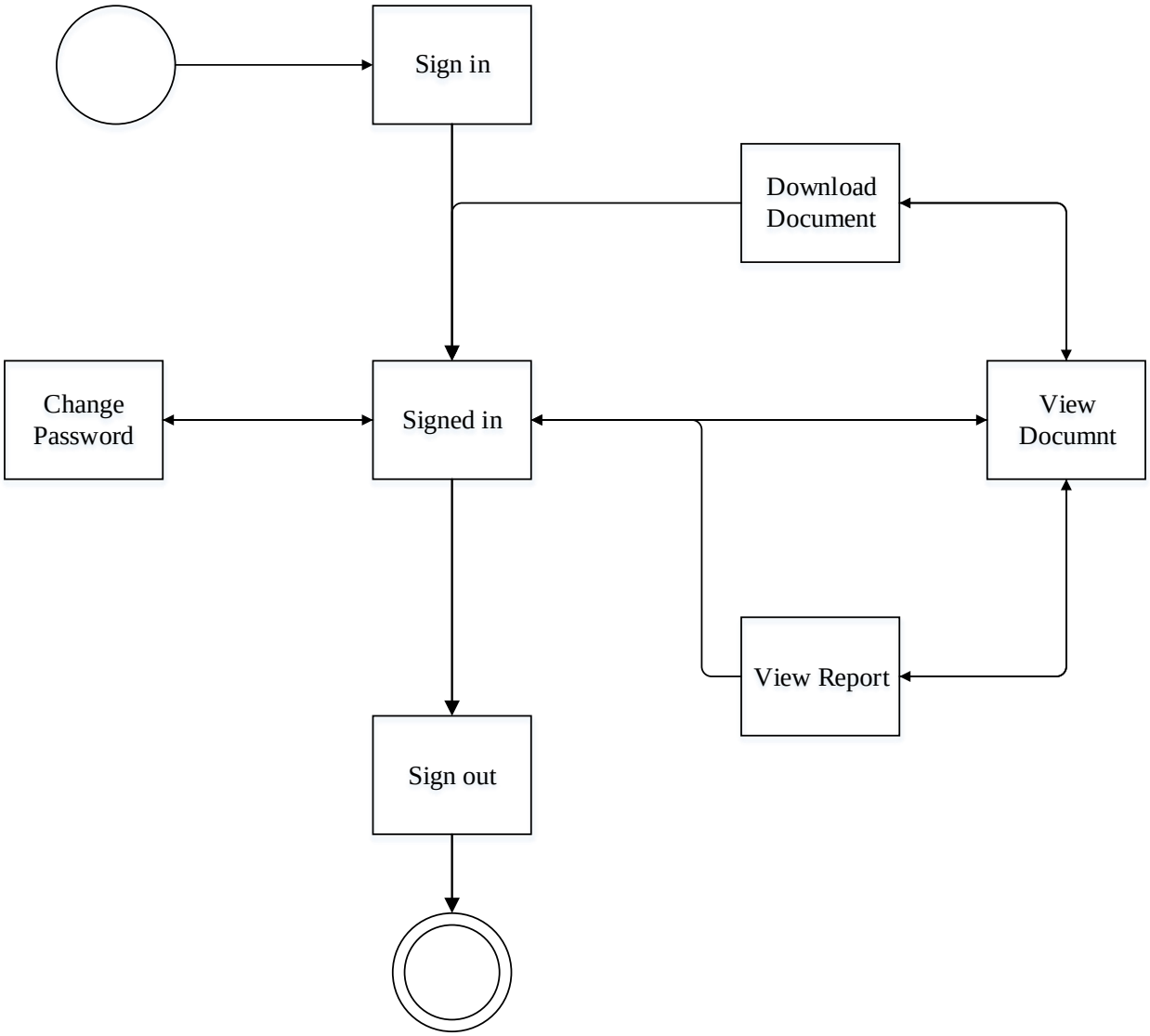


Fig 8.2: State Transition Diagram-Teacher

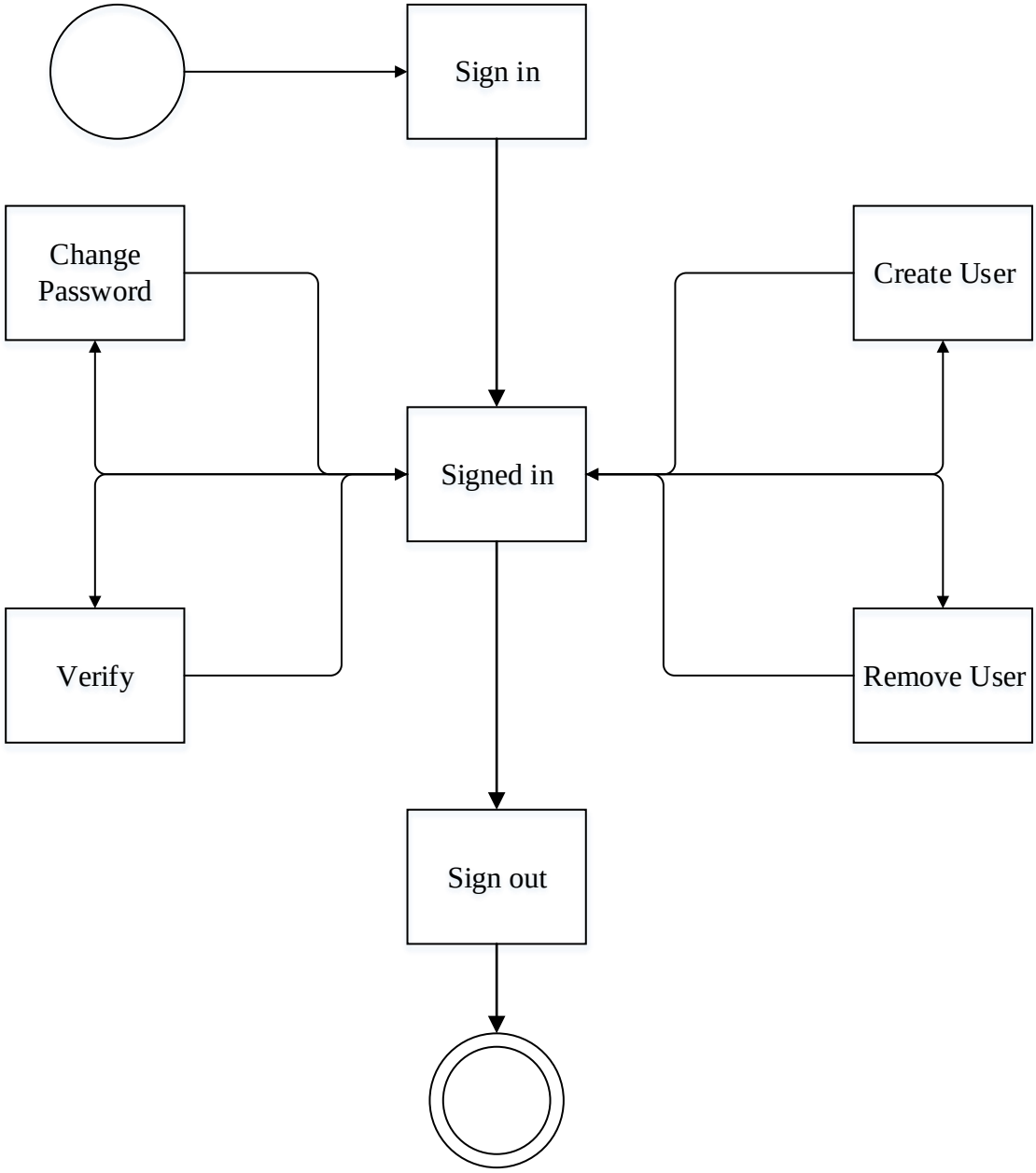


Fig 8.3: State Transition Diagram-Admin

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.

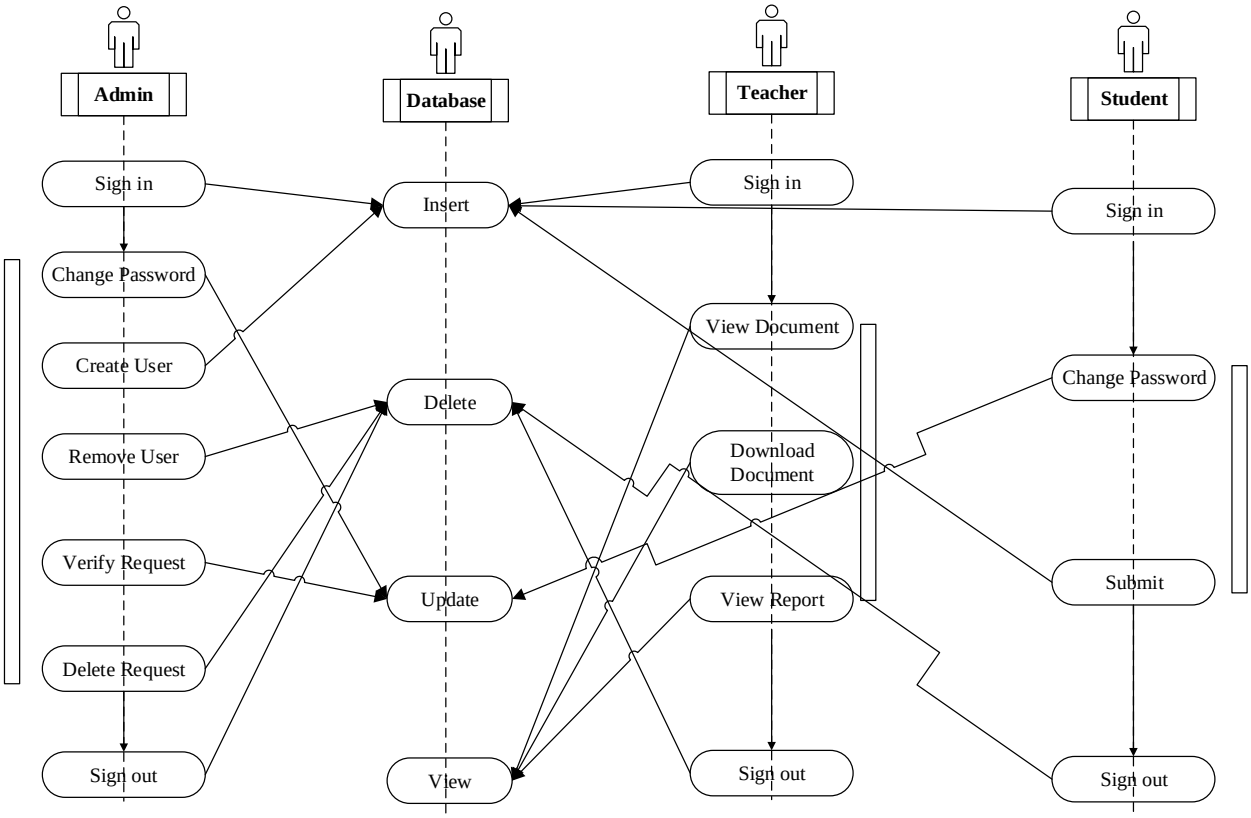


Fig 8.4 : Sequence Diagram

Chapter 9: Conclusion

Plagiarism Detection System is a much needed system to stop unfair activities from student. From this document, the readers will get a clear and easy view of Plagiarism Detection System. 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. I tried my best to remove all dependencies and make effective and fully designed SRS. I believe that reader will find it in order.

References

- Books
  - Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)
- URLs
  - http://www.wikihow.com/Write-Software-Documentation (Last Accessed: May 1, 2016)
  - https://en.wikipedia.org/wiki/Activity_diagram (Last Accessed: May 1, 2016)
  - https://en.wikipedia.org/wiki/Use_Case_Diagram (Last Accessed: May 1, 2016)
  - https://en.wikipedia.org/wiki/State_diagram (Last Accessed: May 1, 2016)