SRENIKOKKHO-WEB BASED LARNING MANAGEMENT SYSTEM

SRS REPORT

Submitted to

Dr.Shafiul Alam
Assistant Professor
Amit Seal Ami
Lecturer
Institute of Information Technology
University of Dhaka
Saeed Siddik
Lecturer
Institute of Information Technology
University of Dhaka

Supervised by

Dr. B. M. Mainul Hossain
Associate Professor
Institute of Information Technology
University of Dhaka

Submitted by

Team Name:
Mystic

Team Members:

Tahlil-803

Saara Sheneen-833

BSSE Session: 2017-2018

Institute of Information Technology
University of Dhaka

20.03.2018

LETTER OF TRANSMITTAL

20 March, 2018

Dr. B. M. Mainul Hossain

Assistant Professor

Institute of Information Technology

University of Dhaka

Subject: Submission of software requirements specifications of Software Project Lab 2.

Dear Sir,

With due respect, we are pleased to submit the final report on software requirements specifications of Learning Management based (LMS) Web Application. Although this report may have shortcomings we did try our level best to produce an acceptable software requirements specifications.

We would be highly obliged if you overlooked our mistakes and accepted our effort we put in this SRS.

Sincerely yours,

Tahlil (BSSE 0803)

Saara Sheneen (BSSE 0833)

BSSE 8th batch

Institute of Information Technology

University of Dhaka

ACKNOWLEDGMENT

At first, we would like to thank the almighty for helping us complete this project. We would like to express our deepest gratitude to all those who provided us the support and encouragement to complete this project. Thanks to our Supervisor, Dr. B.M.Mainul Hossain Assistant Professor, Institute of Information Technology, University of Dhaka, whose continuous suggestions and guidance has been invaluable to us. Also special thanks to Dr. Shafiul Alam, Amit Seal Ami and Saeed Siddik for their guidance and direction. Without such stimulating suggestions, the project could not have made this much progress.

We are grateful to the Institute of Information Technology for giving us the opportunity to do such a project.

Lastly, we would like to thank our classmates. Although they were not on our team, they have always been helpful and provided valuable insights from time to time.

ABSTRACT

This document contains the software requirements and specifications for the Software Project Lab 2. It contains a scenario-based model, data-based model, class-based model and behavioral model. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures, design for database and working sequence of our project.

Our project is about web-based learning management system. Here we will discuss how we will identify the requirements, how to analyze them and how to present a recommended solution for the system. This will help to make the software according to the demand of the stakeholders.

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION OF ‘SRENIKOKKHO’- A LEARNING MANAGEMENT BASED WEB APPLICATION

1.1 Purpose

This document briefly describes the software requirement analysis of a learning management base web application. It contains normal, expected and exciting requirements and establishes a requirement baseline for the development of the system. The requirements contained in this SRS are independent, uniquely numbered and organized. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. This SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Modeling Approach

For preparing the SRS at first a quality function deployment (QFD) was prepared by discussing with the stakeholders. Then a complete scenario was prepared in textual form stating each of the requirements in details. From this text based scenario, four graphical models have been created: scenario based model, data model, class based model and behavioral model. Each of these models will provide valuable insights on how the current system works and how a software for it can be developed.

1.3 Intended Audience

The targeted audience and the purpose of this document will serve to them is discussed below:

I. The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer.

II. 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.

III. 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 the customer's needs.

IV. 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.

V. The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS.

If the final software product is tested on the basis of this SRS document then the purpose of this document will have been fulfilled.

CHAPTER 2: INCEPTION OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

Inception is the beginning phase of requirements engineering. It defines how this web application gets started and what the scope and nature of the problem to be solved are. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of this application. At project inception, we establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the developer team. To establish the groundwork we have worked with the following factors related to the inception phases:

1. List of stakeholders
2. Recognizing multiple viewpoints
3. Working towards collaboration
4. Requirements questionnaire

2.1 List of 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. At inception, a list of people who will contribute input as requirements is elicited. To identify the stakeholders we consulted with the manager of this web application and asked him following questions:

1. Who is paying for ‘Srenikokkho’?
2. Who will be using outcomes of ‘Srenikokkho’?
3. Who gets to make the decisions about ‘Srenikokkho’?
4. Who have resources we need to get ‘Srenikokkho’ done
5. Whose work will affect our project?

We identified the following stakeholders for our project. They are:

1. Instructor
2. Student
3. Designer
4. Developer

I. Programmer

ii. Tester

1. **Instructor:** A user can login either as instructor or Student. An instructor can add course, upload, download, share file/folders. She can also upload assignments, statements, and add comments.

2. **Student:** A student can join course, upload, download, share file/folders. She can also submit assignments, and make/reply comments.

3. **Designer:** They will communicate with all the stakeholders and design according to the demands to provide the documents of the software.

4. **Developers:** Developers are one of the stakeholders because they are also affected by this system. They develop this system and work for further development. If there occurs any system interruption, they will find the problem and try to solve it

i. **Programmer:** They will code according to the supplied document

ii. **Tester:** They will test the software after programmers deliver the software.

2.2 Recognizing Multiple Viewpoints

Different stakeholders achieve different benefits from the system. Consequently, each of them has a different view of the system. So we have to recognize the requirements from multiple points of view, as well as multiple views of requirements. Assumptions are given below

The viewpoint of Instructor:

- ❑ Managing file/folders
- ❑ Creating courses
- ❑ Adding students
- ❑ Providing statements and assignments
- ❑ Creating grade sheet
- ❑ Providing attendance
- ❑ Checking plagiarism
- ❑ Taking classes using live board
- ❑ Uploading video
- ❑ Sending notification to parents of absent students
- ❑ Sending course number to parents
- ❑ Live session
- ❑ Easy/immediate access on any site
- ❑ Hearing lectures of instructor sitting into a vehicle
- ❑ Getting notification if any file is shared, a question is asked, a question is answered, a comment is made on his/her post

The viewpoint of Student:

- ❑ Managing file/folders
- ❑ Submitting assignments
- ❑ Sharing file/folders
- ❑ Uploading file/folder into her account
- ❑ Downloading file/folders from course
- ❑ Managing to-do list, calendars
- ❑ Providing comments
- ❑ Asking questions

- ❑ Providing answer to any question
- ❑ Getting notification if any assignment/new statement is uploaded, one day before deadline, mark sheet is uploaded

The viewpoint of Developer:

- ❑ Cost within budget
- ❑ Detail documentation
- ❑ Enough time for development

2.3 Working towards Collaboration

Every stakeholder has their own requirements. There are some common and conflicting requirements of our stakeholder. That's why we followed the following steps to merge these requirements-

- ❑ Find the common and conflicting requirements
- ❑ Categorize them
- ❑ List the requirements based on stakeholder's priority points
- ❑ Make final decision about requirements

Common requirements:

- ❑ Uploading/downloading/sharing/deleting files
- ❑ Providing comments
- ❑ Asking questions
- ❑ Answering questions
- ❑ Directly open files using software instead of downloading

Conflicting requirements:

- ❑ Sending notification to parents if any student is absent
- ❑ Sending grade sheet of a student to her parents
- ❑ Live session
- ❑ Hearing lectures of instructor sitting into a vehicle

Final requirements:

We finalize the following requirements based on stakeholder's priority point:

- ❑ Managing file/folders
- ❑ Creating courses
- ❑ Adding students
- ❑ Providing statements and assignments
- ❑ Creating grade sheet
- ❑ Submitting assignments
- ❑ Sharing file/folders
- ❑ Uploading file/folder into her account
- ❑ Downloading file/folders from course
- ❑ Managing to-do list
- ❑ Providing comments
- ❑ Asking questions
- ❑ Providing answer to any question
- ❑ Getting notification if any assignment/new statement is uploaded,one day before deadline, mark sheet is uploaded
- ❑ Getting notification if any file is shared and a comment is made on any post

2.4 Requirements Questionnaire

We first ask the stakeholder some context-free questions to understand the project's overall performance and goals. These questions are mentioned below:

- ❑ Would you like to have a virtual classroom along with real classroom where you can upload, organize and download all your learning materials?(if yes answer the following two question)
- ❑ Do you think there should be a search option for navigating to a file?
- ❑ Do you think only you would be able to see your student's activities or the other students would also be able to see?
- ❑ Do you think you would upload your assignments on the web platform?
- ❑ Do you think you would upload your syllabus or course outline and set it on a separate section?

- ❏ Do you think any student should be able to upload any file or other resources?
- ❏ Do you think any student should be able to ask a question in the platform?
- ❏ Do you think anyone should be able to comment on any post?
- ❏ Do you think it's necessary for a notification in the following cases?
- ❏ Please mention your other possible feature(s) which you think will be useful for you and make your work easy if you use this application.(Your answer must be descriptive in this section)

These questions help us to identify the stakeholders of the project. Then we ask our next set of questions to better understand the problem and take stakeholder's opinion about the solution. The final set of question focused on the effectiveness of the communication activity itself. The Inception phase helped us to establish a basic understanding of the web application, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders. In our project, we have established a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. 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 OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

3.1 Quality Function Deployment (QFD)

QFD is the methods or quality management technique that translate the needs of the customer into technical requirements for the software. Ultimately the goal of QFD is to translate often subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a method for maximizing customer satisfaction from the software engineering process. We followed this methodology to identify the requirements for the project which are given below:

3.1.1 NORMAL REQUIREMENTS

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

- A user can upload, download, search and share files/folders.
- An instructor can create course.
- A student can join course.
- Users can maintain to-do.
- An instructor can upload assignments with a deadline and statements.
- A student can submit assignment within deadline.
- A student can delete submitted assignment before deadline if s/he needs to change anything.
- A user can open files with pdf/word/ppt. extension directly using this software.
- If an instructor wants to extend deadline of assignments s/he can do it.
- An instructor can import files/folder to course from her account.

- A user can provide comment in course page.
- A student will get notifications on following circumstances.
    - Assignment upload
    - New Statement
    - One day before deadline
- An instructor will get notification after deadline of any assignment has passed.
- A user will get notification if any file is shared.

3.1.2 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.

- A user must sign in before entering her account for security purpose
- Users are allowed to add their personal information.
- A user can choose any theme for her account.
- A gorgeous user interface should be provided.

3.1.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. Following are some exciting requirements of our project.

- A guideline option will be there for the users so that they can learn about all features.

3.2 Usage Scenario of Srenikokkho

1. Homepage

The homepage will contain a short list of the features. From this page, a user can go to sign up or login page. There is also a ‘learn more’ option through which the user can go to a page where he can get detailed information about all the available functionalities.

2. Authentication Module

In our web application, a new user must open an account first. The user will get to sign up and/or login page through the home page. There will be two types of accounts. One is a ‘Student’ account and the other one is an ‘Instructor’ account. For sign up or registration, user must provide following information

1. Email
2. Password
3. Name of Institution
4. Account type

Password must be at least 8 characters long and can have alphanumeric characters. E-mail id must be a valid one and only one account can be created. None of the fields can remain empty.

After providing these information, the user will get a confirmation code via his/her email. S/he will have to use that code for completing the signup. For log in users must provide email and password. For a wrong password, the user can try for three times. If the third attempt is wrong, then the login will be disabled for 30 minutes. If she forgets her password she will be given chance to set a new password. For setting a new password, she will have to enter her email for verification. A message will be sent to her email for confirmation. If verification is successful, she will be able to set a new password.

The user is redirected to a different page both for signup and login. After logging in, the user may log out of her account. Then, her repository (a place where or receptacle in which things are or may be stored) would be saved and the user would be redirected to the homepage.

3. User Module

The user can log in either as instructor mode, or student mode and go to the instructor specific page or student-specific page respectively. Common and individual features that can be enjoyed by an instructor/student are given below

Instructor

The instructor may add a course and has the ability to export any file/folder to any of her created courses.

Student

A user having ‘student’ account may join courses created by an instructor account using a verification code (see the course module).

Common

Both users have some common functionalities. A user can manage her own files in custom folders in her repository. She can upload files of all formats in the repository. A user has two default folders:

1. to get access to the files, the user had recently accessed
2. the files that are shared with.

These files are called “Recent” folder and “Shared” folder. Files/folders can be identified uniquely by name. The user can search for a file and also download a file. She can delete any file. She also can share a file with anyone with a ‘Srenikokkho’ account’; she would just have to specify the email of the person having a ‘Srenikokkho’ account’. A user can open only three types of file in the web- pdf, word and ppt. file. Both users have to do .A user can add multiple to-do’s, where she specify work name optional description and a checkbox for done and undone. She may remove it.

4. Course Module

Common and individual features of this module that can be enjoyed by an instructor/student are given below

Instructor

When the instructor adds a course, she will be redirected to a ‘create course’ page. Then the instructor fills the following information regarding the course:

Course title

Subject

A course is then created and a verification code is generated and given to the instructor's email. The instructor can upload an assignment or statement along with a description optionally with a file attachment. The attached file can be imported either from the local device or a given repository to that specific course. The instructor can remove or update any statement or assignment before a deadline. She also is able to extend the deadline for any assignment at any time.

Student

Students can submit an assignment by uploading assignment file. A student can remove her submitted assignment before the deadline.

Common

Both types of users can download any type of file and only word, pdf or ppt. file can be opened directly. If any user wants to open files having other types of extensions, she will have to download that file and can open that using existing applications on her device. A user can also search for a file. A user can provide comment.

5. Notification Module

The user gets the notification(s) (if there is any) when she logs in. A user gets a notification of the below circumstances:

- If any file is shared

A student gets a notification in the following cases:

- Assignment upload
- New Statement
- One day before deadline

An instructor gets a notification in the following case:

- After a deadline have passed

Notification consists of a title, id and a short description.

CHAPTER 4: SCENARIO-BASED MODEL OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

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 able to characterize requirements and build meaningful analysis and design models properly. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.1 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.

4.2 Use Case Diagrams

Use Case diagrams give the non-technical view of the overall system.

4.2.1 LEVEL-0 USE CASE DIAGRAM-SRENIKOKKHO

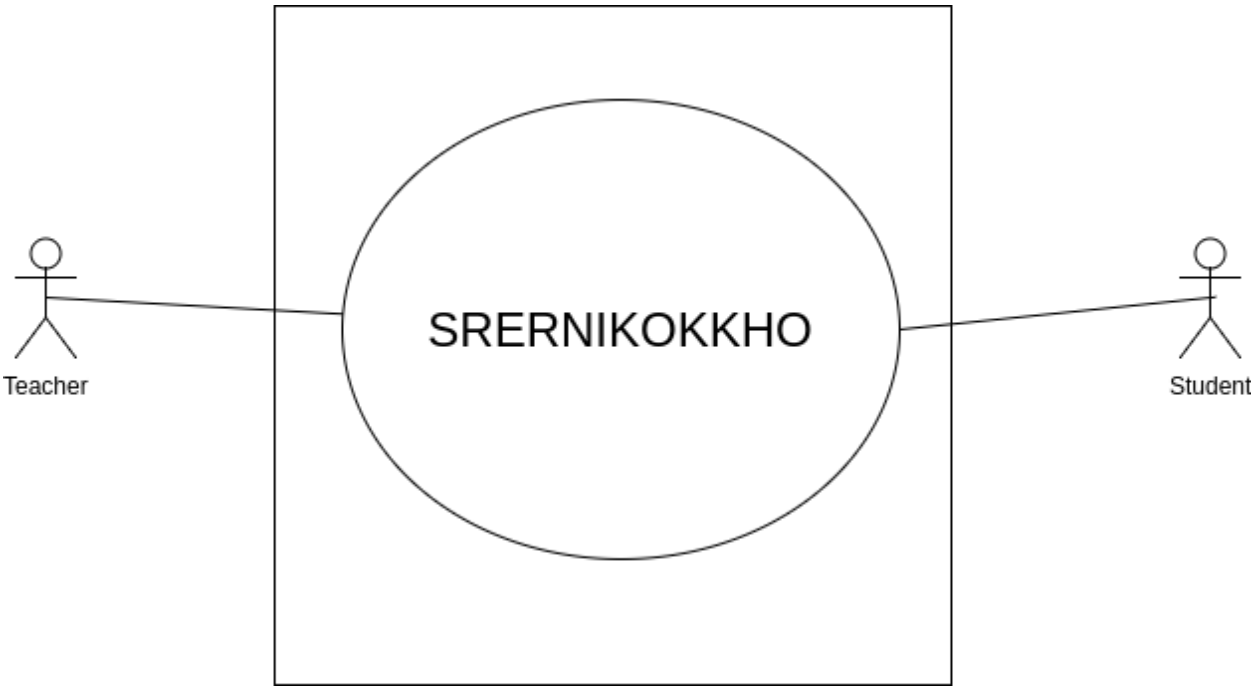

Figure 1: Level 0 Use Case diagram of SRENIKOKKHO

Description of Use Case Diagram Level 0:

After analyzing the user story we found two actors who will directly use the system as a system operator. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Following are the actors of SRENIKOKKHO –

1. Instructor-primary actor
2. Student-primary actor

4.2.2 LEVEL-1 USE CASE DIAGRAM-SUBSYSTEMS OF SRENIKOKKHO

In figure 2, the level 1 diagram of sub-systems of Srenikokkho is shown-

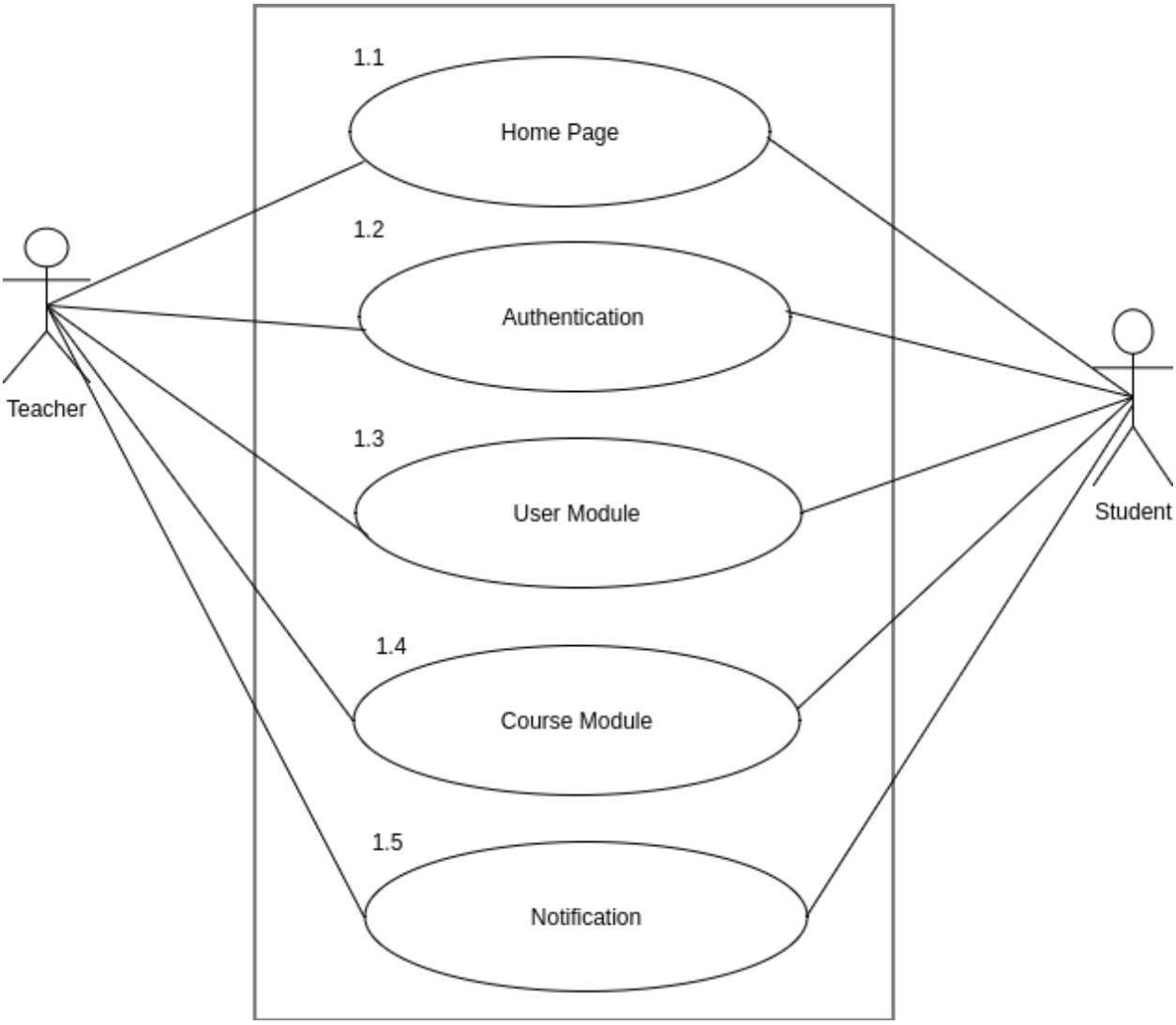

Figure 2: Level 1 Use Case Diagram-Subsystems of SRENIKOKKHO

Description of Use Case Diagram Level 1:

There are five subsystems in the SRENIKOKKHO. They are:

1. Homepage
2. Authentication
3. User module
4. Course module

5. Notification

The homepage will contain a short list of the features. From this page, a user can go to sign up or login page. There is also a ‘learn more’ option through which the user can go to a page where he can get detailed information about all the available functionalities.

The user gets the notification(s) (if there is any) when she logs in. A user gets a notification of the below circumstances:

- If any file is shared

A student gets a notification in the following cases:

- Assignment upload
- New Statement
- One day before deadline

An instructor gets a notification in the following case:

- After a deadline have passed

Notification consists of a title, id, date and a short description.

Others module will be discussed later in this document.

Action reply:

Action: A user will share files/folders with any other user.

Reply: System will send notification.

Action: An instructor will set a deadline for assignment

Reply: System will store that and send notification at proper time

Action: Instructor upload assignment and statement and submission date

Reply: System will store it and send notification.

4.2.3 LEVEL-1.2 USE CASE DIAGRAM-AUTHENTICATION OF SRENIKOKKHO

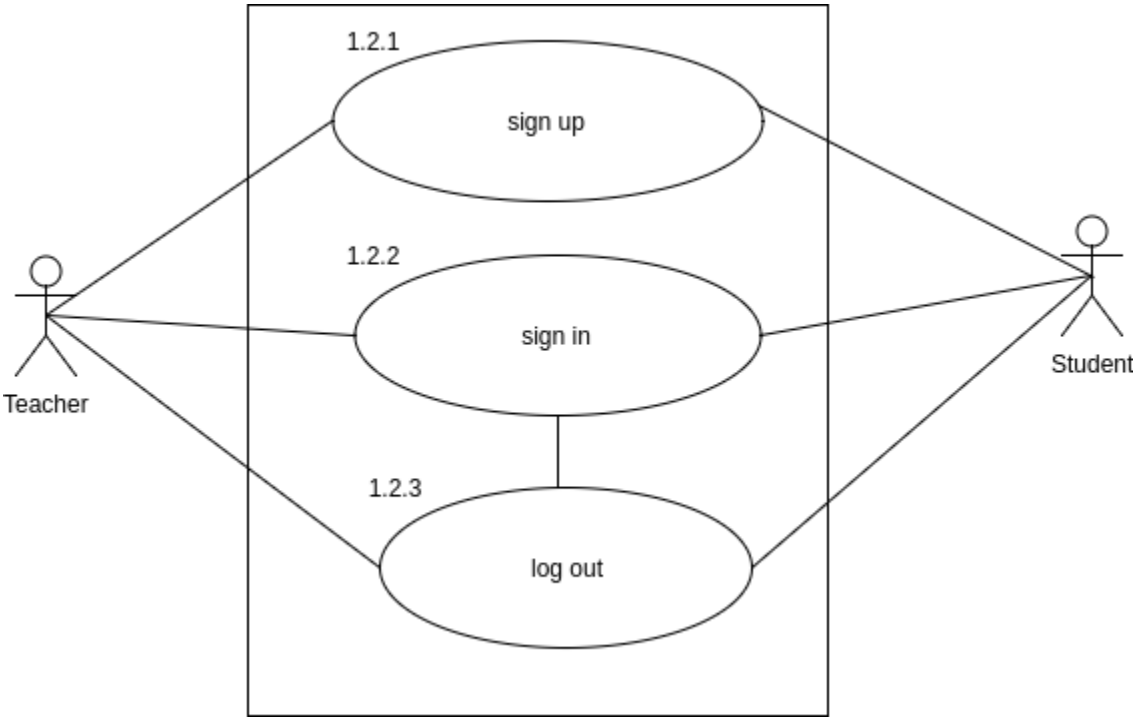

Figure 3: Level 1.2 Use Case Diagram-Authentication of SRENIKOKKHO

Description of Use Case Diagram of Authentication subsystem

This subsystem is divided into three part:

1. Register/Sign Up
2. Sign in/Log in
3. Log out

In our web application, a new user must open an account first. The user will get to sign up and/or login page through the home page. There will be two types of accounts. One is a ‘Student’ account and the other one is an ‘Instructor’ account. After providing all information for registration, the user will get a confirmation code via his/her email. S/he will have to use that code for completing the signup. For log in users must provide email and password. For a wrong password, the user can try for three times. If the third attempt is wrong, then the login will be disabled for 3 minutes. If she forgets her password she will be given chance to set a new password. For setting a new password, she will have to

enter her email for verification. A message will be sent to her email for confirmation. If verification is successful, she will be able to set a new password.
The user is redirected to a different page both for signup and login. After logging in, the user may log out of her account.

Action reply:
Action: A new user must provide information for sign up
Reply: System will store all information
Action: A user must provide email and password
Reply: System will check validity and act accordingly
Action: A user provide wrong password for the third time
Reply: System will be locked for 3 minutes
Action: If a user forgets her password
Reply: System will take information for generating a code and send that code for password recovery
Action: A user logs out
Reply: System will save her repository and redirected to the homepage

4.2.4 LEVEL-1.3 USE CASE DIAGRAM-USER MODULE OF SRENIKOKKHO

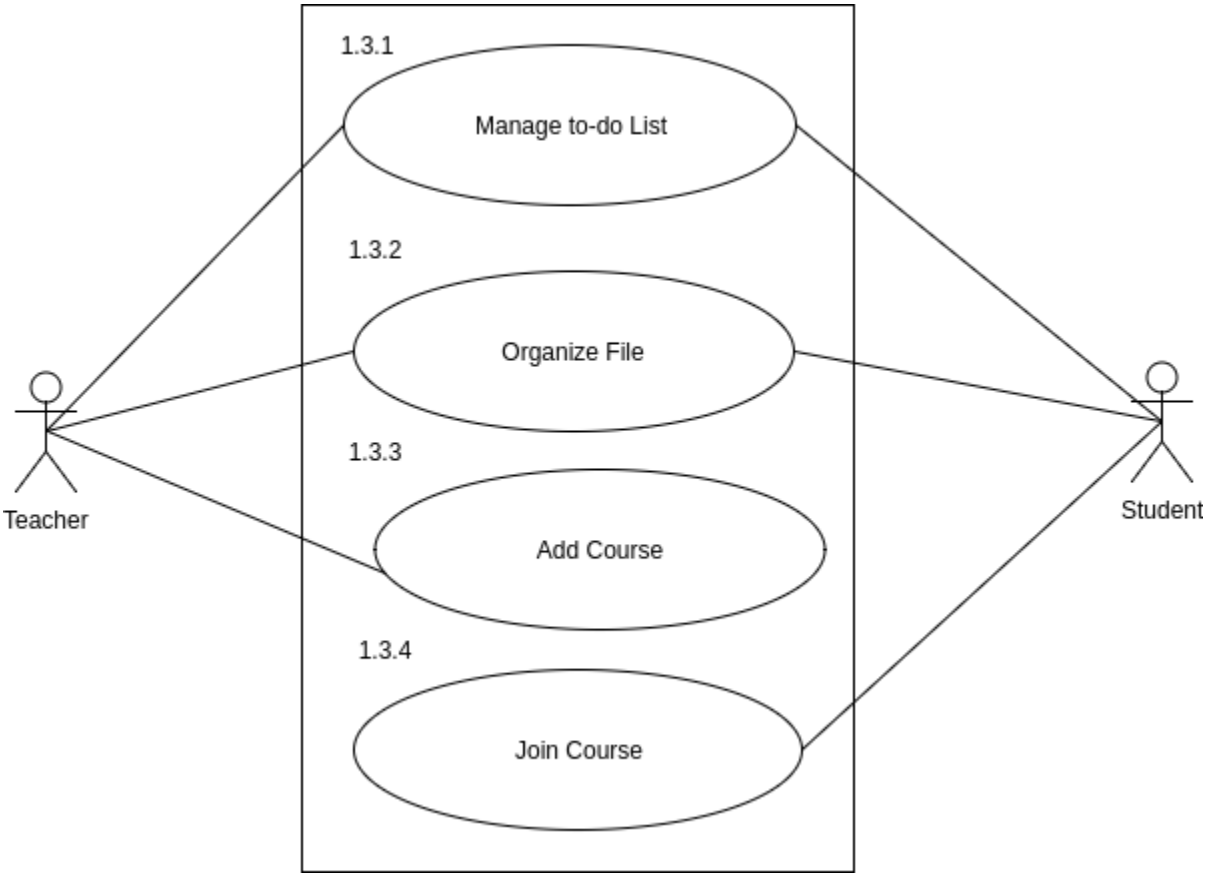

Figure 4: Level 1.3 Use Case Diagram-User Module of SRENIKOKKHO

Description of Use Case Diagram of User Module subsystem

This subsystem is divided into four parts:

1. Manage to-to
2. Organize file
3. Add course
4. Join course

A user can manage to-do and organize file. But an instructor account only can create course and a student account can join course only.

Action reply

Action: User will select option

Reply: System will act accordingly

4.2.5 LEVEL-1.3.1 USE CASE DIAGRAM-TO-DO OF SRENIKOKKHO

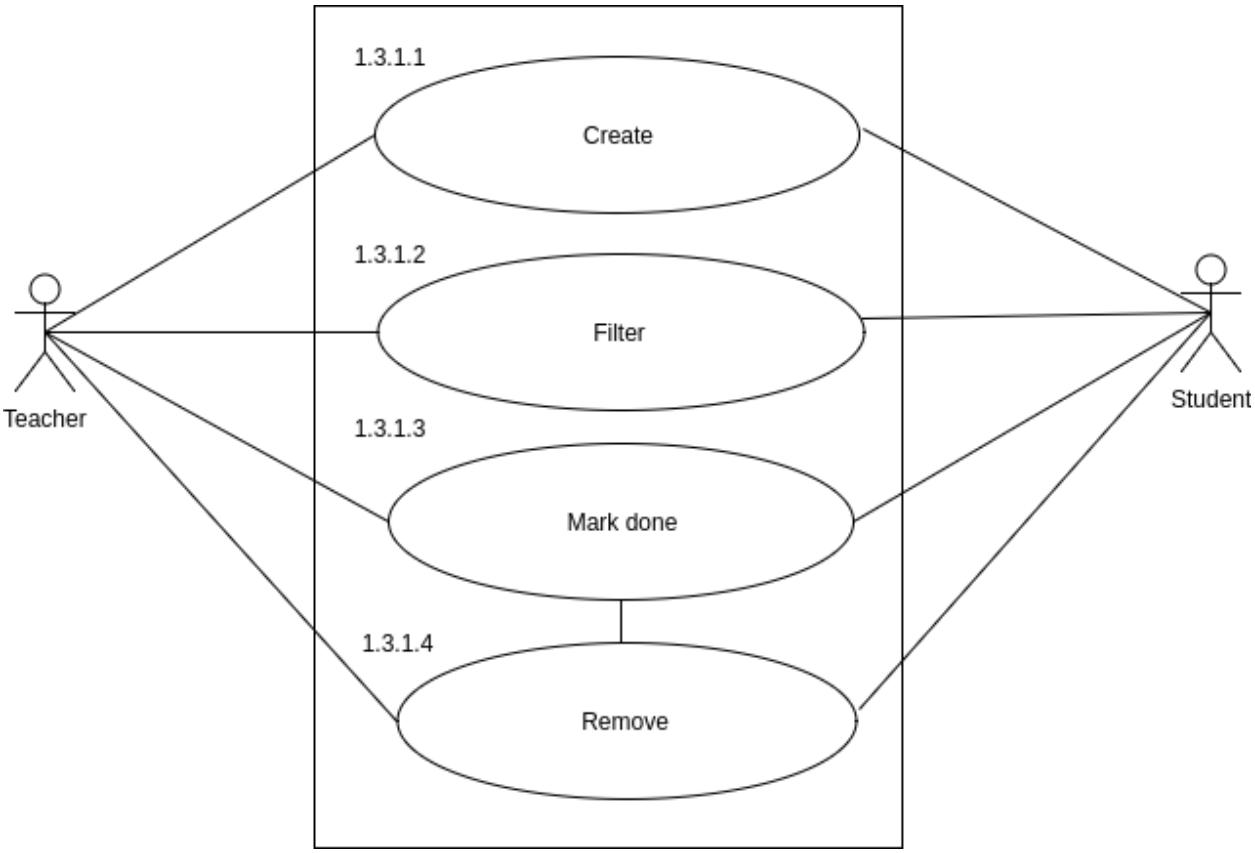

Figure 5: Level 1.3.1 Use Case Diagram-To-Do of SRENIKOKKHO

Description of Use Case Diagram of To-Do subsystem

This part of user module is divided into four parts:

Create

Filter

Mark-done

Remove

A user can add multiple to-dos, where she specify work name optional description and a checkbox for done and undone. She may also remove it.

Action reply
Action: A user will request for creating to-do
Reply: System will create and store
Action: A user will request to filter/mark/remove it
Reply: System will take action according to her choice

4.2.6 LEVEL-1.3.2 USE CASE DIAGRAM-FILE ORGANIZATION OF SRENIKOKKHO

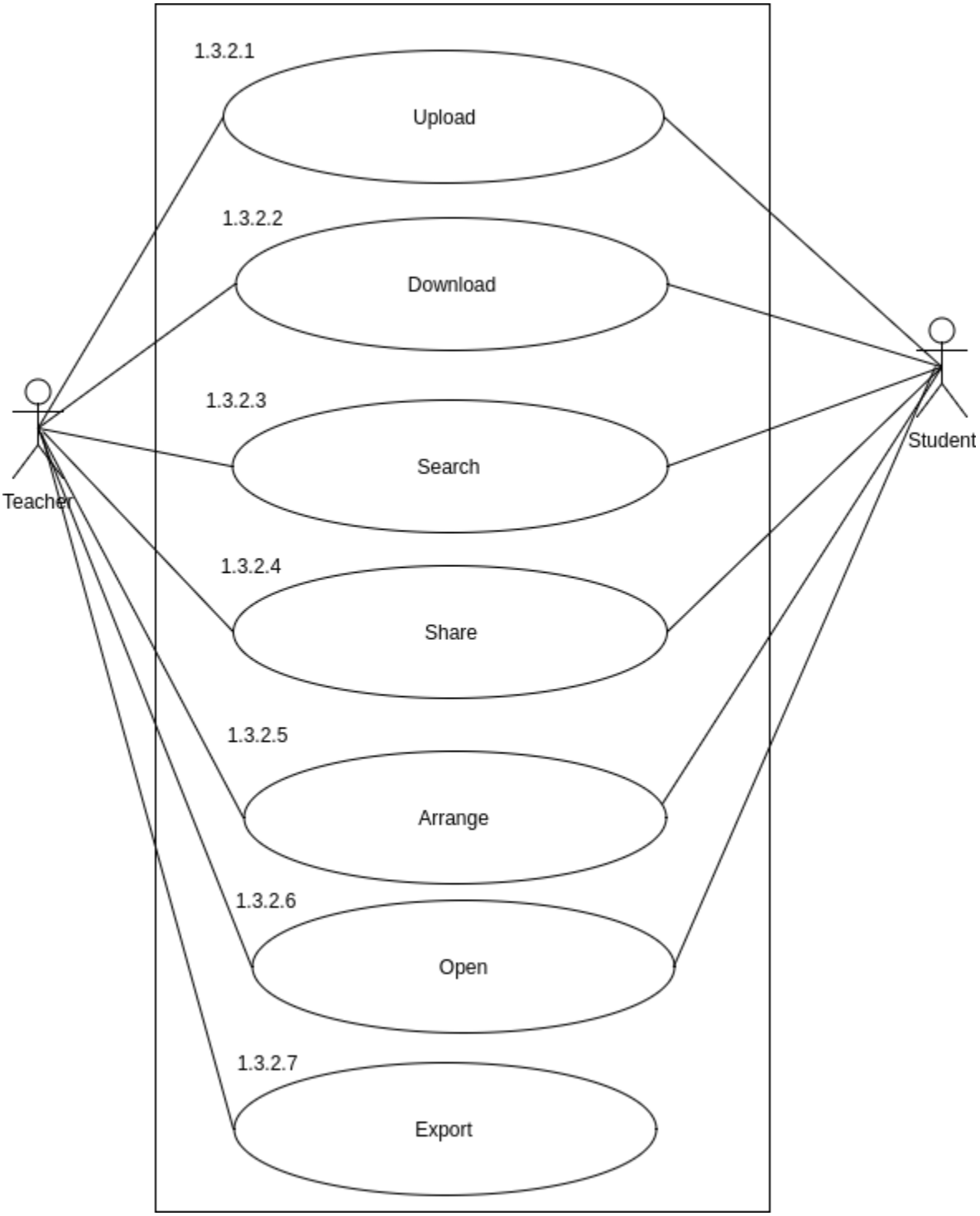

Figure 6: Level 1.3.2 Use Case Diagram-File Organization of SRENIKOKKHO

Description of Use Case Diagram of File Organization subsystem

This part of user module is divided into seven parts:

1. Upload
2. Download
3. Search
4. Share
5. Arrange
6. Open
7. Export

Any user can upload, download, search, share and arrange her files/folders. If any user wants to open a file, s/he can open it directly if the file has word/pdf/ppt. extension. Otherwise s/he needs to download it and can open it. The instructor has the ability to export any file/folder to any of her created courses.

Action reply

Action: A user will upload file/folder from her local device

Reply: System will store

Action: A user will provide a file/folder name

Reply: System will search and show it

Action: A user will request to open file

Reply: System will check the extension and open it

Action: An instructor request to export file into a course

Reply: System will act accordingly

Action: A user will arrange files/folders

Reply: System will store

4.2.7 LEVEL-1.4 USE CASE DIAGRAM-COURSE MODULE OF SRENIKOKKHO

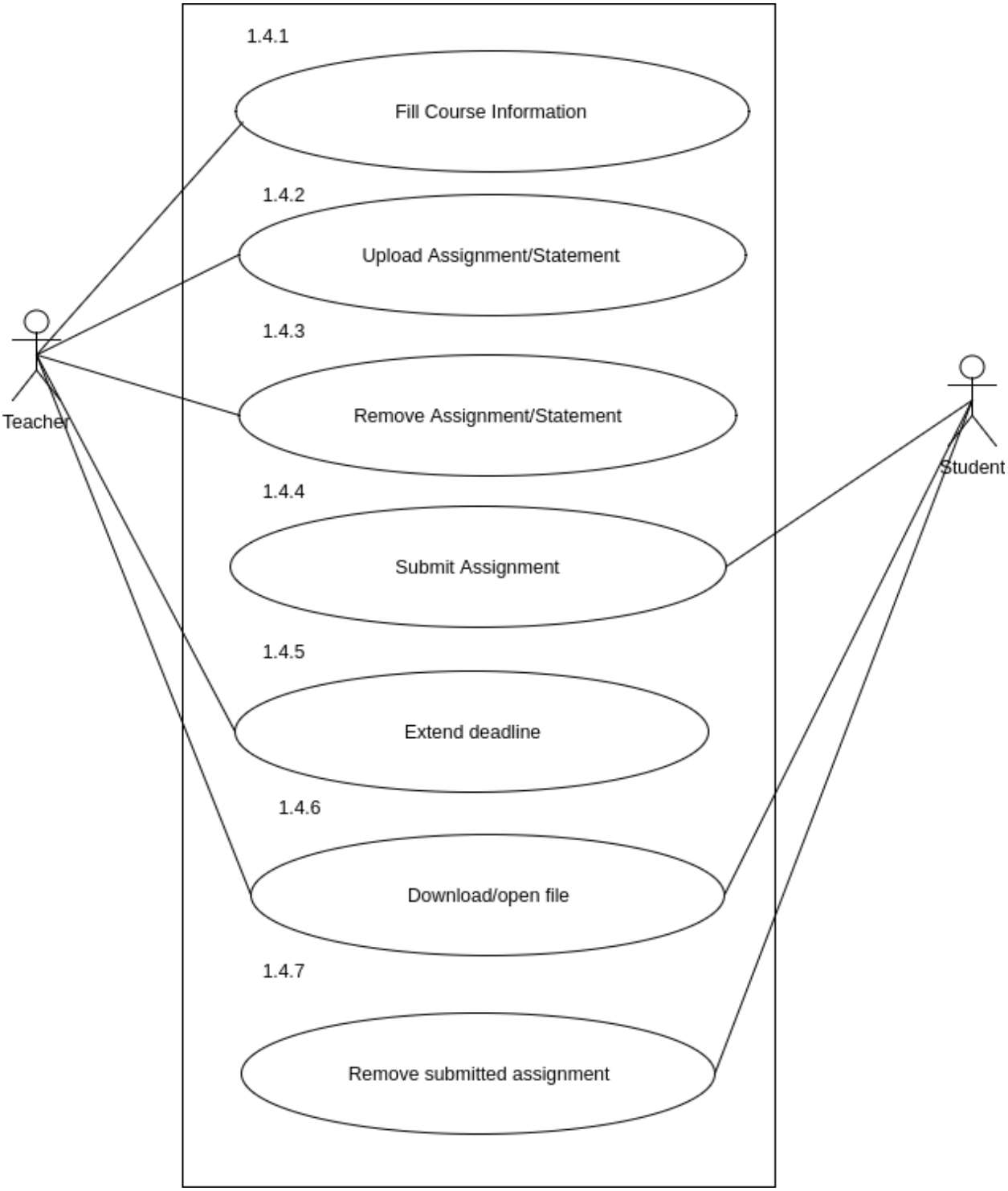

Figure 7: Level 1.4 Use Case Diagram-Course Module of SRENIKOKKHO

Description of Use Case Diagram of Course Module subsystem

This subsystem is divided into seven parts:

1. Fill course information
2. Upload assignment/statement
3. Remove assignments/statements
4. Extend deadline
5. Submit assignment(s)
6. Download/open file
7. Remove submitted assignment(s)

In this module, both types of users can download any type of file and only word, pdf or ppt. file can be opened directly. The instructor can upload an assignment or statement along with a description optionally with a file attachment. The attached file can be imported either from the local device or a given repository to that specific course. The instructor can remove or update any statement or assignment before a deadline. She also is able to extend the deadline for any assignment at any time. Students can submit an assignment by uploading assignment file. A student can remove her submitted assignment before the deadline.

Action reply

Action: A user will request to open file

Reply: System will check the extension and open it

Action: An instructor will provide course information

Reply: System will store it and show it

Action: An instructor will upload statement/assignment(s) with deadline

Reply: System will store all information

Action: An instructor will change or extend the deadline of existing assignment

Reply: System will store that change

Action: A student will submit assignment and can remove it within deadline

Reply: System will check the deadline and let the student submit or remove assignment

4.3 Activity Diagrams of Srenikokkho

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

The activity diagrams of the modules described in the previous chapter is shown in the following figures:

4.3.1 Activity Diagram-Subsystems of Srenikokkho

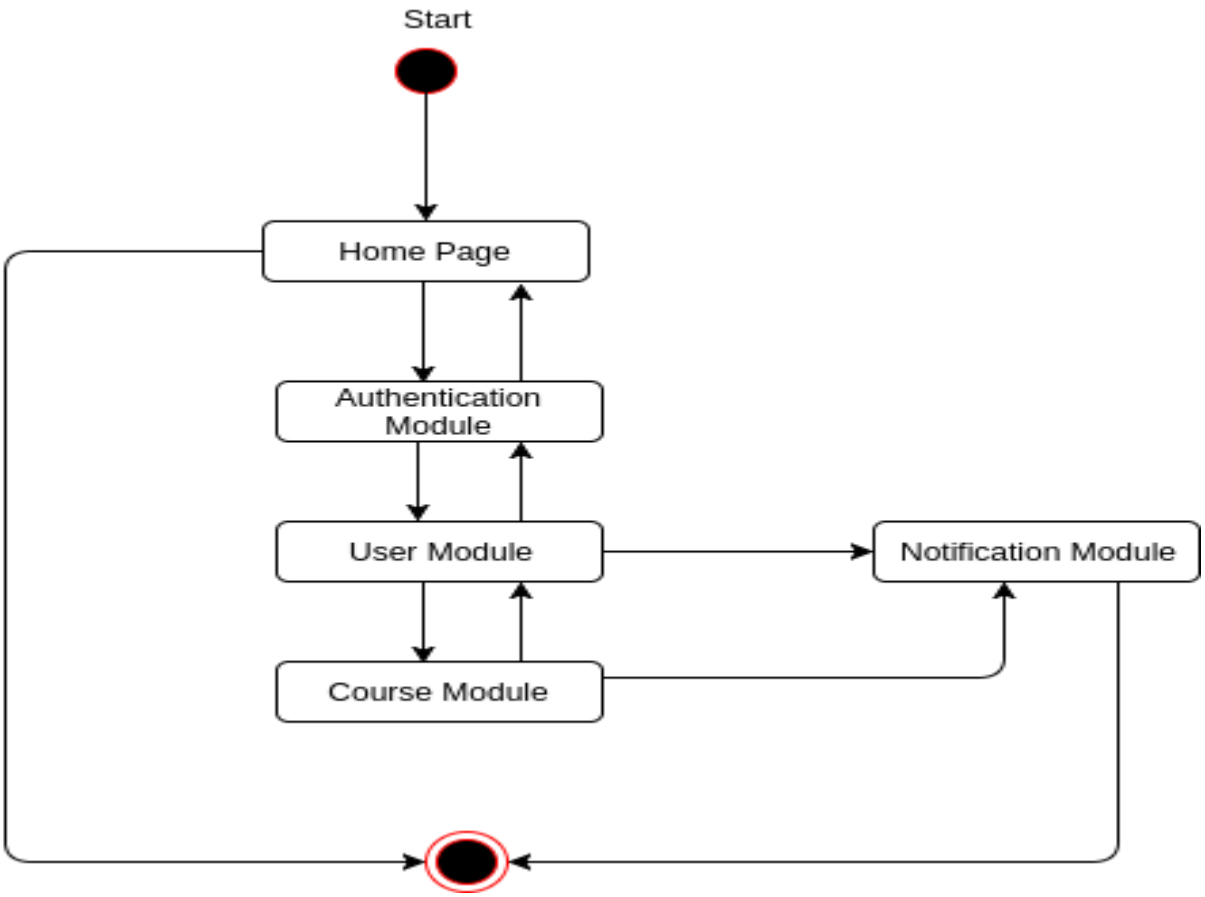

Figure 8: Activity Diagram-Main Subsystems of SRENIKOKKHO

4.3.2 Activity Diagram-Authentication of Srenikokkho

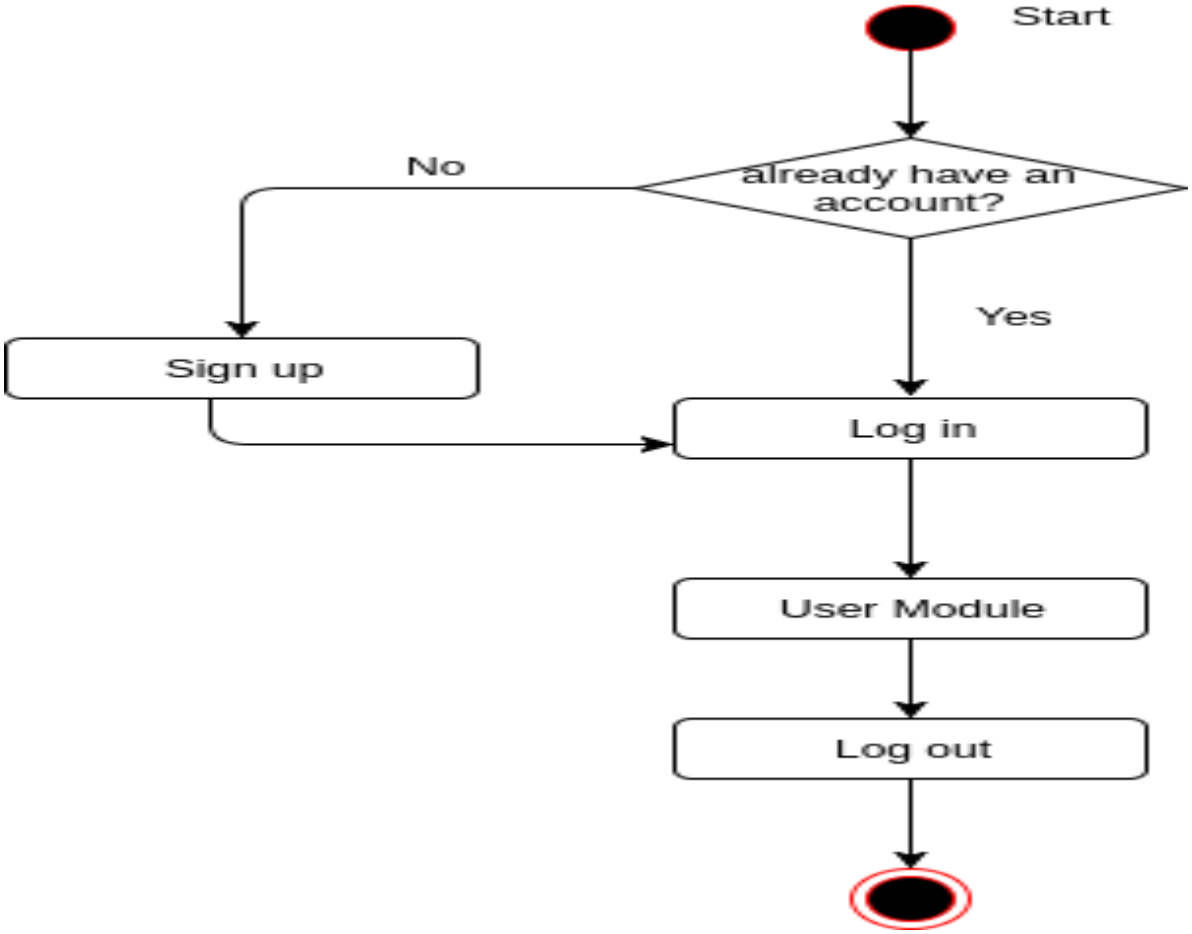

Figure 9: Activity Diagram-Authentication of SRENIKOKKHO

4.3.3 Activity Diagram-Sign Up of Srenikokkho

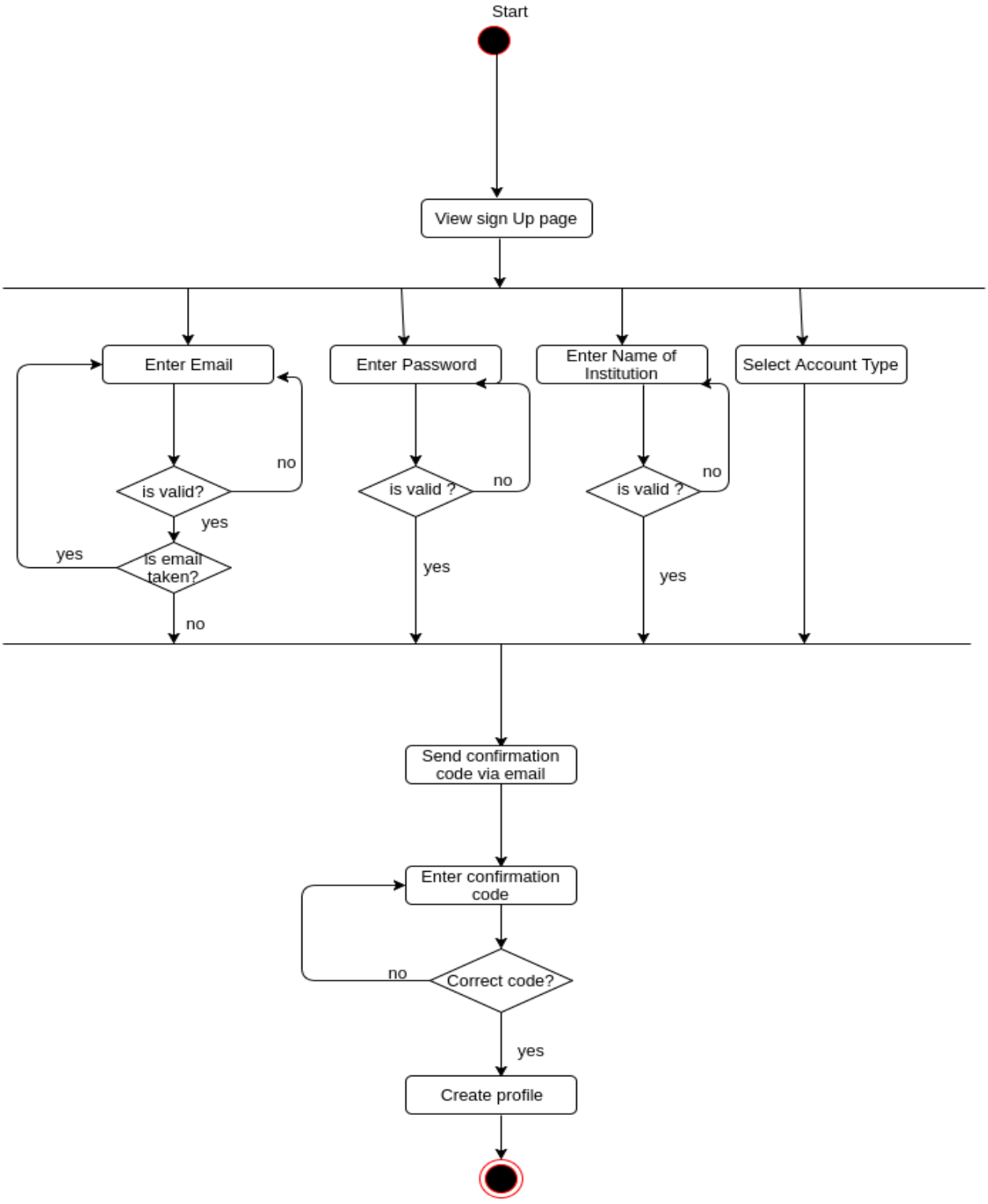

Figure 10: Activity Diagram-Sign Up of SRENIKOKKHO

4.3.4 Activity Diagram-Login of Srenikokkho

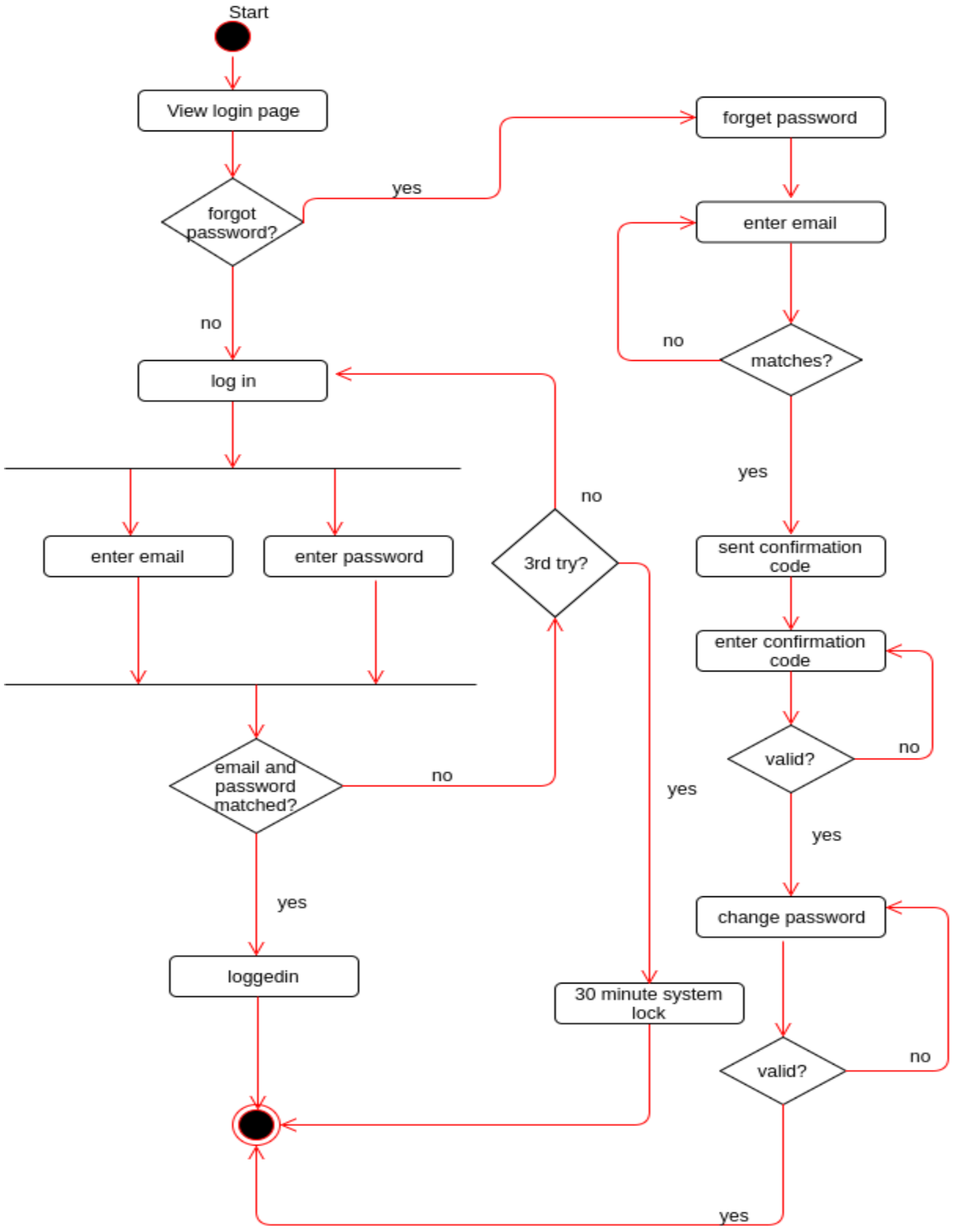

Figure 11: Activity Diagram-Login of SRENIKOKKHO

4.3.5 Activity Diagram-User of Srenikokkho

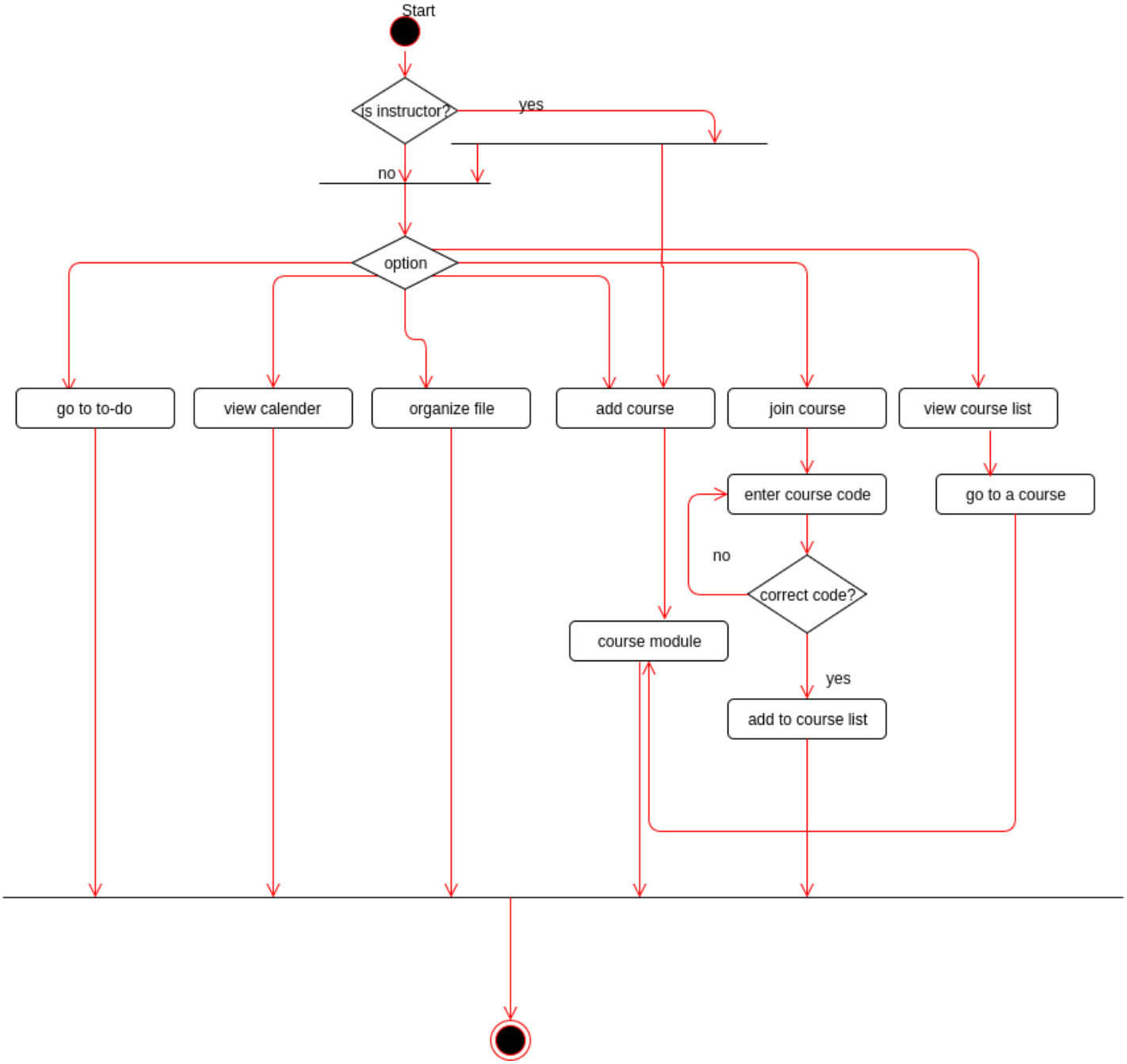

Figure 12: Activity Diagram-User of SRENIKOKKHO

4.3.6 Activity Diagram-To-Do of Srenikokkho

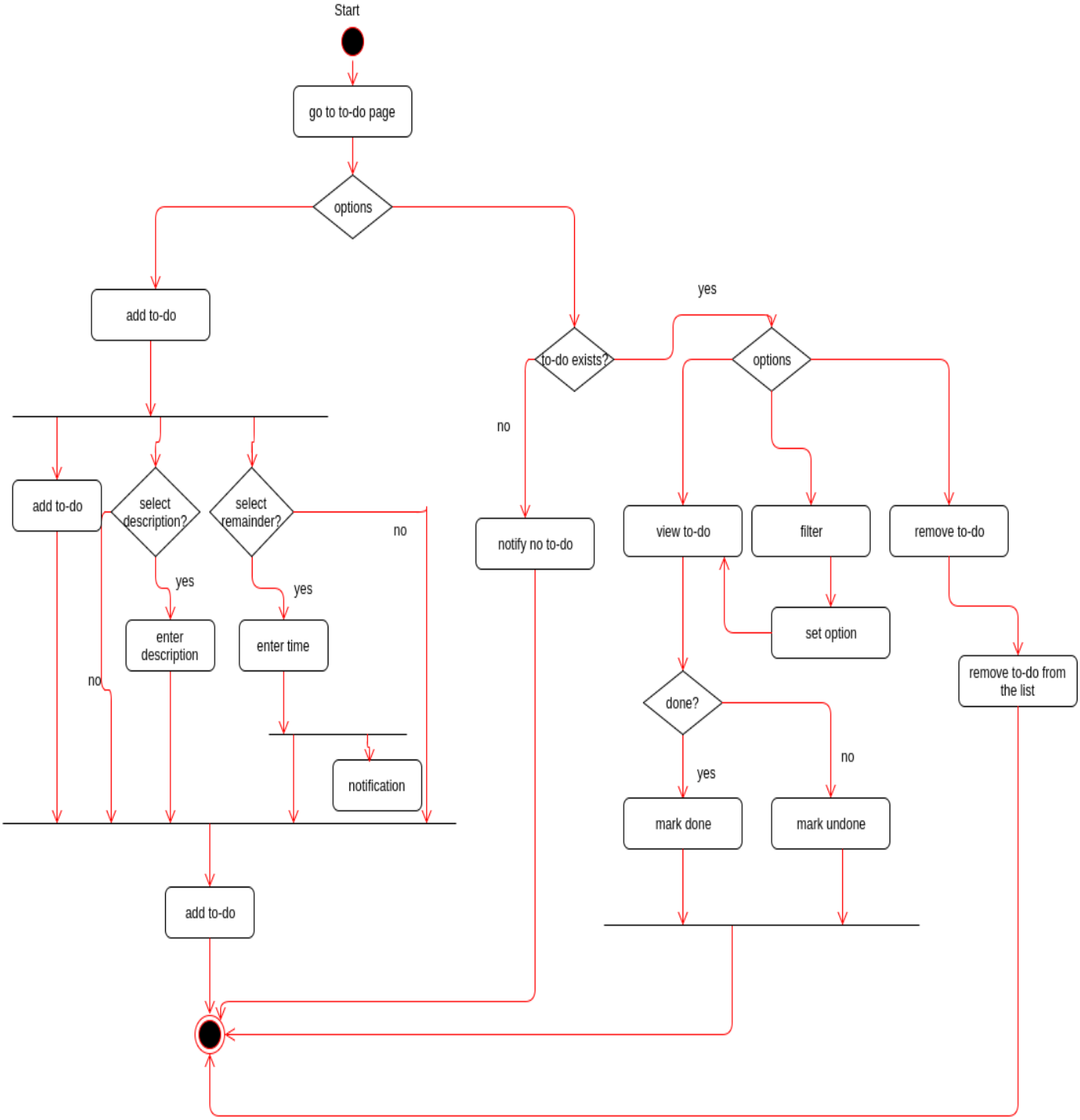

Figure 13: Activity Diagram-To-Do of SRENIKOKKHO

4.3.7 Activity Diagram-Organize File of Srenikokkho

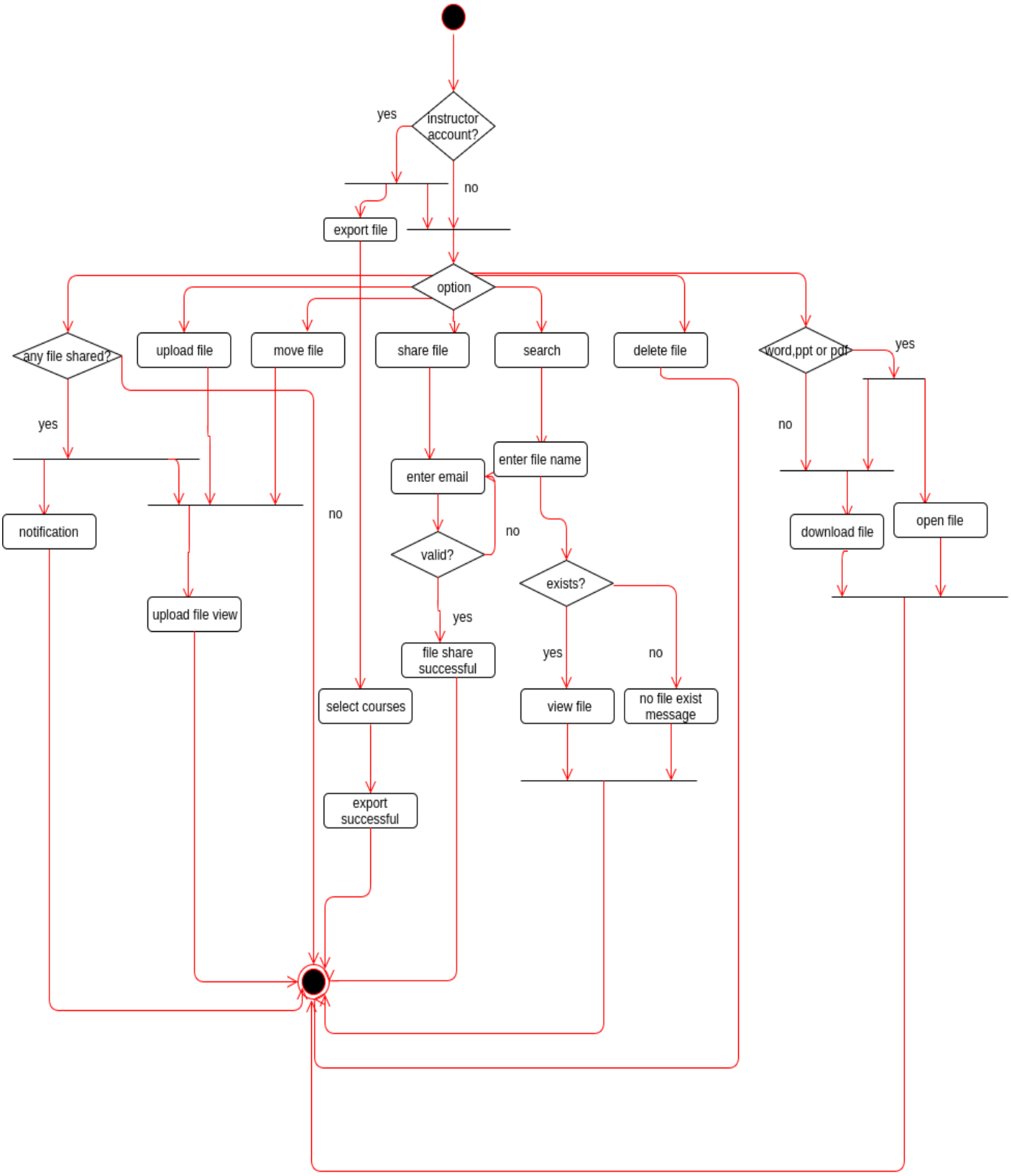

Figure 14: Activity Diagram-Organize File of SRENIKOKKHO

4.3.8 Activity Diagram-Course of Srenikokkho

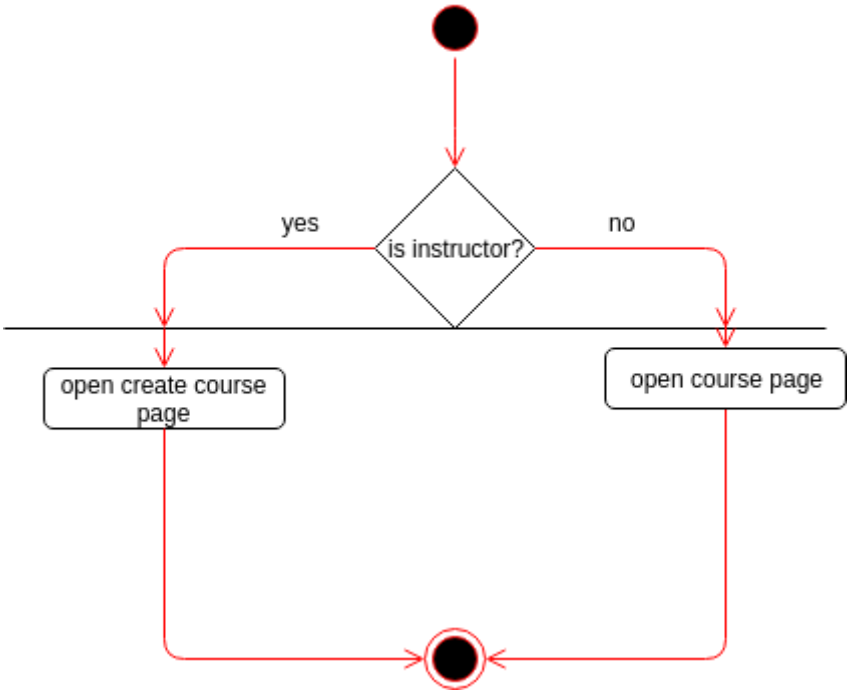

Figure 15: Activity Diagram-Course of SRENIKOKKHO

4.3.9 Activity Diagram-Create Course of Srenikokkho

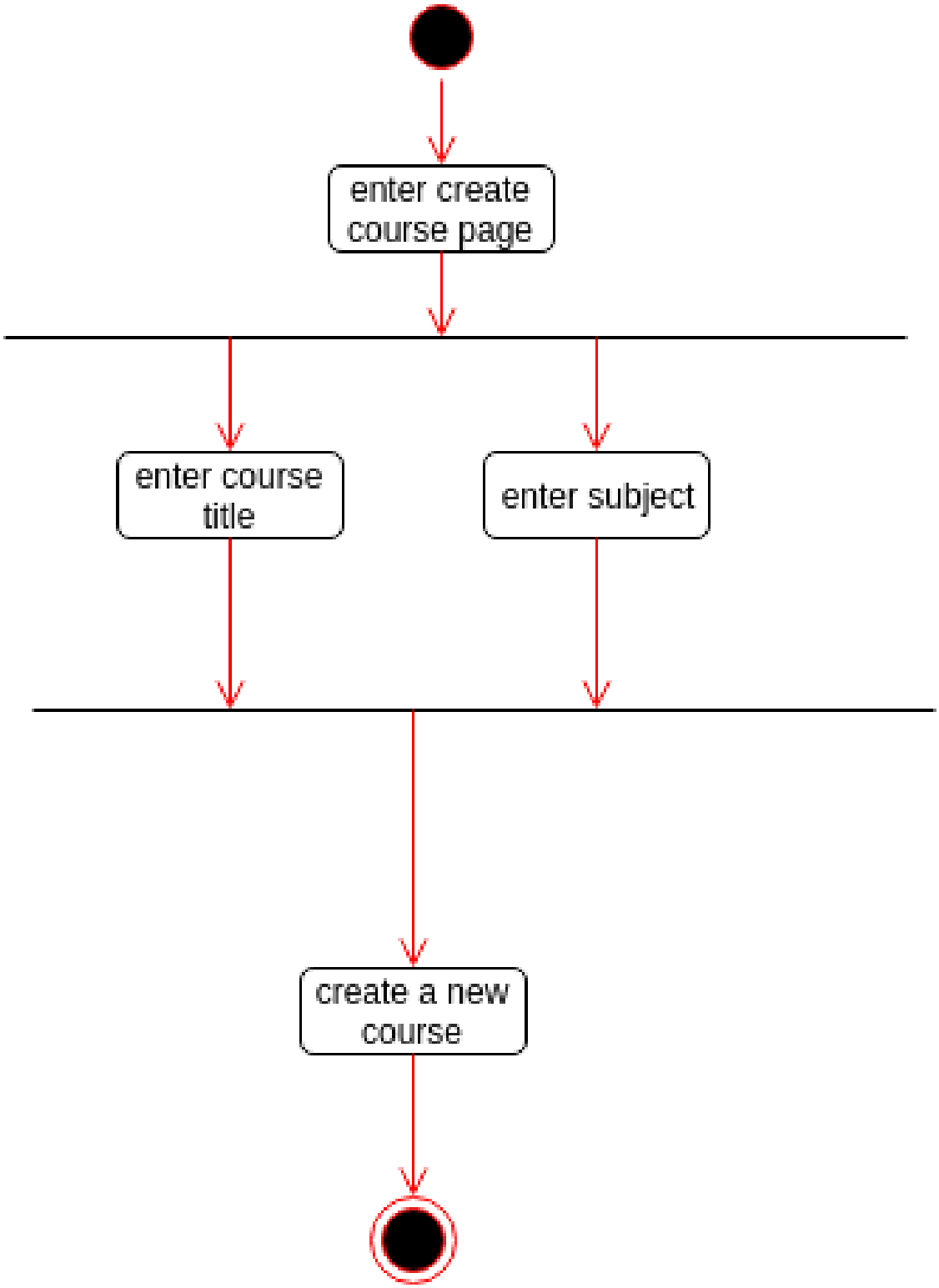

Figure 16: Activity Diagram-Create Course of SRENIKOKKHO

4.3.10 Activity Diagram-Open Course of Srenikokkho

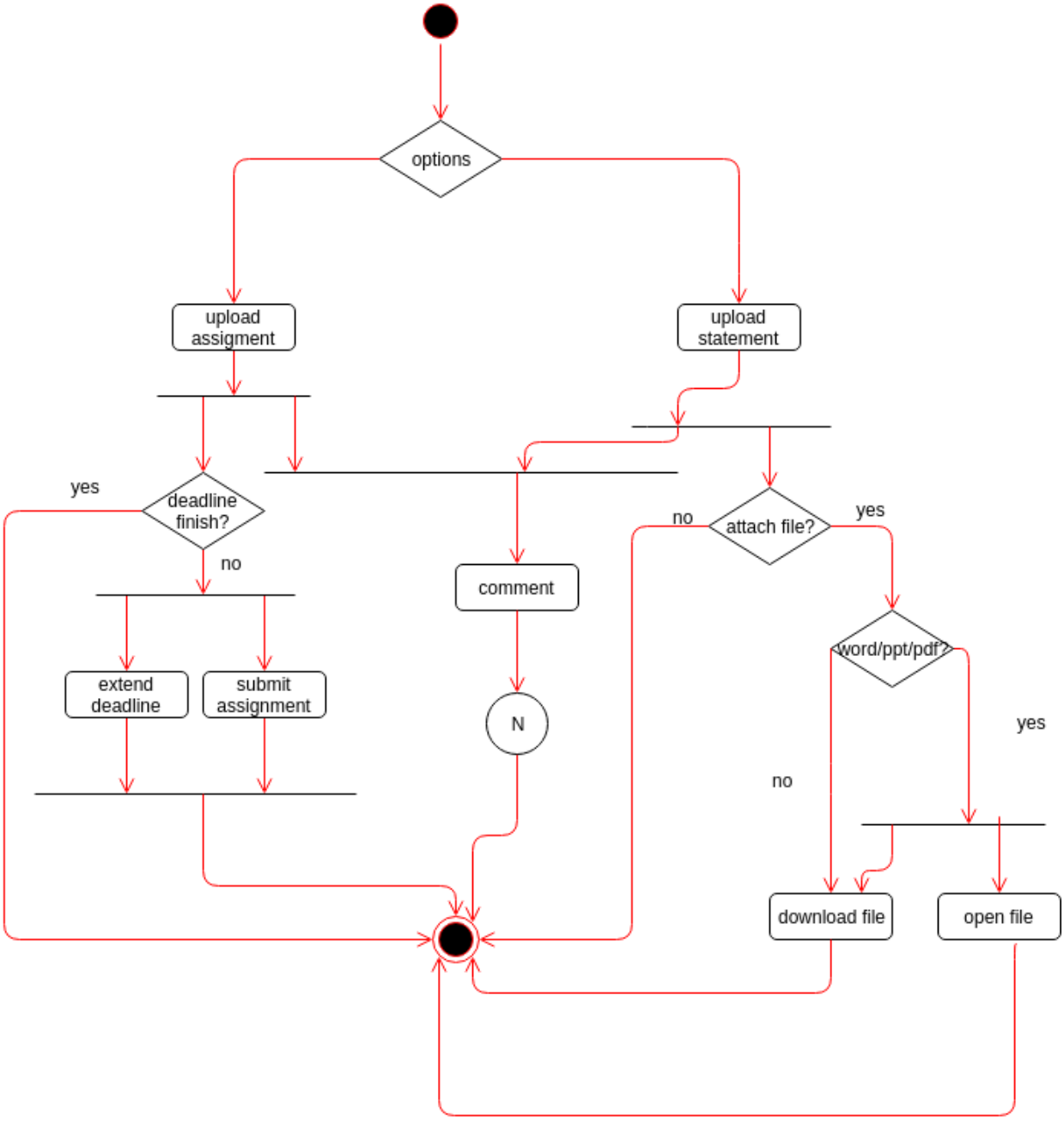

Figure 17: Activity Diagram-Pen Course of SRENIKOKKHO

4.3.11 Activity Diagram- Notification Srenikokkho

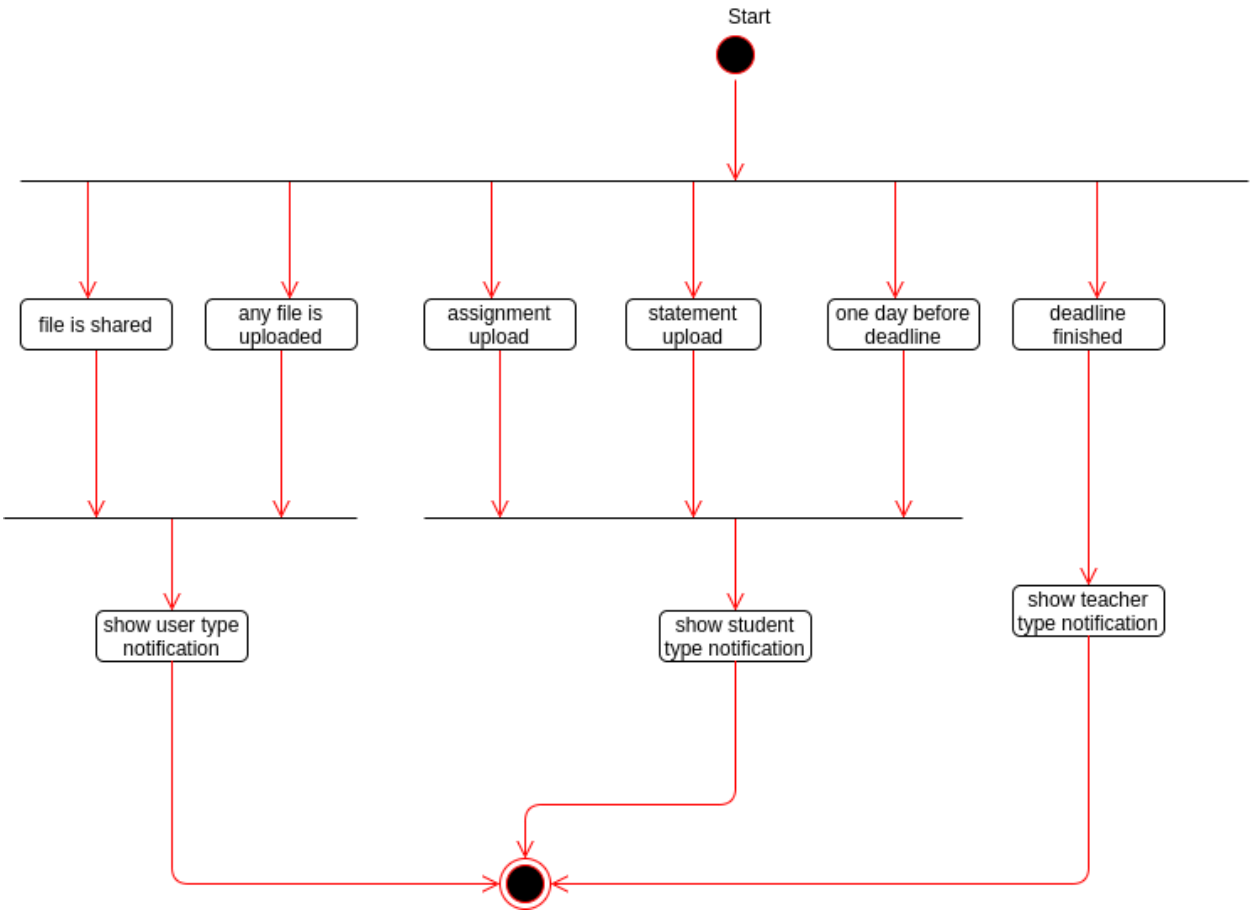

Figure 18: Activity Diagram-Notification of SRENIKOKKHO

4.4 Swim lane Diagrams of SRENIKOKKHO

A swim lane diagram is a visual element used in process flow diagrams, or flowcharts, which visually distinguishes job sharing and responsibilities for sub-processes of a business process. The swim-lane diagrams of the modules described in the previous chapter is shown below:

4.4.1 SWIMLANE DIAGRAM OF AUTHENTICATION

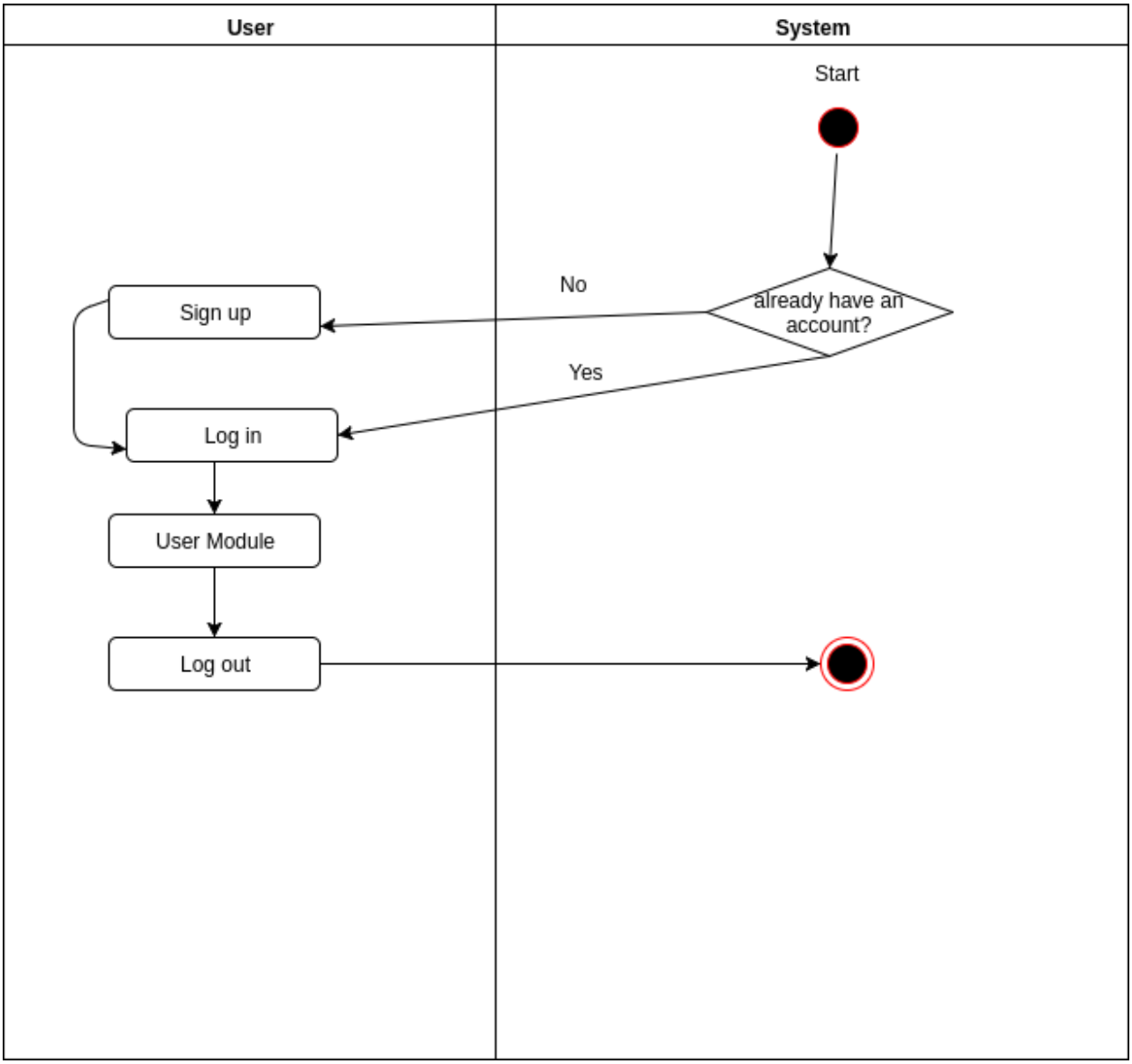

Figure 19: Swim-lane Diagram: Authentication of SRENIKOKKHO

4.4.2 SWIMLANE DIAGRAM OF SIGNUP

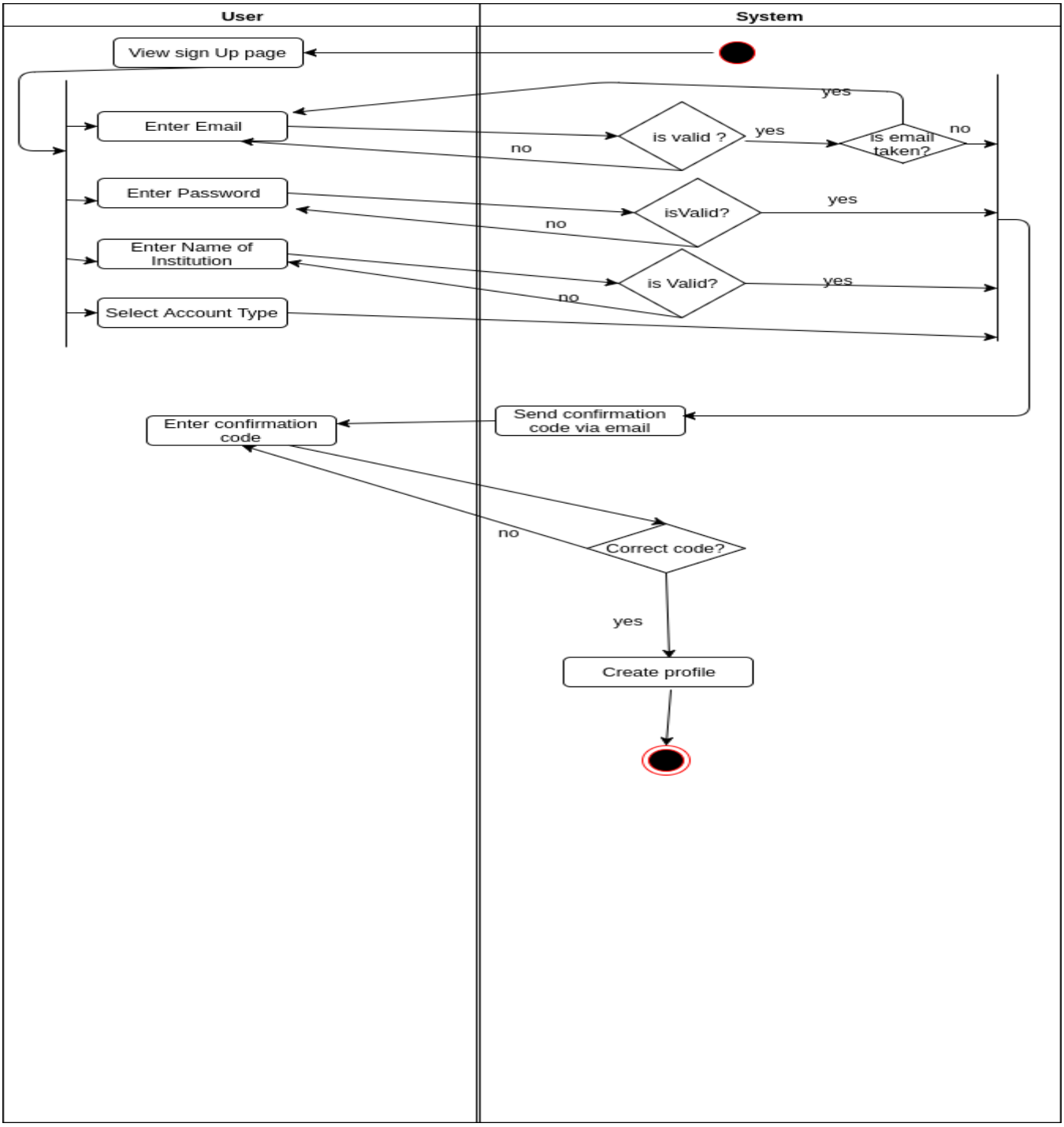

Figure 20: Swim-lane Diagram-Sign Up of SRENIKOKKHO

4.4.3 SWIMLANE DIAGRAM OF LOGIN

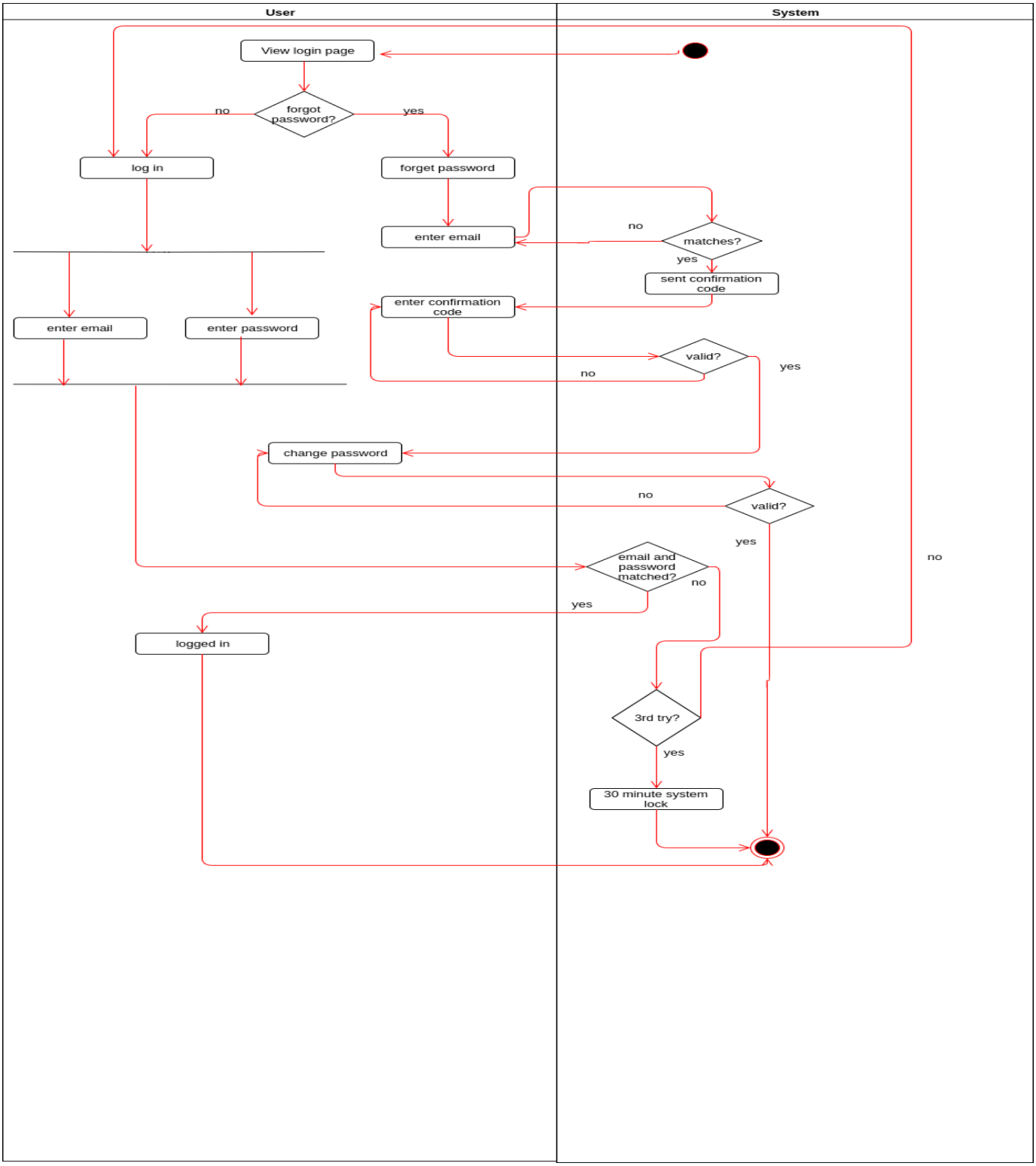

Figure 21: Swim-lane Diagram-Log In of SRENIKOKKHO

4.4.4 SWIMLANE DIAGRAM OF User Module

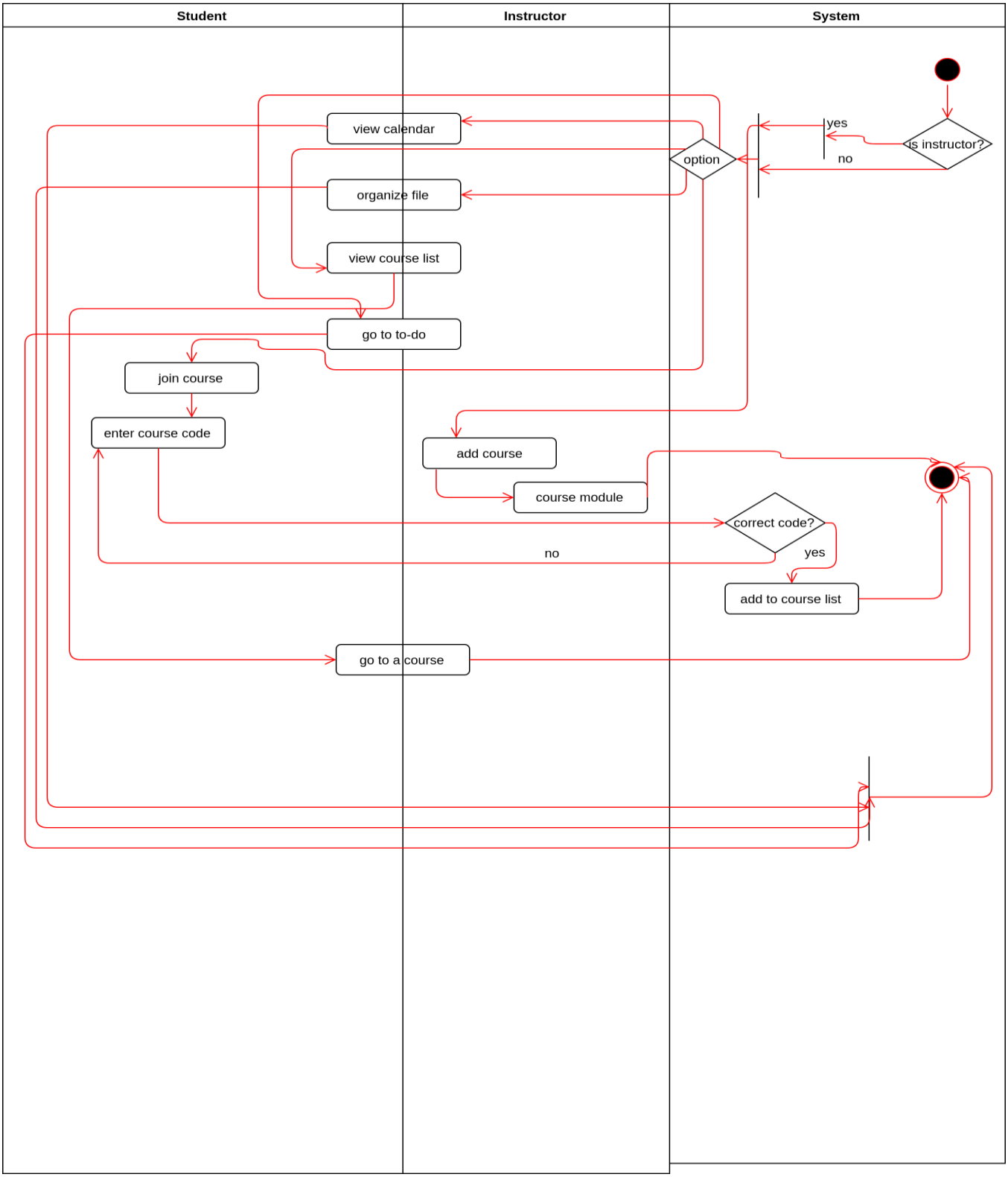

Figure 22: Swim-lane Diagram-User of SRENIKOKKHO

4.4.5 SWIMLANE DIAGRAM OF TO-DO

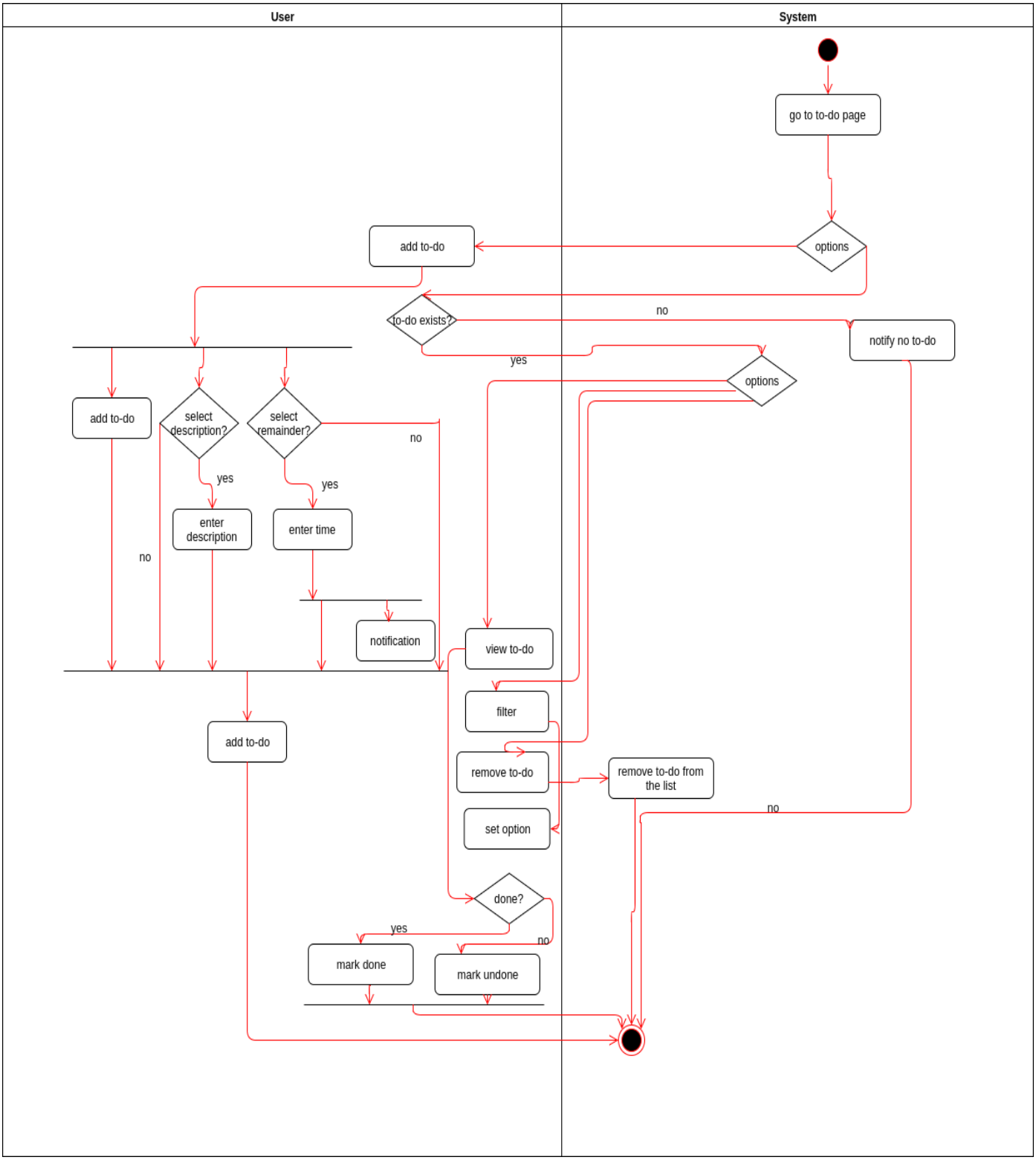

Figure 23: Swim-lane Diagram-To-Do of SRENIKOKKHO

4.4.6 SWIMLANE DIAGRAM OF FILE ORGANIZATION

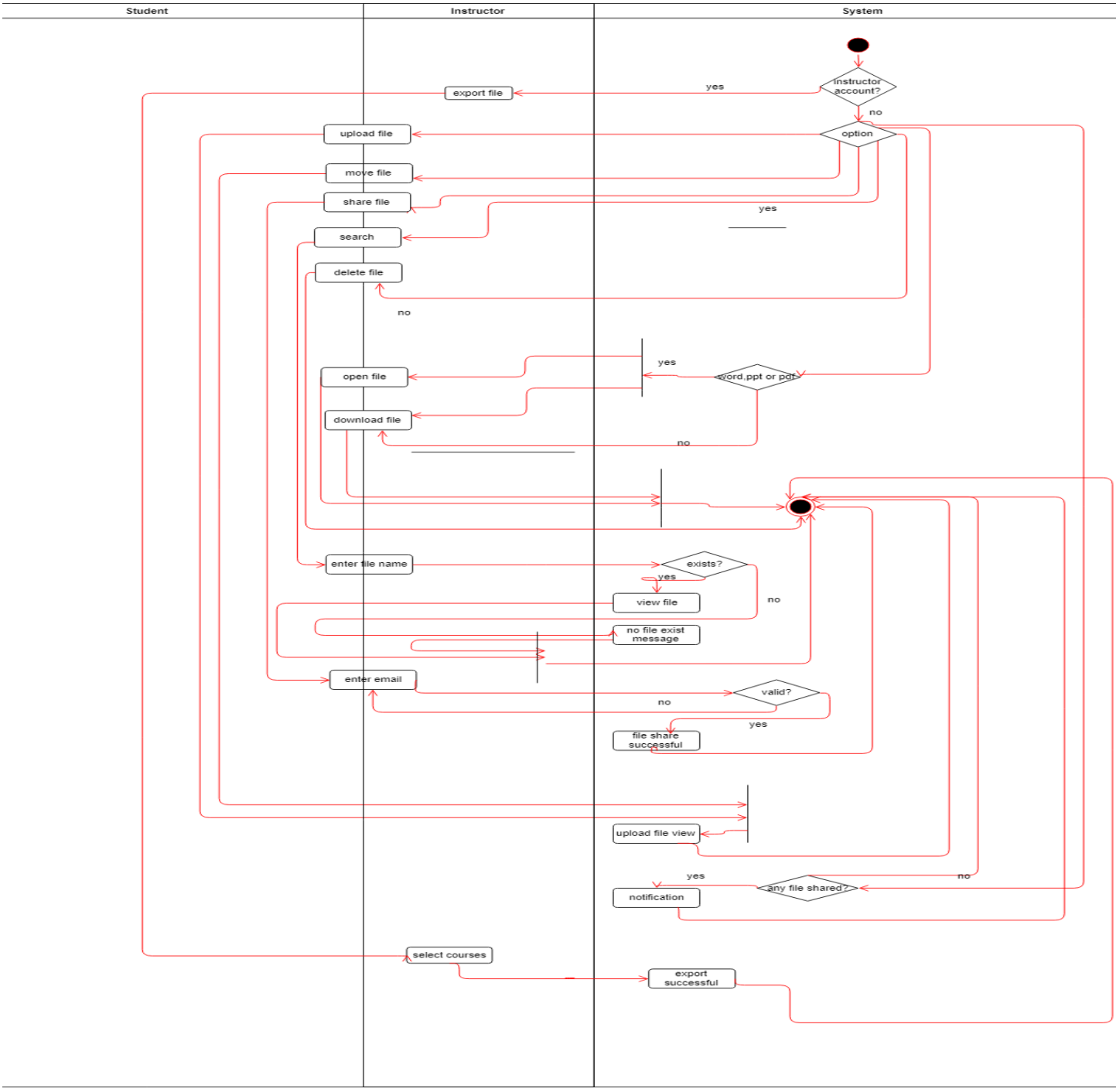

Figure 24: Swim-lane Diagram-Organize file of SRENIKOKKHO

4.4.7 SWIMLANE DIAGRAM OF COURSE

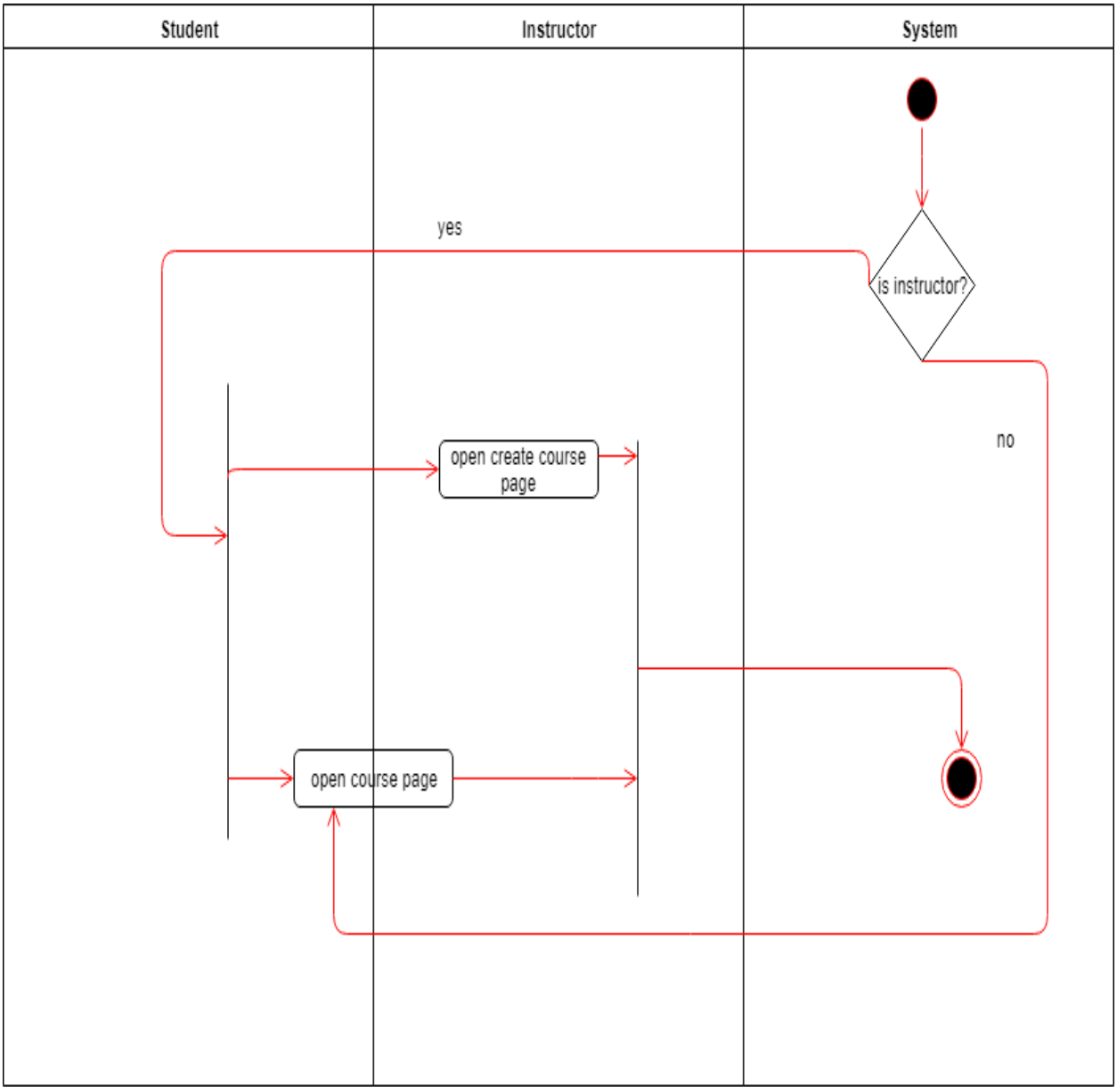

Figure 25: Swim-lane Diagram-Course of SRENIKOKKHO

4.4.7 SWIMLANE DIAGRAM OF CREATE COURSE

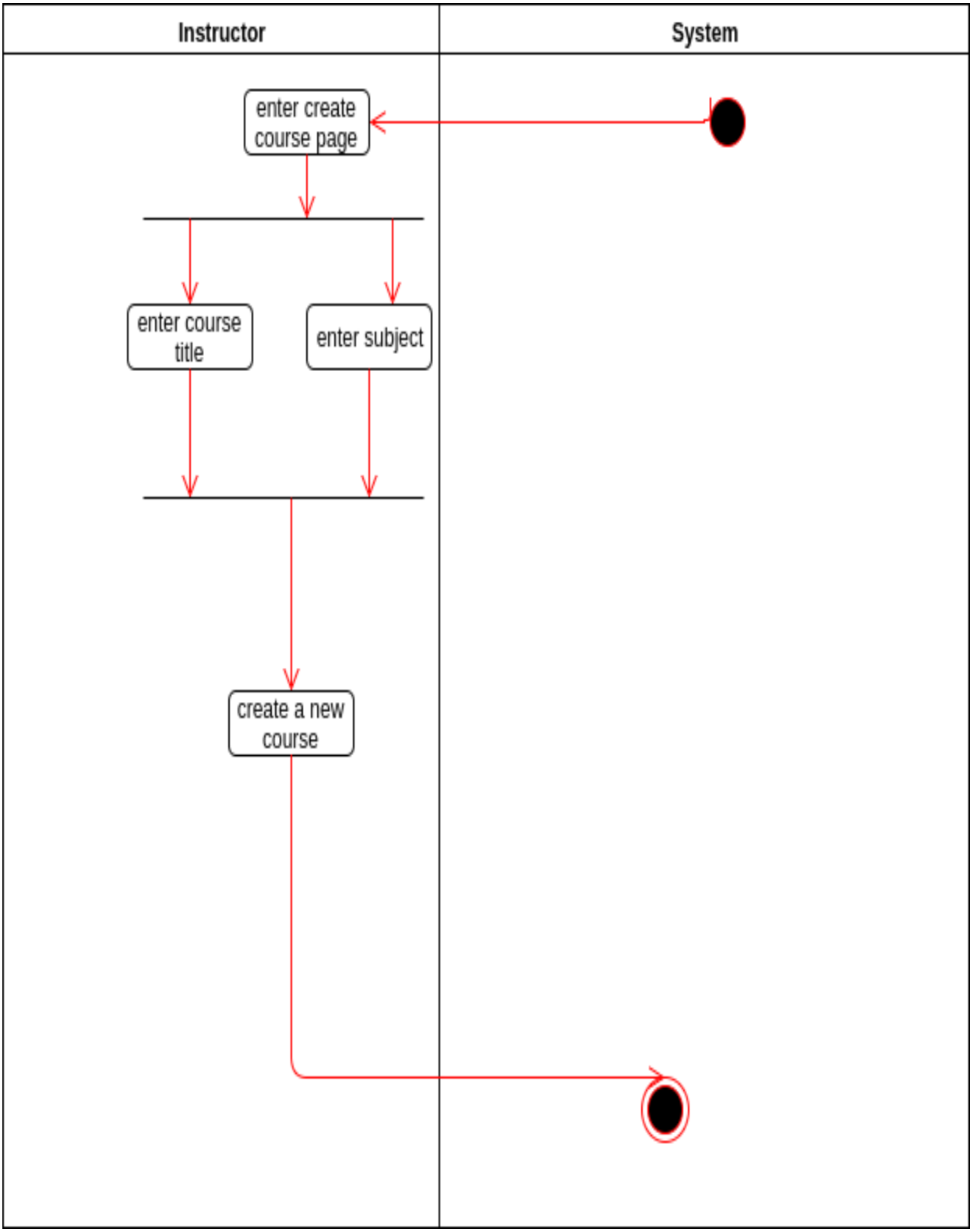

Figure 26: Swim-lane Diagram-Course of SRENIKOKKHO

4.4.8 SWIMLANE DIAGRAM OF OPEN COURSE

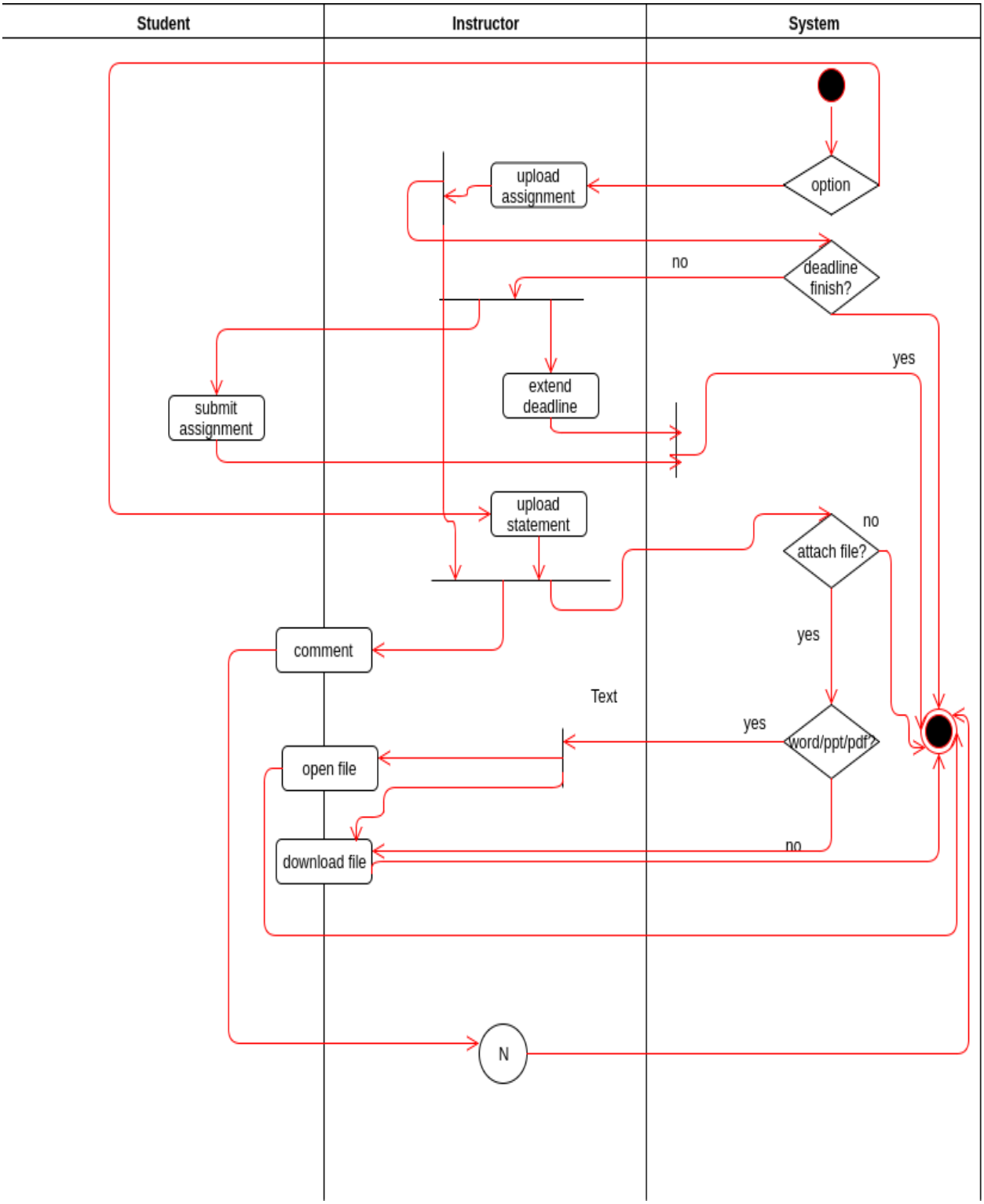

Figure 27: Swim-lane Diagram-Course of SRENIKOKKHO

4.4.9 SWIMLANE DIAGRAM OF NOTIFICATION

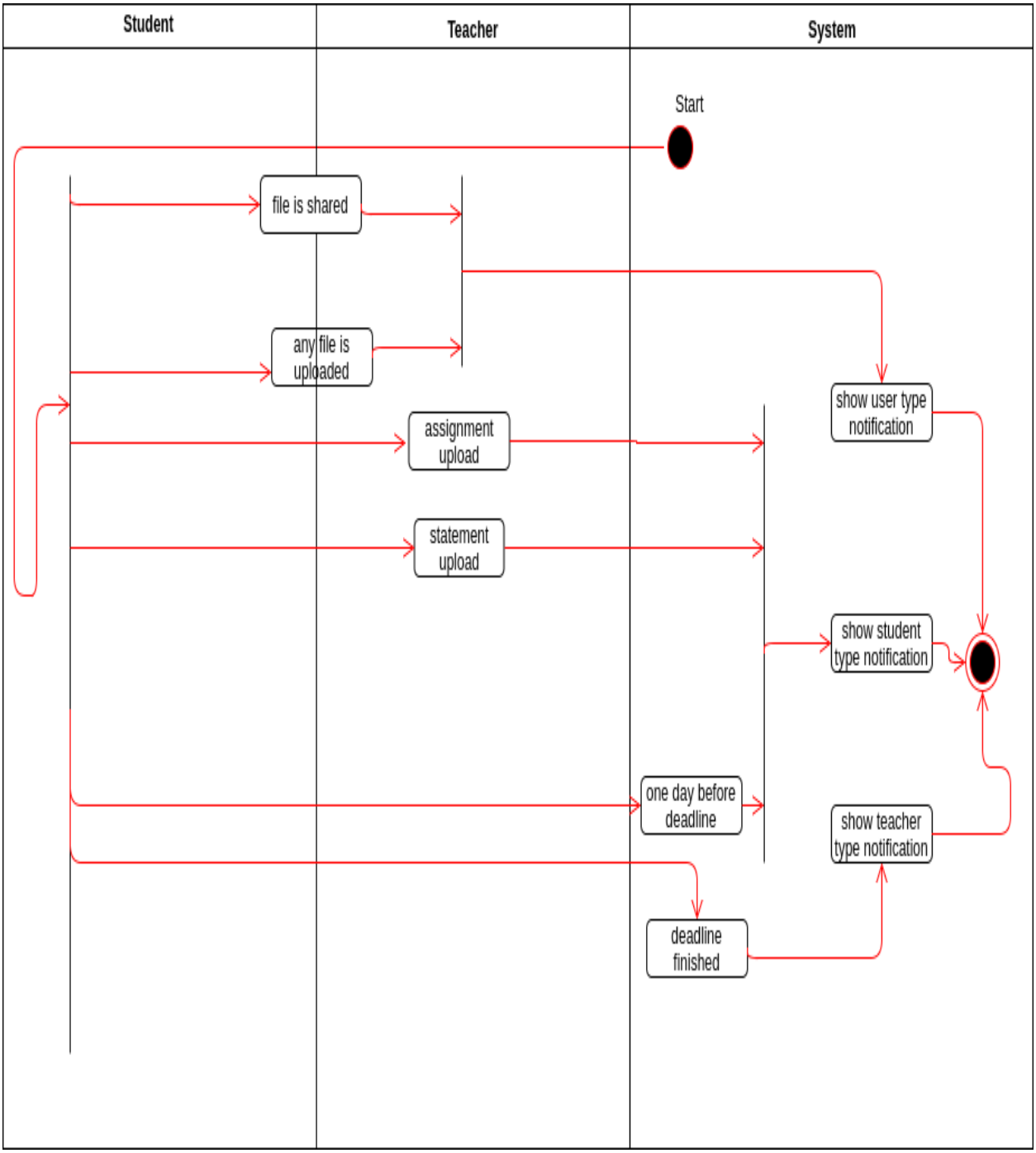

Figure 28: Swim-lane Diagram-Notification of SRENIKOKKHO

CHAPTER 5: DATA-BASED MODEL OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

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 choose to create data model as part of overall requirements modeling. The entity relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects and the information that how the data objects are entered, stored, transformed and produced within the system.

5.1 Grammatical Parsing and Analysis

We identified all the nouns whether they are in problem space or in solution space from our usage scenario and categorized them according to their attributes. In the following table, “P” stands for problem domain and “S” stands for solution space. In table 1, the nouns are identified from the usage scenario of the project-

Table 1: Noun Identification

Serial No	Noun	Only Problem space/ Solution Space	Attributes
1	homepage	P	
2	list	P	
3	features	P	
4	page	P	

5	user	S	17-20
6	Sign up	P	
7	login	P	
8	option	P	
9	information	P	
10	functionalities	P	
11	web	P	
12	application	P	
13	account	S	17-20
14	student	S	17-20
15	instructor	S	17-20
16	registration	P	
17	email	S	
18	password	S	
19	institution	S	
20	Account type	S	
21	characters	P	
22	fields	P	
23	user	S	17-20
24	confirmation	P	
25	code	P	

26	verification	P	
27	logout	P	
28	repository	P	
29	system	P	
30	mode	P	
31	course	S	49,50
32	file	S	34,36
33	folder	S	34
34	name	S	
35	module	P	
36	format	S	
37	recent folder	S	34
38	shared folder	S	34
39	srenikokkho	P	
40	pdf	P	
41	word	P	
42	ppt	P	
43	to-do	S	45-48
44	calendar	S	45-48
45	work	S	
46	description	S	

47	checkbox	S	
48	remainder	S	
49	title	S	
50	subject	P	
51	assignment	S	55
52	statement	S	
53	attachment	S	
54	device	P	
55	deadline	S	
56	comment	S	
57	section	P	
58	message	S	
59	notification	S	65-67
60	circumstances	P	
61	post	S	65,67
62	day	P	
63	date	S	
64	title	S	
65	id	S	

5.2 Potential Data Objects

After grammatical parsing, we identified the following data objects that may need in our system for information storage.

User-17-20

Account-17-20

Instructor-17-20

Student-17-20

Course-49-50

File-34, 36

Folder-34

Recent folder-34

Shared folder-34

To-do-45-48

Calendar- 45-48

Assignment-55

Notification-65-67

Post-65-67

5.3 Analysis of Potential Data Objects

- We do not need to store individual shared files and recent files. Instead of making two different entity we can store any type of file as “user-folder”.
- To-do and calendar have same attributes so those can be merged as to-do.

5.4 Final Data Objects

In table 2, the attributes of the final data objects are shown-

Table 2: Attributes identification of final data objects

Serial no	Data Objects	Attributes
1	User	email, password, institute, username
2	Teacher	email, password, institute, username
3	Student	email, password, institute, username
4	Course	id, title, subject
5	Assignment	id, deadline, time
6	Attached file	type, name
7	Statement	id, time
8	Comment	id, date, description
9	Folder	name
10	File	name, type
11	Notification	id, date, title, description, link

5.5 Data Object Relationship

In figure 29, the relationship among the data object member, playlist, admin and songs are shown-

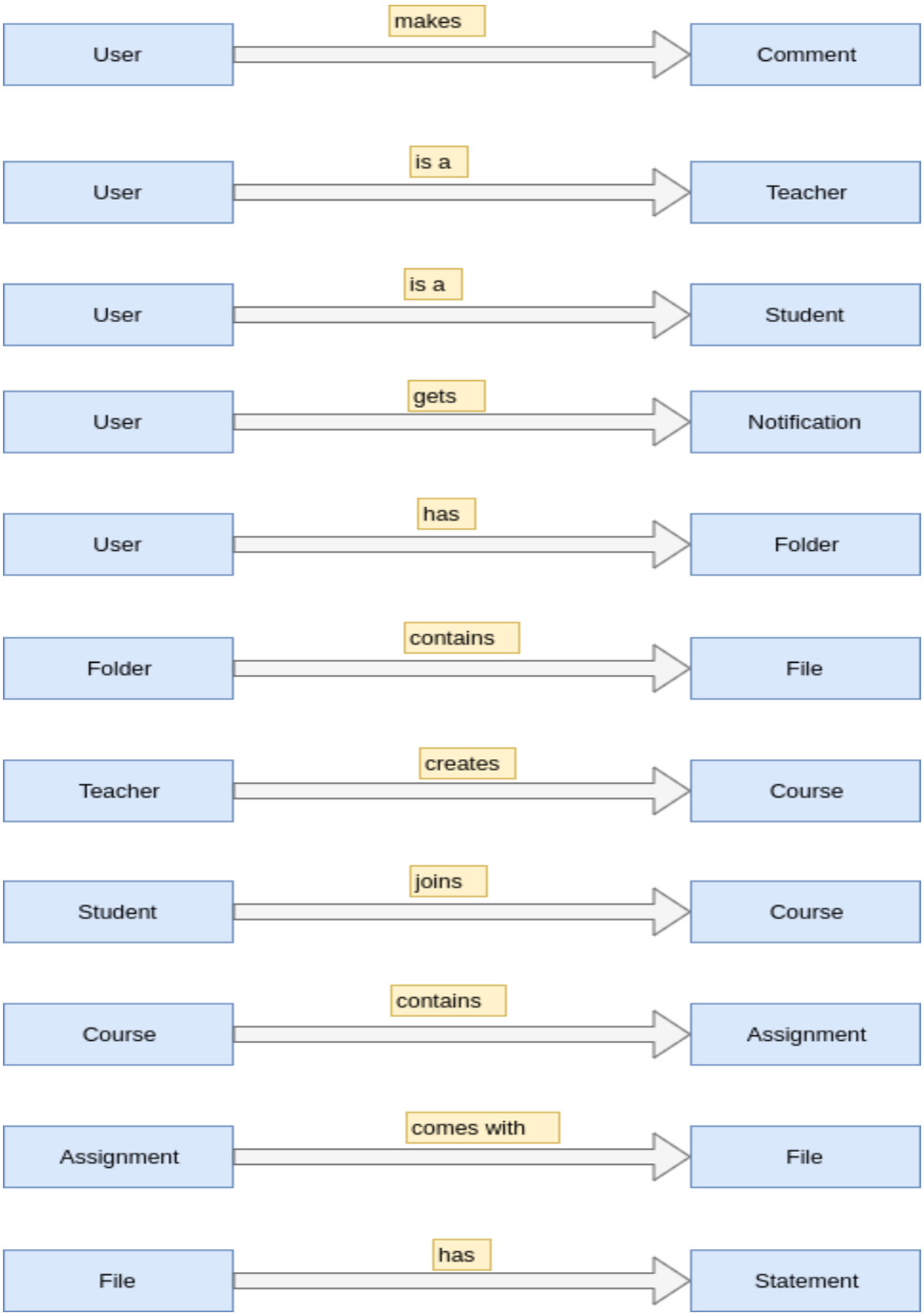

Figure 29: Data object relationship

5.6 Entity Relation Diagram

An entity-relationship diagram (ER Diagram) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities. The ER diagram of the system of our application is shown in figure 30:

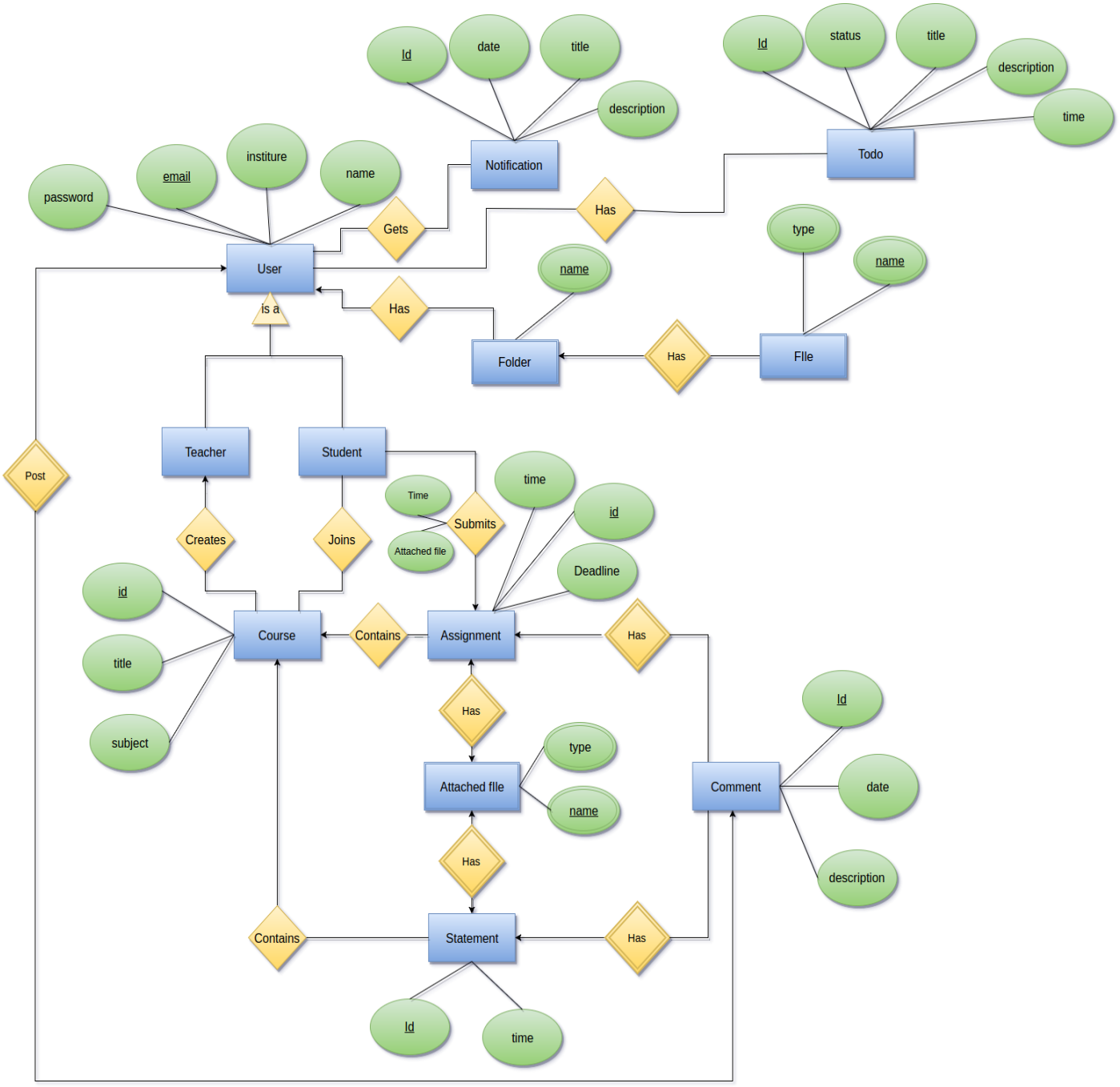

Figure 30: ER diagram

5.7 NoSQL data modeling

We will use MongoDB which is a document based nosql for this project. Data in MongoDB has a flexible schema. Unlike SQL databases, where you must determine and declare a table's schema before inserting data, MongoDB's collections (equivalent to a table of a SQL database) do not enforce document (equivalent to a row of a SQL database) to be structured. This flexibility facilitates the mapping of documents to an entity or an object. Each document can match the data fields of the represented entity, even if the data has substantial variation. In practice, however, the documents in a collection share a similar structure.

5.7.1 No-SQL Schema Diagram

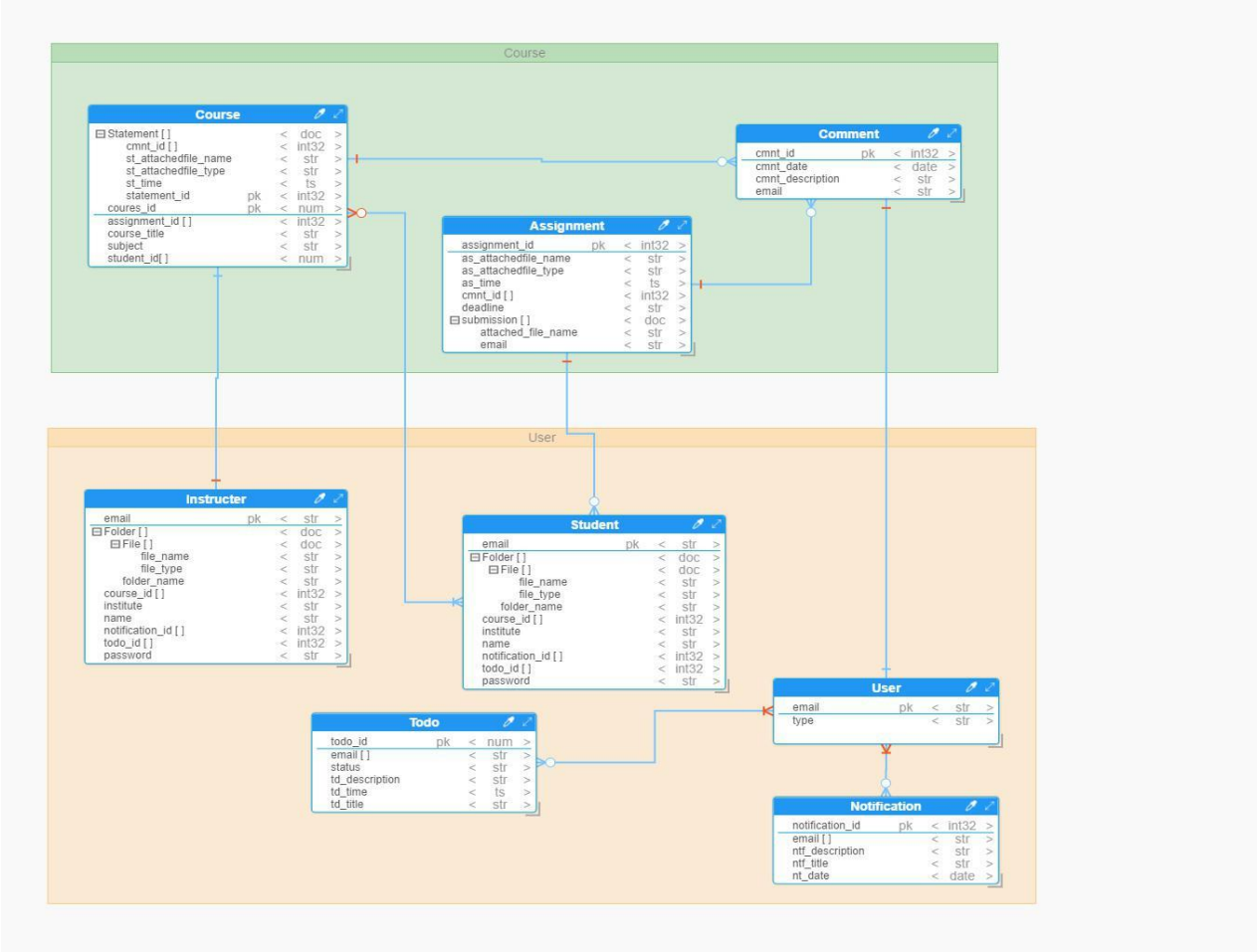

Figure: 31 No-SQL Schema Diagram

CHAPTER 6: CLASS-BASED MODEL OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

Class-based modeling represents the objects that the system will manipulate, the operations that will apply to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.
As we will be using functional programming to complete this project, we do not need to find out objects. So this modeling is not necessary for this project.

CHAPTER 7: BEHAVIORAL MODEL OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

The behavioral model indicates how software will respond to external events. Two different behavioral representations are discussed in this chapter. The first indicates how individual class changes state based on external events and the second shows the behavior of the software as a function of time.
As we will be using functional programming to complete this project, we do not have any object or class. So this modeling is not necessary for this project.

CHAPTER 8: DATA FLOW MODEL OF ‘SRENIKOKKHO’-A LEARNING MANAGEMENT BASED WEB APPLICATION

A data flow model is diagramatic representation of the flow and exchange of information within a system. Data flow models are used to graphically represent the flow of data in an information system by describing the processes involved in transferring data from input to file storage and reports generation. A data flow model may also be known as a data flow diagram (DFD).

Data flow modeling can be used to identify a variety of different things, such as:

- Information that is received from or sent to other individuals, organizations, or other computer systems
- Areas within a system where information is stored and the flows of information within the system are being modeled
- The processes of a system that act upon information received and produce the resulting outputs

Data Flow Diagrams of “Srenikokkho” is given below

8.1: Context-Level

Figure 32: context-level

8.2: Level-1

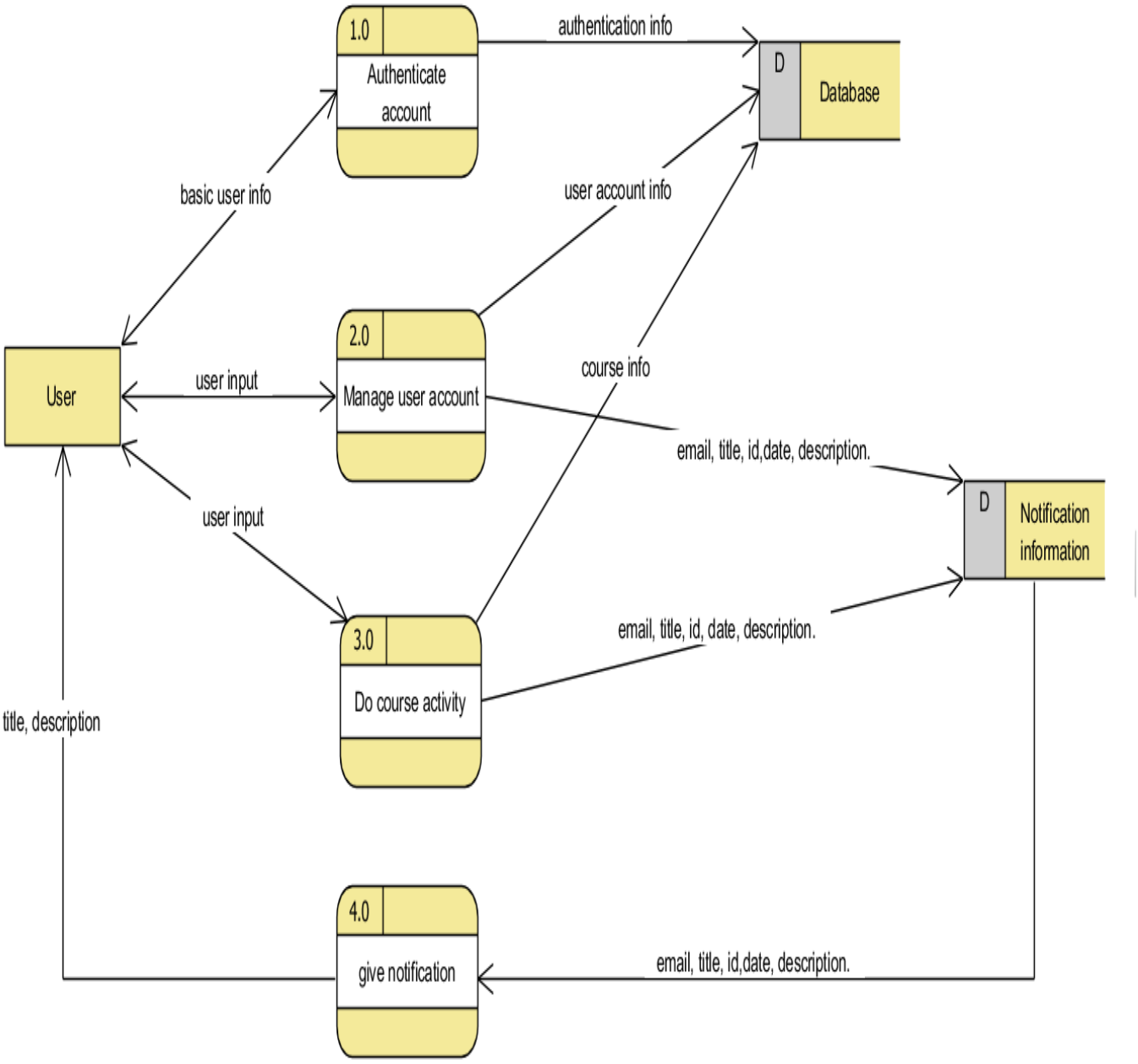

Figure 33: Level-1

8.3: Level 2 Of 1.0

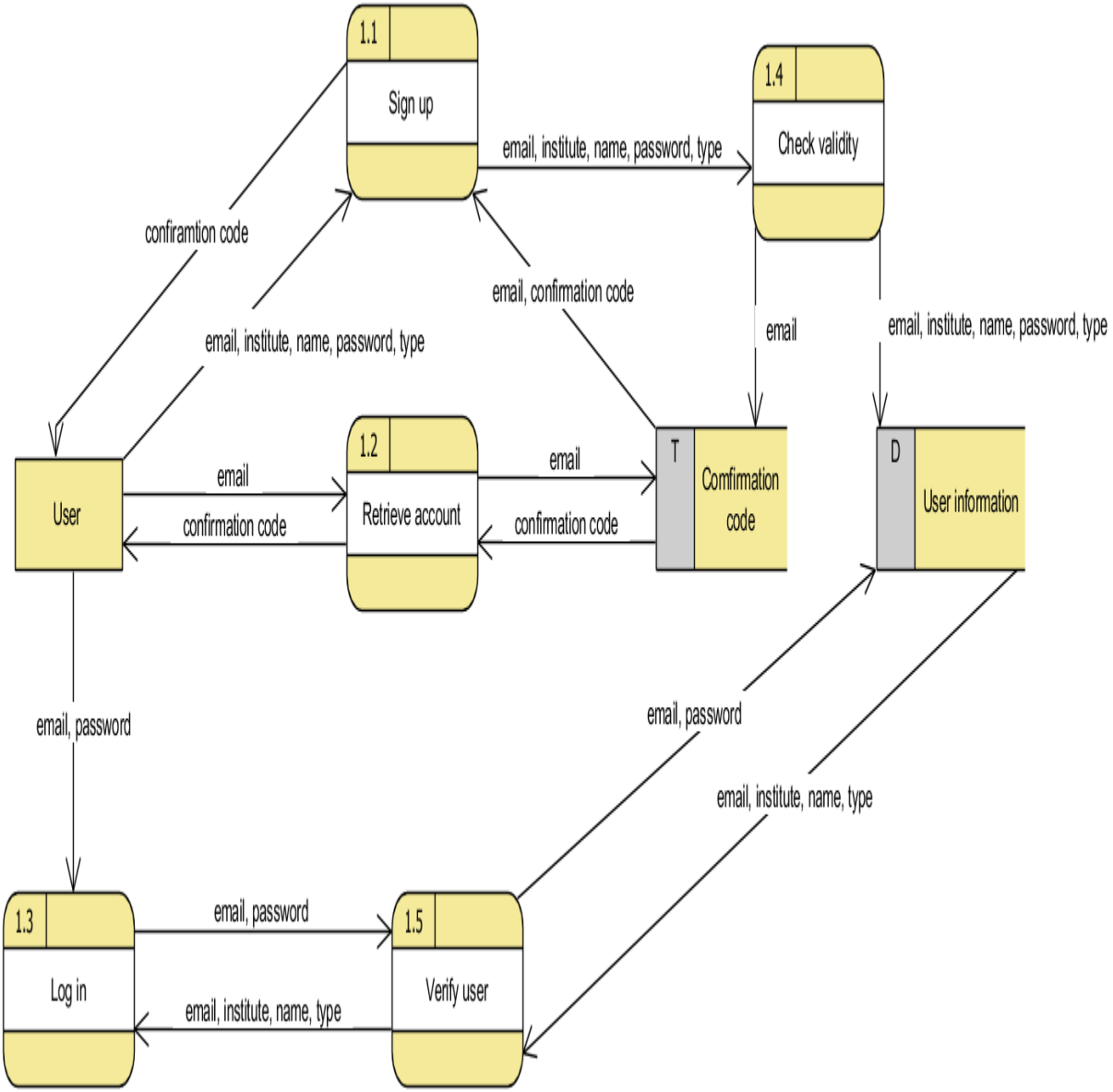

Figure 34: Authentication

8.4: Level 2 Of 2.0

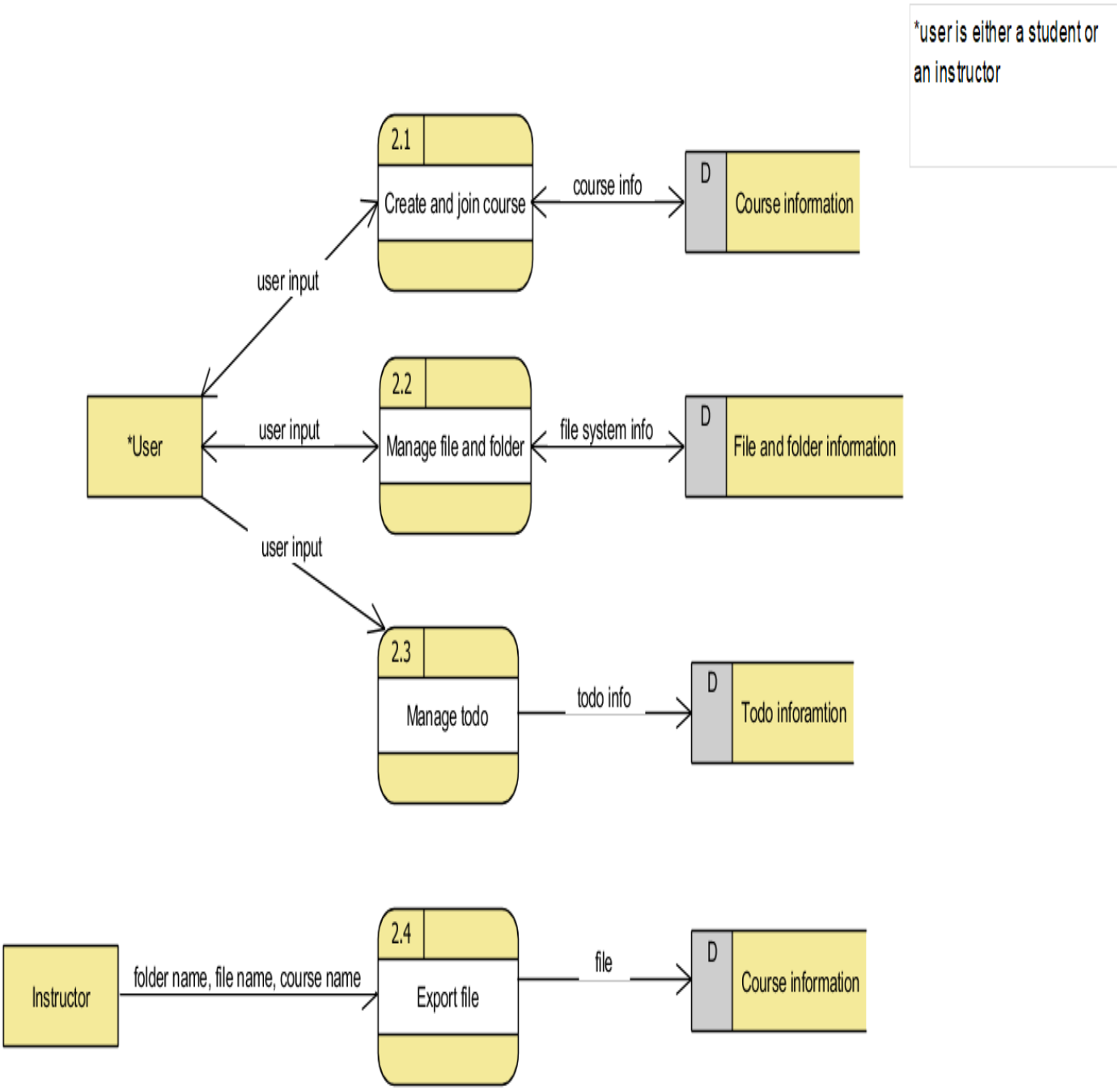

Figure 35: Manage User Account

8.5: Level 2 Of 3.0

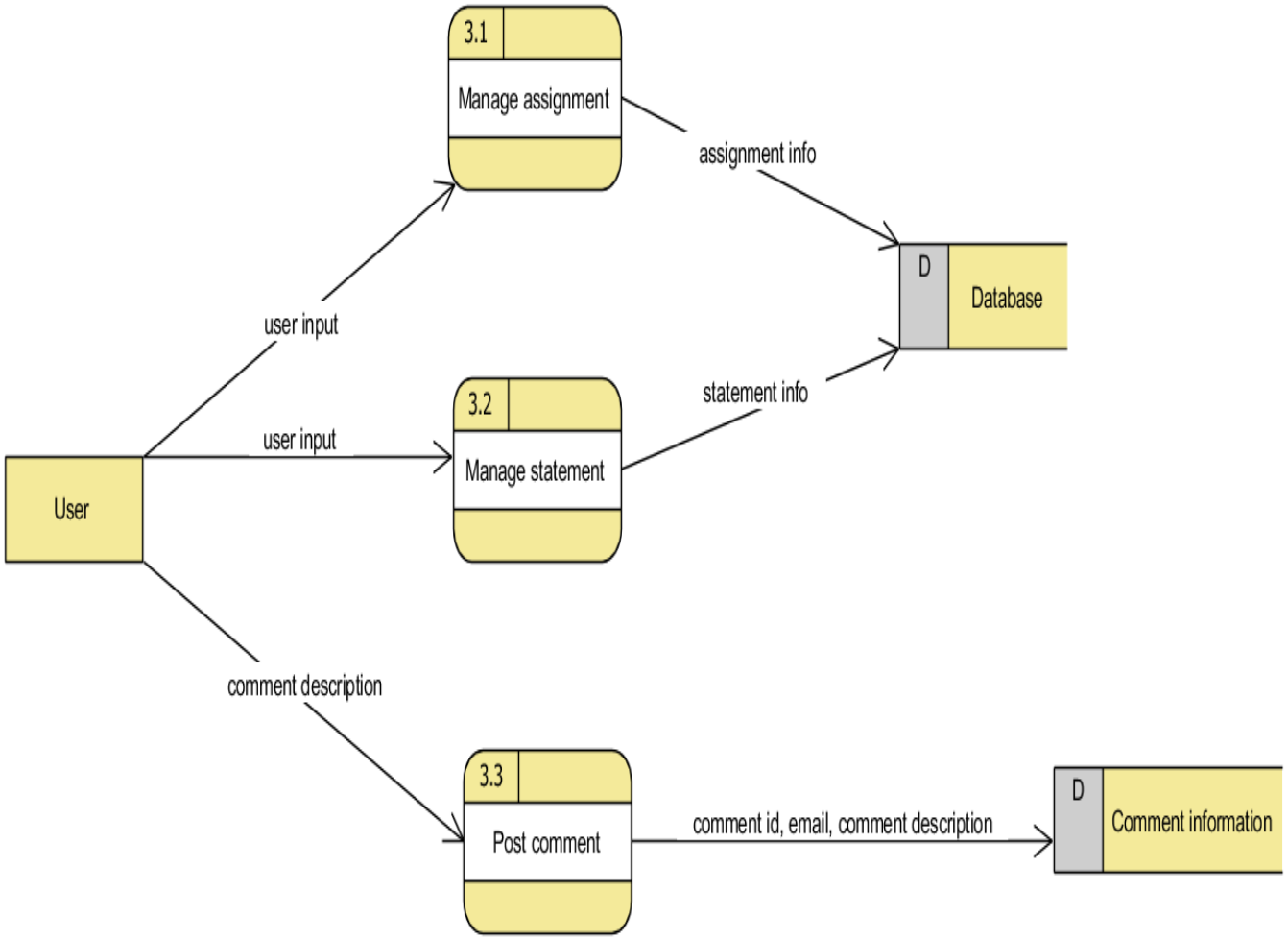

Figure 36: Manage Course

8.6: Level 3 of 2.1

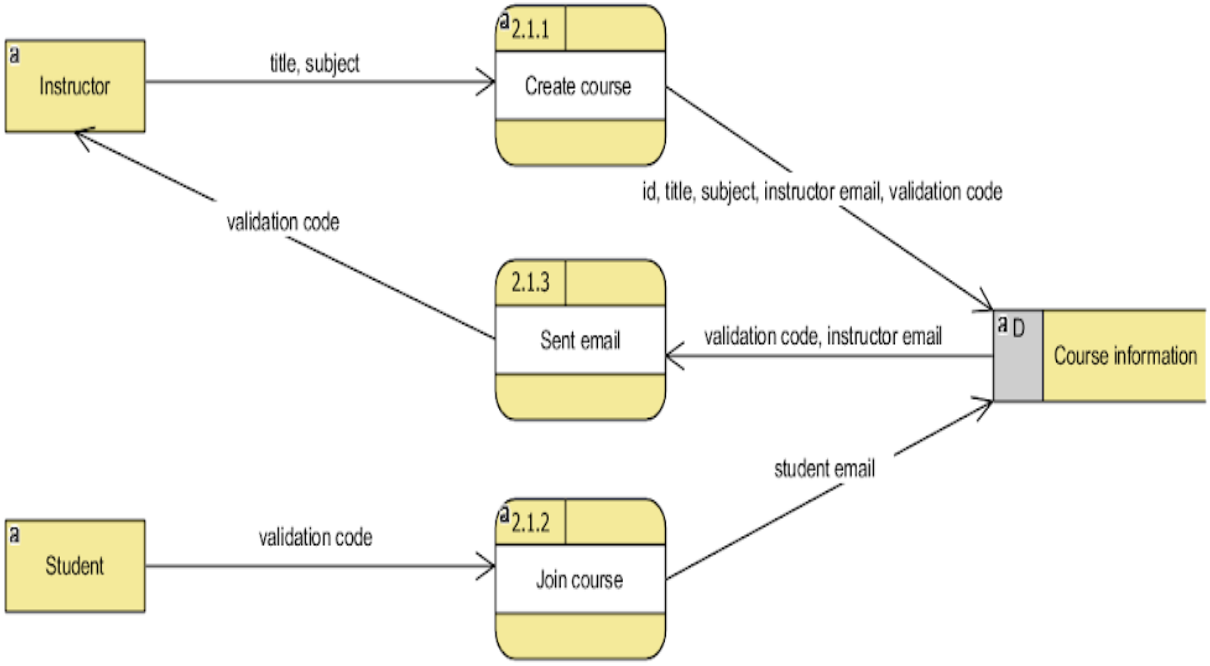

Figure 37: Create and join Course

8.7: Level 3 of 2.2

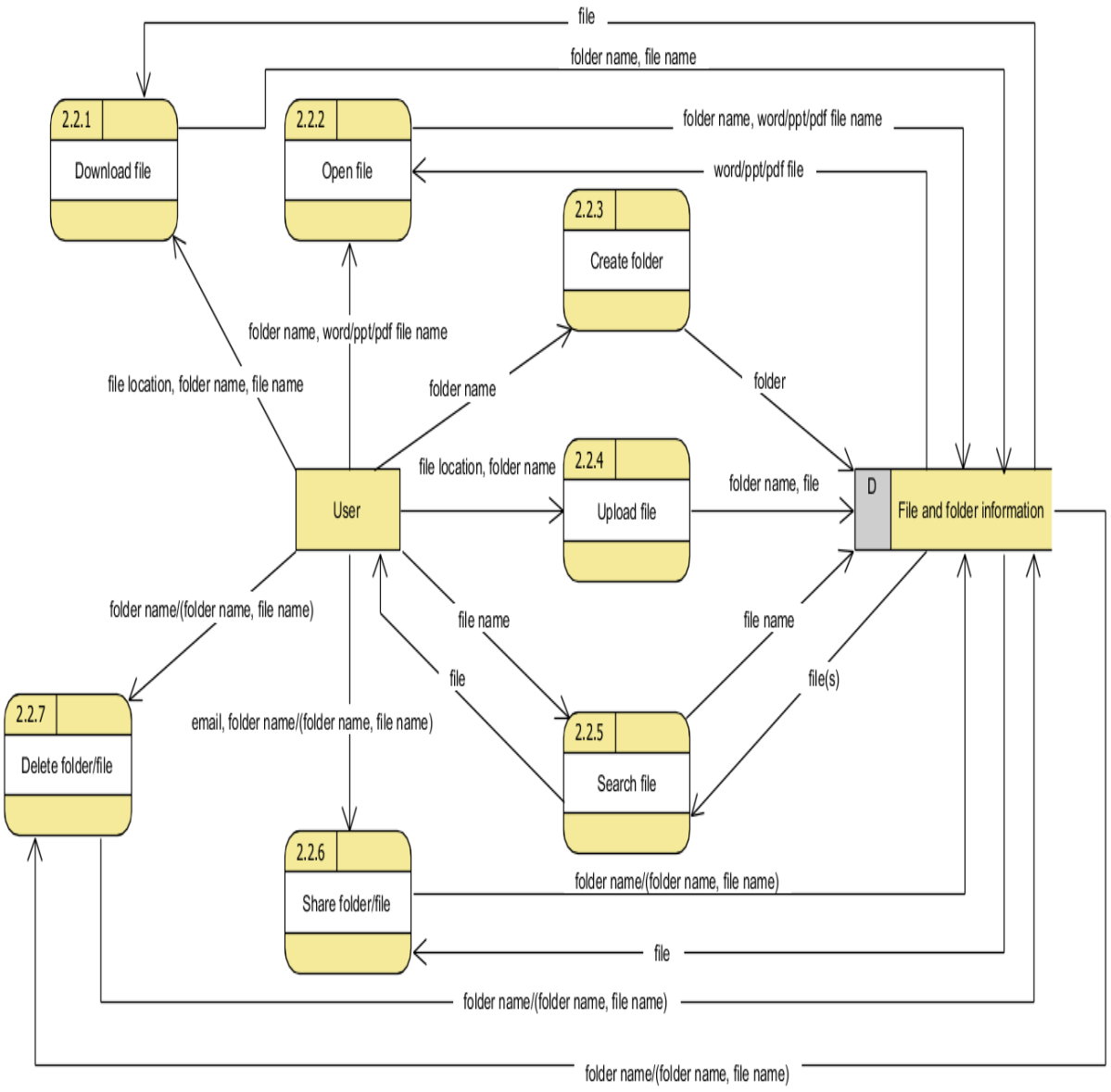

Figure 38: File Management

8.8: Level 3 of 2.3

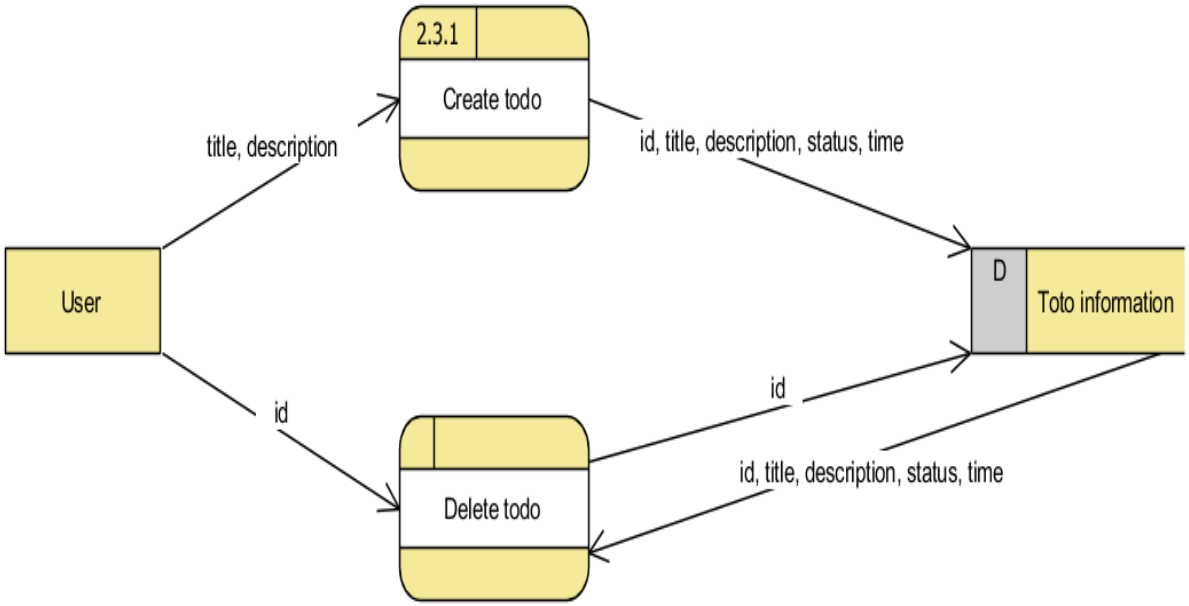

Figure 39: To-do

8.9: Level 3 of 3.1

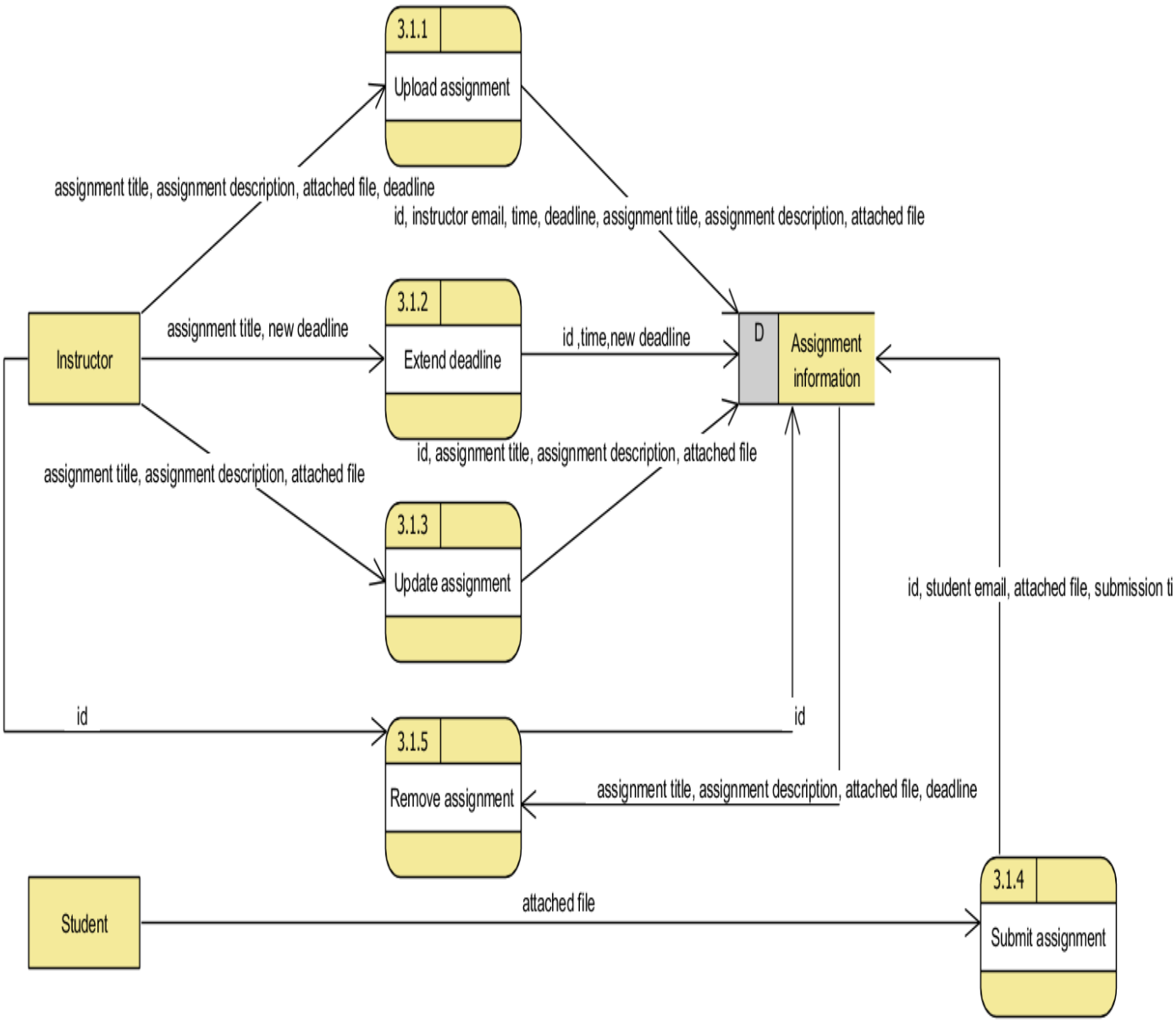

Figure 40: Assignment Management

8.10: Level 3 of 3.2

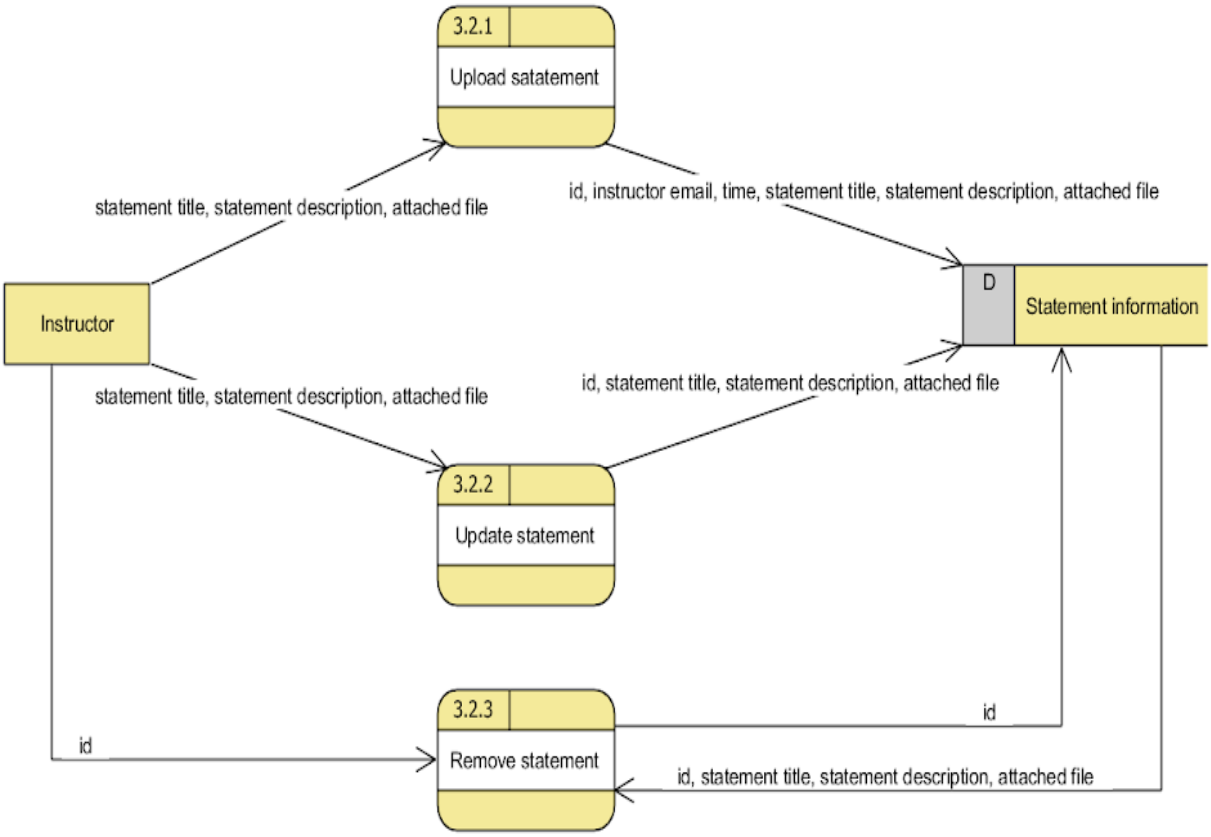

Figure 41: Statement Management

CHAPTER 9: CONCLUSION

We are pleased to submit the final SRS report of our project. From this, the readers will get a clear and easy view of the overall system. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using this SRS.

REFERENCES

1. Book- Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)

2. Book-Abraham Silberchatz, Henry F.Korth, S.Sudarshan: Database System Concept (6th Edition)

3. https://www.techopedia.com/definition/28523/data-flow-model Data Flow Model, last accessed on 19/03/2018

4. https://docs.mongodb.com/manual/core/data-modeling-introduction/ Data Modeling Introduction, last accessed on 19/03/2018