# *PIICHKARI*

*An Online Paint Editor*

# PIICHKARI (ONLINE PAINT EDITOR)

## Software Requirement Specification

**Course: SE 505 Software Project Lab II**

Submitted to
**Manager SPL 2**
Institute of Information Technology
University of Dhaka

Supervised By
**Md. Nurul Ahad Tawhid**
Assistant Professor
Institute of Information Technology
University of Dhaka

Submitted By
**Shahla Shaan Ahmed (BSSE 0810)**
**Taslima Binte Kamal (BSSE 0835)**
BSSE Session: 2015-2016


Institute of Information Technology
University of Dhaka

20-03-2018

# LETTER of TRANSMITTAL

19th March, 2018

Md. Nurul Ahad Tawhid

Assistant Professor,

Institute of Information Technology,

University of Dhaka

Subject: Submission of Software Requirement Specification report of Software Project Lab-II

Sir,

With due respect, we are very glad to submit our Software Requirement Specification report of our project PIICHKARI- An online paint editor. Although this report may have many drawbacks, we have given our highest effort to fulfill all the requirements of our stakeholders and to prepare a detailed documentation that will help us in future.

Therefore, we hope that you will be kind enough to overlook our mistakes and acknowledge the our attempt that we have put in this document.

Sincerely,

Shahla Shaan Ahmed (0810)

Taslima Binte Kamal (0835)

BSSE08 Batch

Institute of Information Technology,

University of Dhaka

---

Signature of Supervisor

# ACKNOWLEDGEMENT

We have given our level best to complete this document. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We are highly indebted to our supervisor Md. Nurul Ahad Tawhid, Assistant Professor, Institute of Information Technology for his constant guidance and supervision as well as for providing necessary information regarding the project.

We would like to express our gratitude towards the coordinators of Software Project Lab-II- Dr. Mohammed Shafiul Alam Khan (Assistant Professor), Amit Seal Ami (Lecturer) and Md. Saeed Siddik (Lecturer) for their guidance and encouragement which helped us in completion of this document.

We would like to express our special gratitude and thanks to the students of Institution of Fine Arts for giving us their valuable opinion regarding this project.

Our thanks and appreciations also go to our classmates who have willingly helped us out with their abilities.

# ABSTRACT

This document contains the requirements and specifications of our project PIICHKARI- An online paint editor. This document incorporates inception, elicitation, scenario based model, data model, class based model, behavioral model. In this document we tired to define the requirements based on our stakeholders' point of view. Taking the stakeholders' input we tried to resolve the issues that may come up in future implementation.

PIICHKARI is an online paint editor which provides a facility for the users to draw and paint pictures and also provides a chance to interact with one another by giving likes and comments in their pictures. Through this document we have gathered the requirements, analyzed them and tried to come up with a solution for designing the system.

This document will provide a workflow to guide us through the implementation.

# Table of Contents

# List of Figures

# List of Tables

# CHAPTER ONE: INTRODUCTION OF PIICHKARI- ONLINE PAINT EDITOR

## 1.1 Purpose

The purpose of this document is to collect and analyze all assorted ideas that have come up to define this paint editor, its requirements with respect to stakeholders. In short, the purpose of this SRS document is to provide a detailed overview of our paint editor, its parameters and goals. This document describes our project's target audience and its requirements. The software requirement specification (SRS) contains the normal, expected and exciting requirements. By resolving these requirements, we can determine the baseline requirements for our painting tool. It defines how we and our stakeholders view the painting tool and its functionalities.

### 1.1.1 Vision Statement

The aim of this project is to create an online painting tool which will provide the facility for the users to draw and paint. This painting tool will also enable the users to view other user's pictures and to give likes and comments.

### 1.1.2 Scope

The scope of our project involves the development of a painting tool which includes a canvas, different user interface for different audience, providing various options for the users such as: save image, download image, report other user, like and comment in other user's images, edit previous image.

## 1.2 Intended Audience

This SRS is intended for various users including users, admin, designer and developer.

- ❏ User will use this SRS document to ensure that the final product will contain all the requirements they have asked for.
- ❏ Designers will use SRS to get a idea of stakeholder's point of view and design the painting tool accordingly.
- ❏ Developers will use this document to develop a workflow for implementation and to fulfill the requirements that the stakeholders have asked for.

The analysis of our intended audience helped us to have a better understandings of the view of our audience.

# CHAPTER TWO: INCEPTION OF PIICHKARI- ONLINE PAINT EDITOR

Inception forms the basic understanding of the problem and the nature of the solution. Inception is the beginning phase of requirements engineering. It defines how a software project 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 a software project. At project inception, we established 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 software team.

During inception we asked a series of questions to establish

- ➔ A basic understanding of the problem.
- ➔ People who want a solution.
- ➔ The nature of the solution that is desired.
- ➔ The effectiveness of preliminary communication and collaboration between us and the other stakeholders.

Through these questions we-

- ➢ Identified the list stakeholders.
- ➢ Recognized multiple viewpoints.
- ➢ Worked towards collaboration.
- ➢ Broke the ice and initiated the communication.

## 2.1 List of Stakeholders

A stakeholder in any group or individual who can affect or is affected by the system. [1] Stakeholders include end-users who will interact with the painting tool and everyone else that may be affected by its installation. The list of stakeholders kept growing as we kept asking every stakeholder, "Whom else do you think we should talk to?"

To identify the stakeholders we consulted with the Admin of the web application and asked him following questions:

- ➢ Who will be using this painting tool's outcomes?
- ➢ Who gets to make the decisions about PIICHKARI?
- ➢ Who have resources we need to get the painting tool done?
- ➢ Whose work will affect the painting tool?

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

1. Admin
2. User
3. Developer

**1. Admin:** Admin will handle all the server side activities for PIICHKARI. Admin can view the necessary information from the users but cannot modify them.

**2. User:** They will use the application. They can draw in their canvas and save their picture in their own account and create their own album. They can search other users and their pictures. They can also give like and comment other users' picture.

**3. 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.

## 2.2 Recognizing Multiple Viewpoints

Different stakeholders are connected to our paint editor differently. They have multiple views of the editor. They attain different benefits from the editor. So through this phase we tried to recognize every stakeholders view about the project.

The viewpoints of the stakeholders are given below:

**Viewpoints of Admin:**

- ❑ User will have to register first in order to use the paint editor.
- ❑ Members will be authenticated before logging in to the homepage.
- ❑ Each user will have their personal account.
- ❑ User can draw and paint using the canvas.
- ❑ User can save the picture in the gallery.
- ❑ User can edit and download picture.
- ❑ Any user can search other users and view their pictures.
- ❑ Any user can give like and make a comment in other user's pictures.
- ❑ User can comment in her own picture.
- ❑ Admin can view user information.
- ❑ Admin can ban any user.

**Viewpoints of Designer:**

- ❑ Suggestion will be provided during searching other user.
- ❑ Allow users to add or edit their personal information.
- ❑ Every confirmation message will be sent to email addresses of the users.
- ❑ Allow members to delete personal account.
- ❑ Canvas will be provided to the user to draw and paint.
- ❑ Different size of brushes will be provided in the canvas.
- ❑ User can save the picture in the gallery.
- ❑ User can edit and download picture.
- ❑ Admin will have the same user interface with one additional facility to view other users' information.

**Viewpoints of User:**

- ❑ User's personal information will be hidden from other user.
- ❑ Each user will have their personal account.
- ❑ User can draw and paint using the canvas.
- ❑ User can save the picture in the gallery.
- ❑ User can edit and download picture.
- ❑ Canvas will contain a color palette.
- ❑ User can search other user with their name.
- ❑ Suggestion will be provided during searching other user.
- ❑ User can make album with drawn pictures.
- ❑ One user can report another user.

**Viewpoints of Developer:**

- ❑ Cost within budget
- ❑ Detail documentation

## 2.3 Working towards Collaboration

Every stakeholder has their own requirements for the paint editor. Through this phase we have identified the common and conflicting requirements of the stakeholders. We tried to resolve the conflicting requirements and finalize the requirements by keeping the stakeholders' priorities.

**Common Requirements:**

- ❑ User will have to register first in order to use the paint editor.
- ❑ Members will be authenticated before logging in to the homepage.
- ❑ User can save the picture in the gallery.
- ❑ User can edit and download picture.
- ❑ Any user can search other users and view their pictures.
- ❑ Any user can give like and make a comment in other user's pictures.
- ❑ User can comment in her own picture.

**Conflicting Requirements:**

- ❑ One user can report another user.
- ❑ Admin can view user information.
- ❑ Admin can ban any user.
- ❑ User can make album with drawn pictures.
- ❑ Canvas will contain a color palette.

**Final Requirements:**

- ❑ User will have to register first in order to use the paint editor.
- ❑ Members will be authenticated before logging in to the homepage.
- ❑ Each user will have their personal account.
- ❑ User can draw and paint using the canvas.
- ❑ User can save the picture in the gallery.
- ❑ User can edit and download picture.
- ❑ Any user can search other users and view their pictures.
- ❑ Any user can give like and make a comment in other user's pictures.
- ❑ User can comment in her own picture.
- ❑ Admin will have the same user interface with one additional facility to view other users' information.
- ❑ One user can report another user.
- ❑ Admin can ban any user if there is a report about the user.

## 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 in section 2.1.1. 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 has helped us to develop the basic understanding of our project. Through the stages of inception we identified the stakeholders and their view of the project. Inception has helped us to effectively communicate with the stakeholders.

# CHAPTER THREE: ELICITATION OF PIICHKARI- ONLINE PAINT EDITOR

## 3.1 Quality Function Deployment

Quality function deployment (QFD) is the translation of user requirements and requests into product designs. The goal of QFD is to build a product that does exactly what the customer wants instead of delivering a product that emphasizes expertise the builder already has. It is a method for maximizing customer satisfaction from the software engineering process. Through this phase we have identified the normal, expected and exciting requirements 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 normal requirements are given below:

1. Members/admins can save the pictures individually.
2. A member can see other member's pictures.
3. One member can like and comment in other member's picture.
4. Members/admins can comment on their own picture.
5. A member can download her own picture.
6. Any user who wishes to draw or search will have to become a member first. They will be provided a registration form.
7. Admin can view the details of members but cannot modify them.
8. Member/admin can edit her pictures.
9. Members/admins can search others.

### 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. These are written below:

1. Members/admins can draw picture in the editor.
2. Member/admin can change her personal information.
3. Each picture is public.

### 3.1.3 Exciting Requirements

1. Members can report other members.
2. Admin can ban/unban any member but cannot modify any personal information of the members.

## 3.2 Usage Scenario

Piichkari is a painting based web application with the feature of interaction between users. The system is divided into five sub systems.

### 3.2.1 Homepage

Homepage will contain authentication and registration option.

Registration

Users have to register for using the software. In the registration form, user needs to provide an name, an email address and a password (Password must be between 8-16 characters). Given email address will be checked for validation in database. If the email address has been used before for other account, then it will prompt the user requesting for providing another one. After getting proper information, an account will be created.

Authentication

Members/admins can enter the email address and password for using the website. If the members/admins enter an invalid email then the system will show “Invalid Credential”. If the password does not match, then the system will show wrong password. Then members/admins can try again or select forget password option. If members/admins select “forget password” option, then the system will ask for an email address. After getting the address, system will send an email containing a link to let the member/admin to change the password.

### 3.2.2 Personal Account

Members/Admins will have their personal accounts. The account will contain members’/admins’ personal information- name, email address and password. Member/Admin can view previously drawn images from the gallery and can also go to canvas to draw new picture. Members/Admins can modify their name and password. To change name or password, they will have to provide old password. They can log out from their accounts after finishing their activities. One can delete his/her account permanently and it will be deleted from the database.

### 3.2.3 Admin Panel

There will be two different roles of users. One is admin role and another is member role. Admins will have an interface similar to normal members. Only difference is they can view every member’s/admin’s information. Admins cannot modify them. Admins can ban a member if any other member reports about him/her. Admins can log out after finishing their activities.

### 3.2.4 Canvas

There will a canvas for drawing and painting. Member/admin can use a color palette containing various colors and paint brushes with different sizes (small, medium and large) to paint on the given canvas. Member/admin can have a pencil and eraser to draw. Member/admin can save

their pictures in their accounts. They will have to provide a name in order to save as image. They can download them later. They can also edit their saved pictures.

### 3.2.5 Search

Members/Admins can search others. They can search using their names and system will search for it database. When the member/admin types letters for searching, suggestions will show up according to the letters combination. Then they can visit their accounts and visit the gallery of the pictures that the member/admin has drawn. They can open any image. The image will contain options including like and comment option. Member/admin can see who has given like or who has commented in her picture. They can see the comments but cannot reply to the comment. They can also comment on their own pictures. Member/admin can see the total number of likes a picture has got.

# CHAPTER FOUR: SCENARIO BASED MODEL OF PIICHKARI- ONLINE PAINT EDITOR

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. [2]

## 4.1 Definition of Use Case

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The use case should contain all system activities that have significance to the users. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.

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.

**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

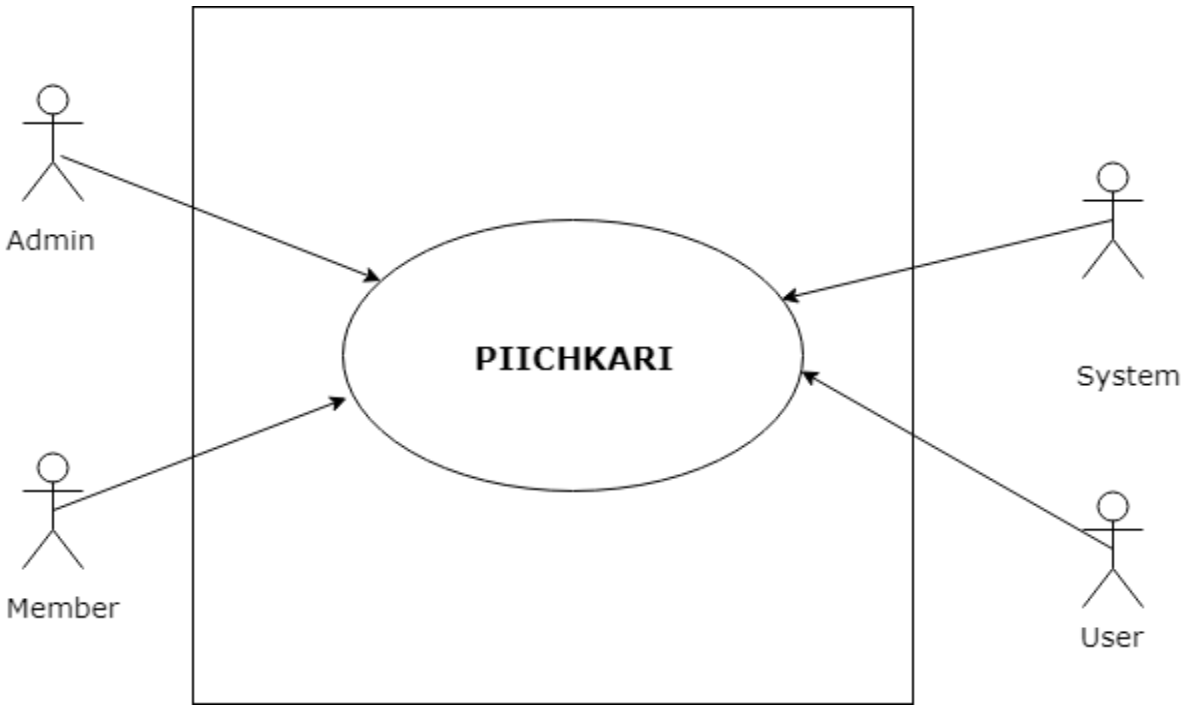


*Figure 1 Level-0 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-0** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System, User** |

**Description of Level-0 Use Case Diagram**

We identified the four actors who will interact with the system. The actors are categorized in to primary and secondary actors. Primary actors perform some actions and get a reply from the system. Secondary actors either produce or consume information.

The actors are written below:

1. Admin (Primary)
2. Member (Primary)
3. User (Secondary)
4. System (Secondary)

### 4.2.2 Level-1 Use Case Diagram

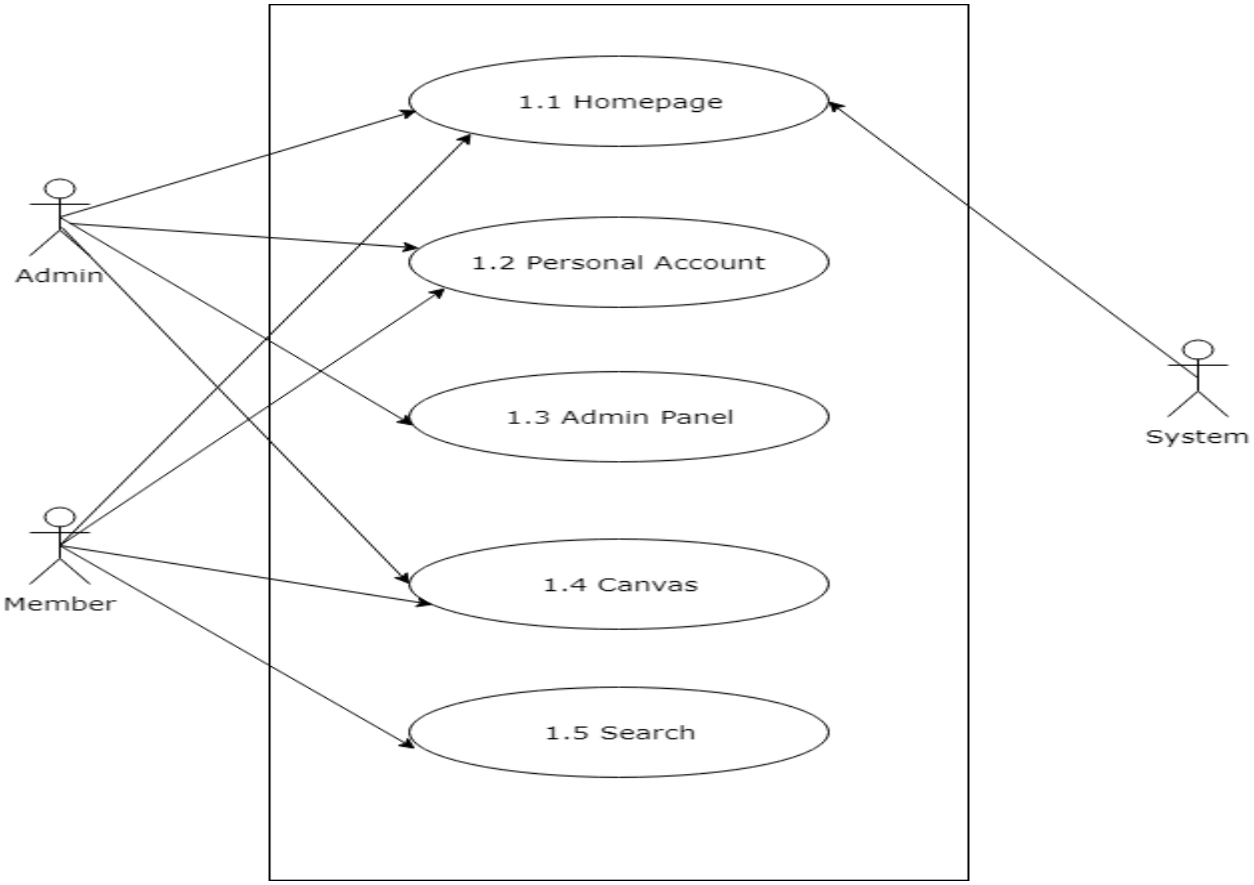


*Figure 2 Level-1 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1 Use Case Diagram**

There are five subsystems in PIICHKARI. They are:

1. Homepage
2. Personal account
3. Admin panel

4. Canvas
5. Search

### 4.2.3 Level-1.1 Use Case Diagram- Homepage


*Figure 3 Level-1.1 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.1** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.1 Use Case Diagram**

Homepage will contain two options for the user.

1. Authentication
2. Registration

### 4.2.4 Level-1.1.1 Use Case Diagram- Registration

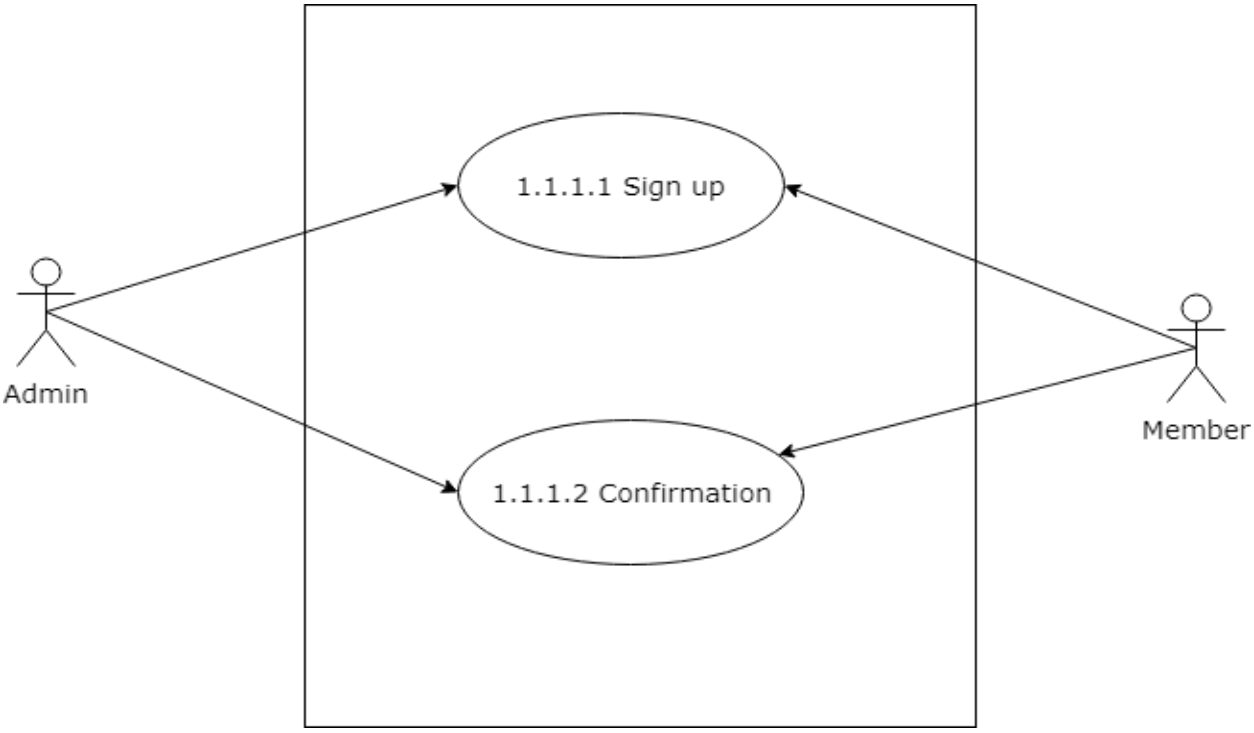


*Figure 4 Level-1.1.1 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.1.1** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.1.1 Use Case Diagram- Registration**

Users have to register for using the painting tool. In the registration form, user needs to provide an name, an email address and a password (Password must be between 8-16 characters). Given email address will be checked for validation in database. If the email address has been used before for other account, then it will prompt the user requesting for providing another one. After getting proper information, an account will be created.

**Action-Reply of Level-1.1.1 Use Case Diagram**

**Action 1:** User will provide name, email address and password for registration.

**Reply 1:** Given email address will be checked for validation. After validation, account will be created and confirmation email will be sent.

### 4.2.5 Level-1.1.2 Use Case Diagram- Authentication

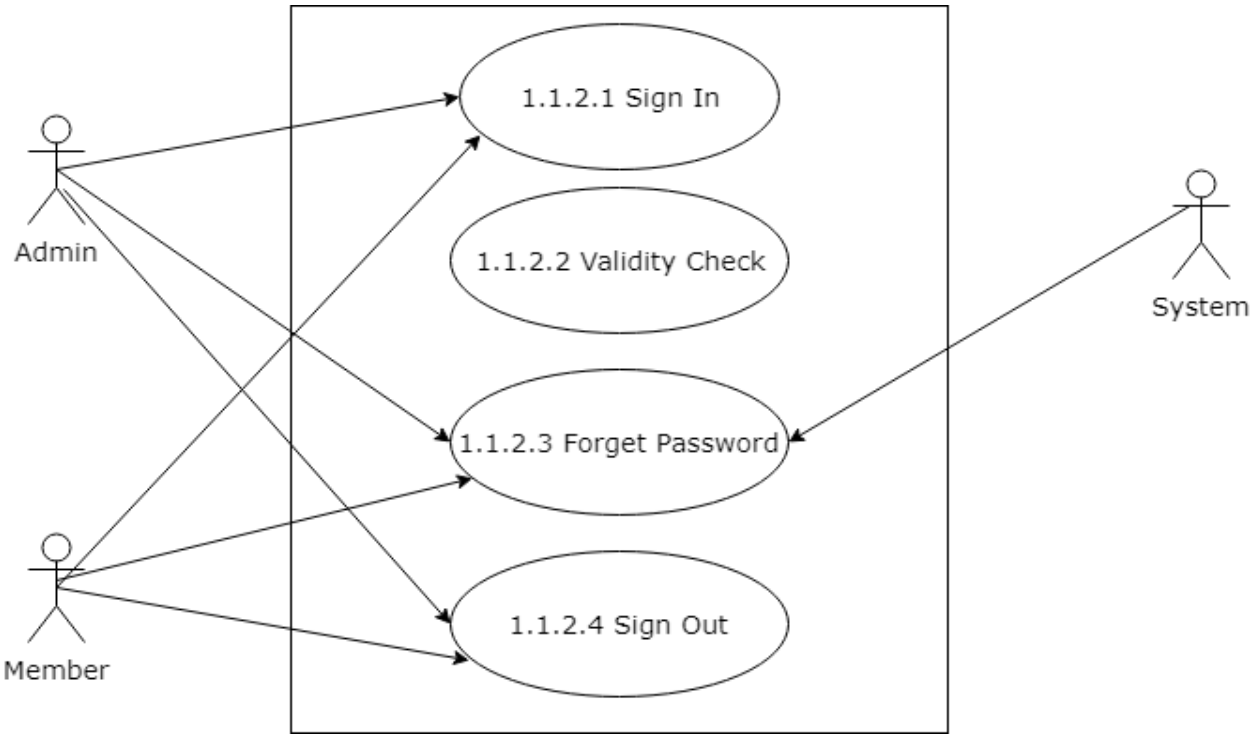


*Figure 5 Level-1.1.2 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.1.2** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.1.2 Use Case Diagram- Authentication**

Member/admin can enter the email address and password for using the website. If the entered email an invalid email then the system will show “Invalid Credential”. If the password does not match, then the system will show wrong password. Then user can try again or select forget password option. If user selects “forget password” option, then the system will ask for an email address. After getting the address, system will send an email containing a link to let the user to change the password.

**Action-Reply of Level-1.1.1 Use Case Diagram**

- **Action 1:** Member/admin will provide email and password to log in to the system.
  **Reply 1:** Email and password will be checked for validation. After that, member/admin will be logged in to the system.
- **Action 2:** Member/admin selects the forget password option.
  **Reply 2:** Member/admin will be asked for an email address. After getting the address, system will send an email containing a link to let the user to change the password.

### 4.2.6 Level-1.2 Use Case Diagram- Personal Account

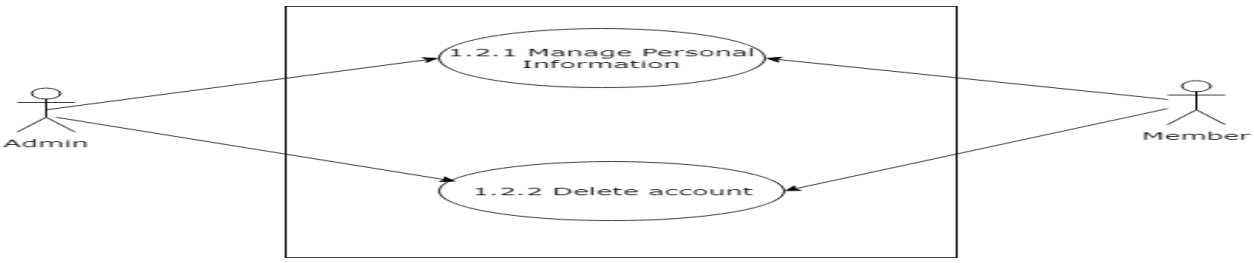


*Figure 6 Level-1.2 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.2** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.2 Use Case Diagram- Personal Account**

The subsystem personal account consists of two options for the member/admin.

1. Manage personal information
2. Delete account

Member/admin will have her personal accounts. The account will contain user's personal information- name, email address and password. Member/admin can view previously drawn images from the gallery and can also go to canvas to draw new picture. Member/admin can modify their name and password. To change name or password member/admin will have to provide old password. They can log out from their accounts after finishing their activities. One can delete his/her account permanently and it will be deleted from the database. One member can report other member.

**Action-Reply of Level-1.2 Use Case Diagram**

- **Action 1:** Member/admin can edit her personal information.

  **Reply 1:** System will ask for old password. After validation, modified information will be updated.

- **Action 2:** Member/admin can delete her account.

  **Reply 2:** Account will be deleted from the system.

- **Action 3:** Member/admin can view her previously drawn images.

  **Reply 3:** System will provide interface for that.

- **Action 4:** Member reports another member.

  **Reply 4:** Reported member is banned by admin.

### 4.2.7 Level-1.3 Use Case Diagram- Admin Panel

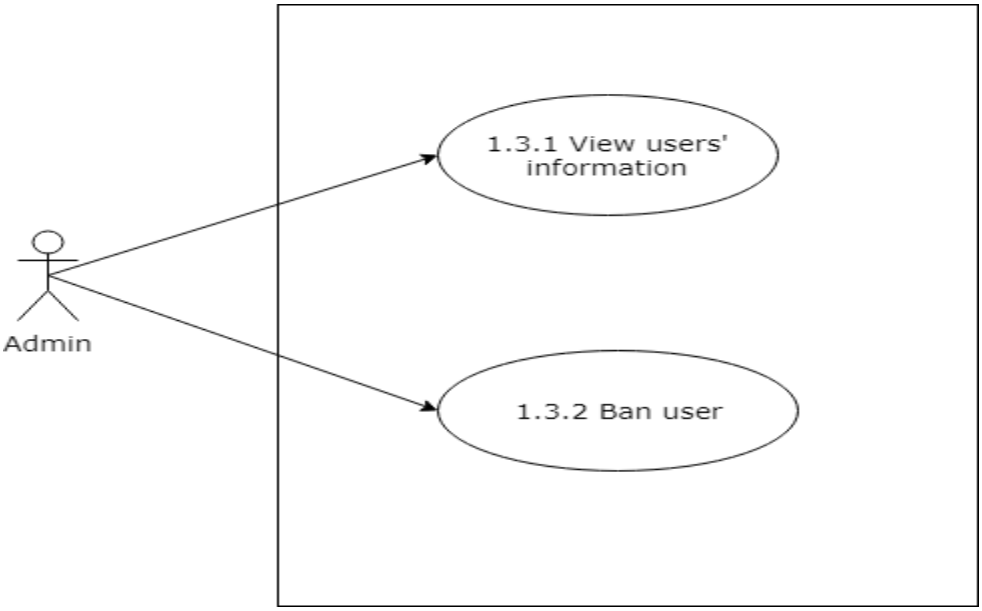


*Figure 7 Level-1.3 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.3** |
| **Primary Actor** | **Admin** |
| **Secondary Actor** | **System** |

**Description of Level-1.3 Use Case Diagram- Admin Panel**

There will be two different roles of users. One is admin role and another is member role. Admins will have an interface similar to normal users. Only difference is they can view every user's information. Admins cannot modify them. Admins can ban a member if any other member reports about her.

**Action-Reply of Level-1.3 Use Case Diagram**

- **Action 1:** Admin can view other member's personal information.

  **Reply 1:** Information will be shown in read only mode.
- **Action 2:** Admin can ban a member if she is reported by another member.

  **Reply 2:** Banned member's account will be removed.

### 4.2.8 Level-1.4 Use Case Diagram- Canvas

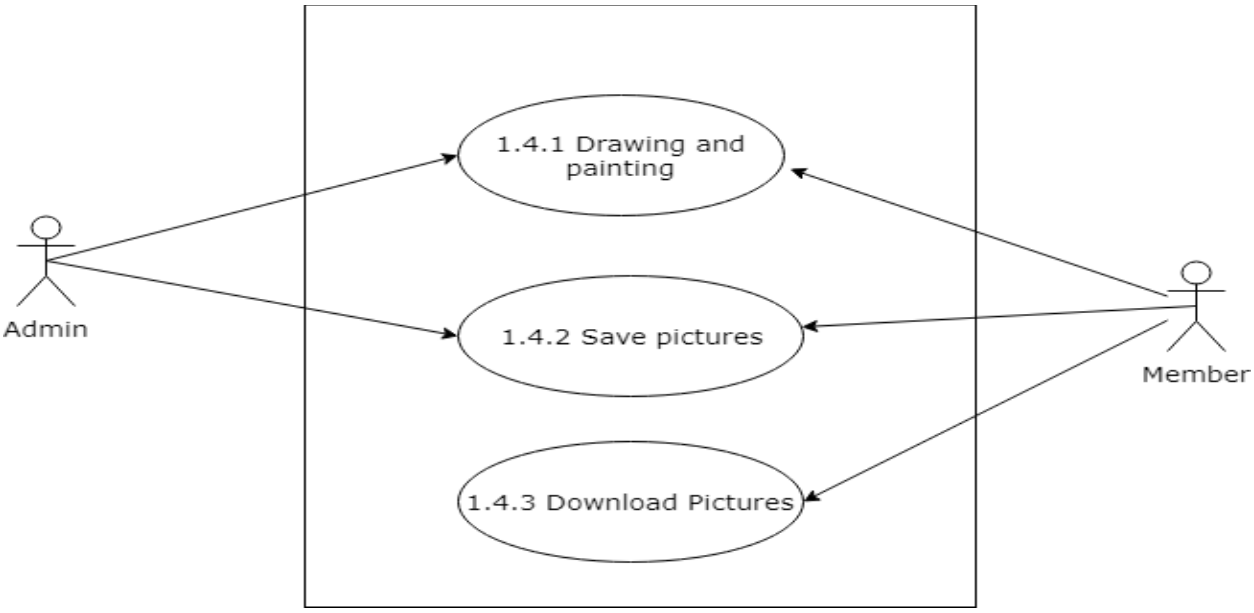


*Figure 8 Figure 8 Level-1.4 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.4** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.4 Use Case Diagram- Canvas**

There will a canvas for drawing and painting. Member/admin can use a color palette containing various colors and paint brushes with different sizes to paint on the given canvas. Member/admin can have a pencil and eraser to draw. Member/admin can save their pictures in their accounts.

They will have to provide a name in order to save as image. They can download them later. They can also edit their saved pictures.

**Action-Reply of Level-1.4 Use Case Diagram**

- **Action 1:** Member/admin can draw and paint in canvas.

  **Reply 1:** Drawn pictures will be shown in canvas.
- **Action 2:** Member/admin can save the picture.

  **Reply 2:** Picture is saved as image.
- **Action 3:** Member/admin can download the image.

  **Reply 3:** Image will be downloaded, member will have to give a name to the image before downloading.
- **Action 4:** Member/admin can edit previously drawn images.

  **Reply 4:** Image will be converted to picture and will be opened in the canvas.

### 4.2.9 Level-1.5 Use Case Diagram- Search

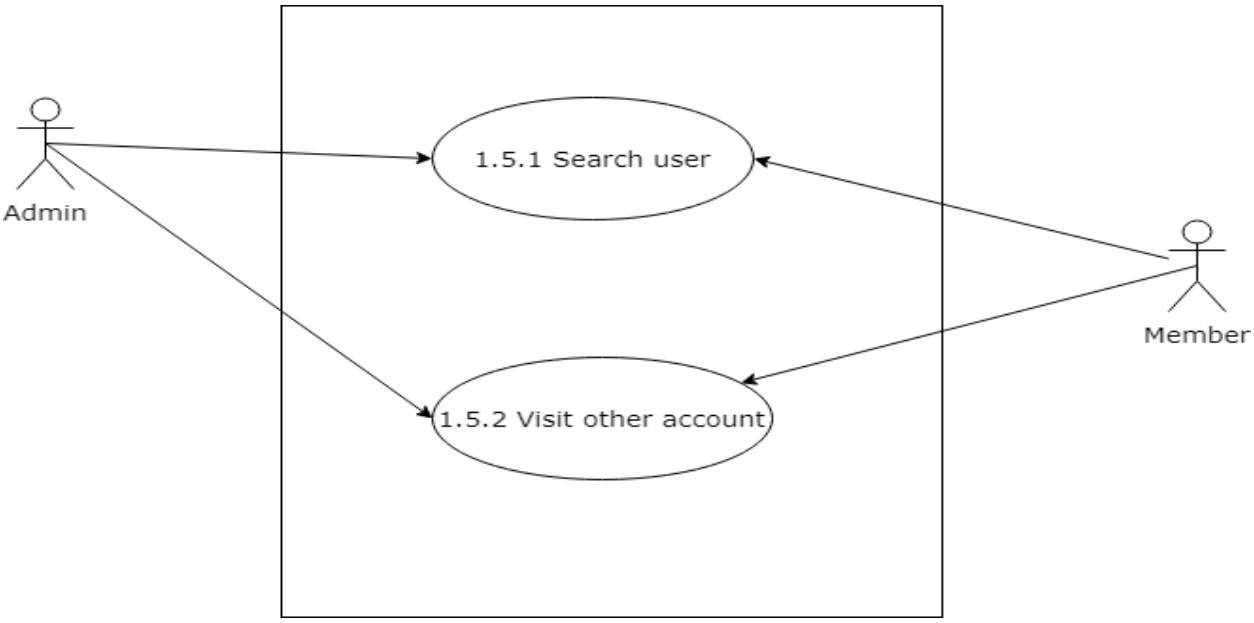


*Figure 9 Level-1.5 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.5** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.5 Use Case Diagram- Search**

Members/admin can search other users. They can search using their names and system will search for it in the database. When the member types letters for searching, suggestions will show up according to the letters combination. Then they can visit their accounts and visit the gallery of the pictures that the user has drawn.

**Action-Reply of Level-1.5 Use Case Diagram**

- **Action 1:** Member/admin searches other members using their names.

  **Reply 1:** Suggestions of name will be shown according to letter combination and selected member's profile will be shown.

### 4.2.10 Level-1.5.2 Use Case Diagram- Visit other member's account

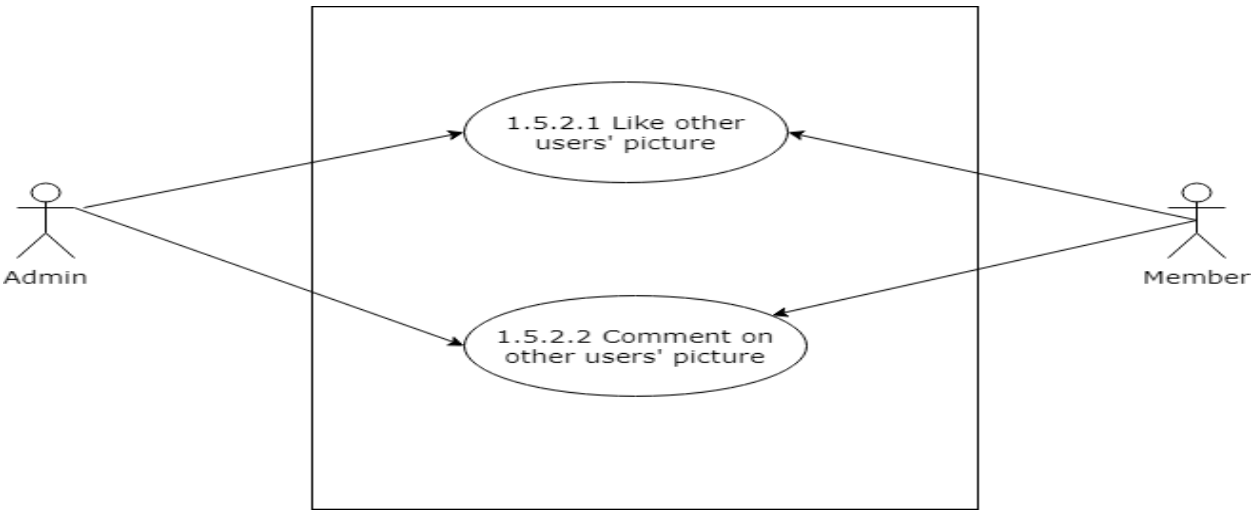


*Figure 10 Level-1.5.2 Use Case Diagram*

| | |
|---|---|
| **Name** | **PIICHKARI** |
| **ID** | **PIICHKARI-L-1.5.2** |
| **Primary Actor** | **Admin, Member** |
| **Secondary Actor** | **System** |

**Description of Level-1.5.2 Use Case Diagram- Visit other member's account**

Member/admin can visit other member's account and view the gallery. They can view any image and the image will contain options including like and comment option. Member/admin can see who has given like or who has commented in her picture. Member/admin can see the comments but can not reply to the comment. They can also comment on their own pictures. Member/admin can see the total number of likes a picture has got.

**Action-Reply of Level-1.5.2 Use Case Diagram**

- **Action 1:** Member/admin can view other member's profile.

  **Reply 1:** Member's gallery will be shown in her profile.
- **Action 2:** Member/admin can give likes and also comment in an image.

  **Reply 2:** Number of likes will be increased and comments will be added.

## 4.3 Activity Diagrams

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

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

### 4.3.1 Activity Diagram of Registration

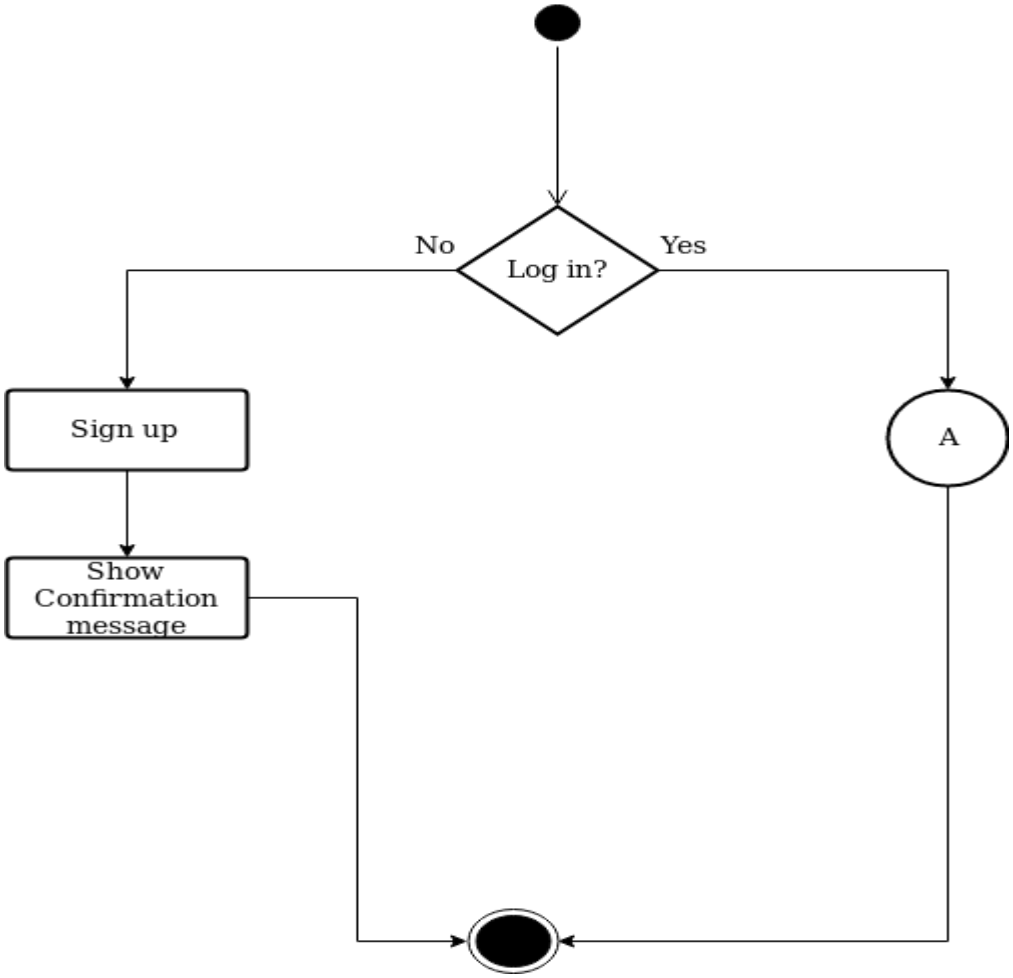


*Figure 11 Activity Diagram of Registration*

### 4.3.2 Activity Diagram of Authentication

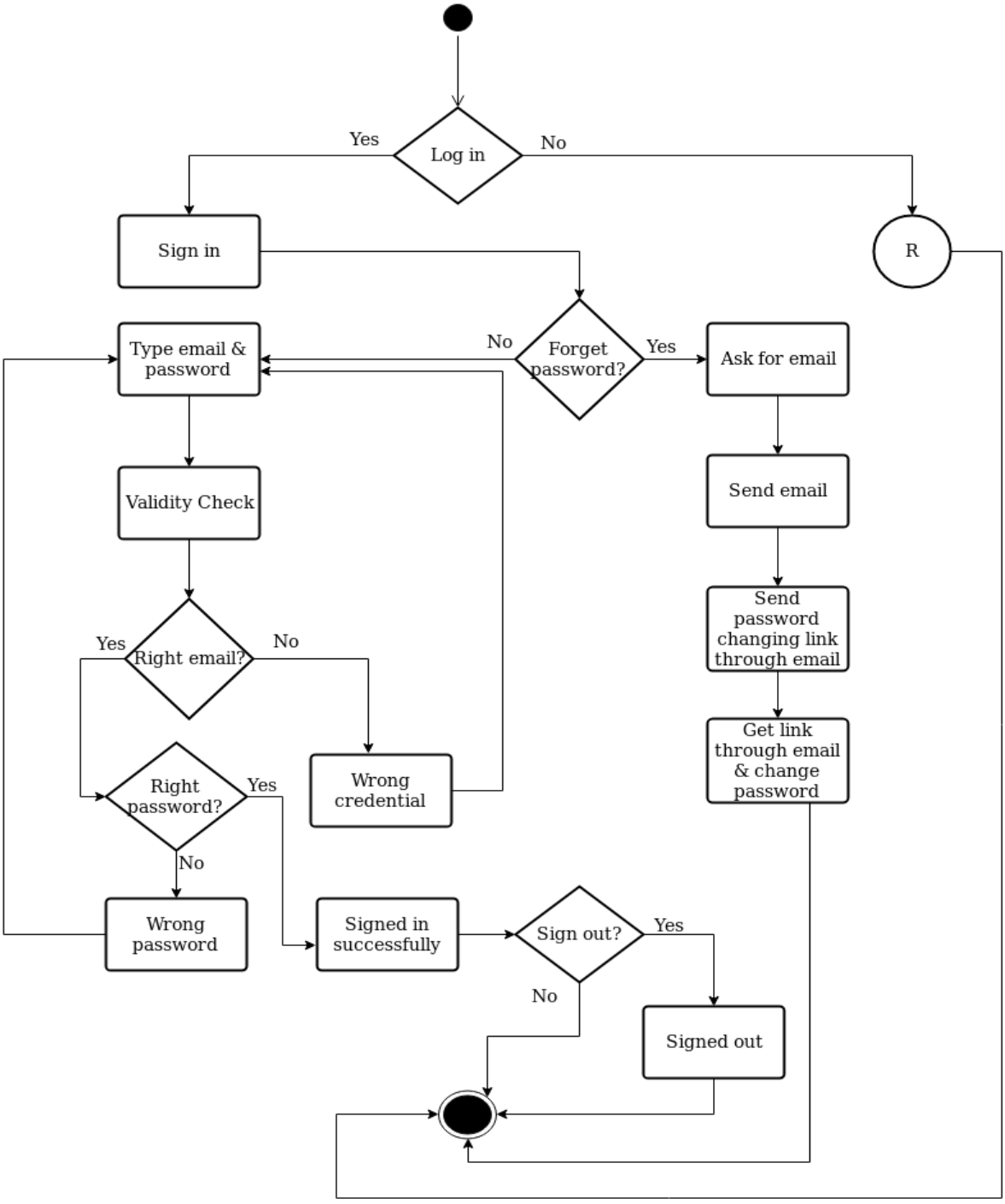


*Figure 12 Activity Diagram of Authentication*

### 4.3.3 Activity Diagram of Personal Account

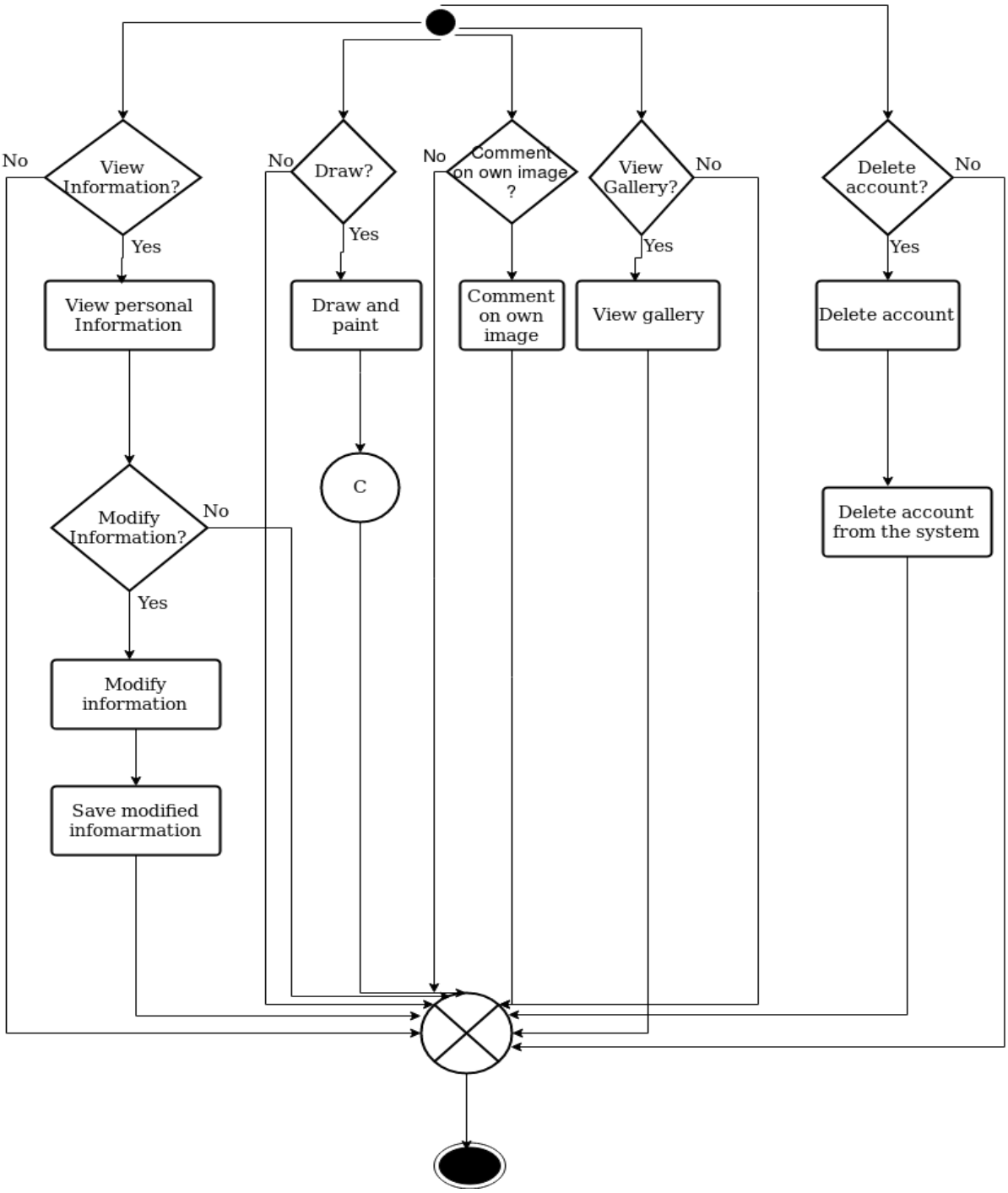


*Figure 13 Activity Diagram of Personal Account*

### 4.3.4 Activity Diagram of Admin Panel

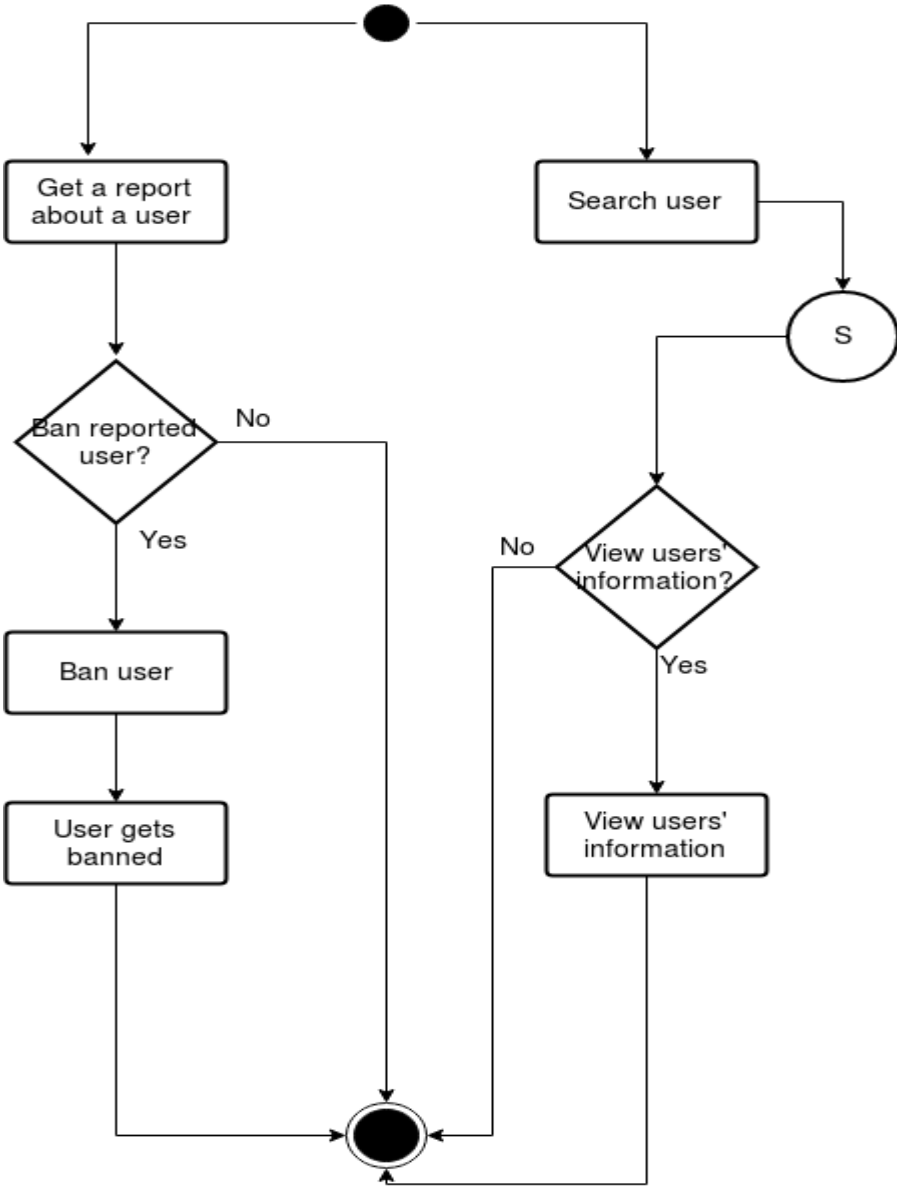


*Figure 14 Activity Diagram of Admin Panel*

### 4.3.5 Activity Diagram of Canvas

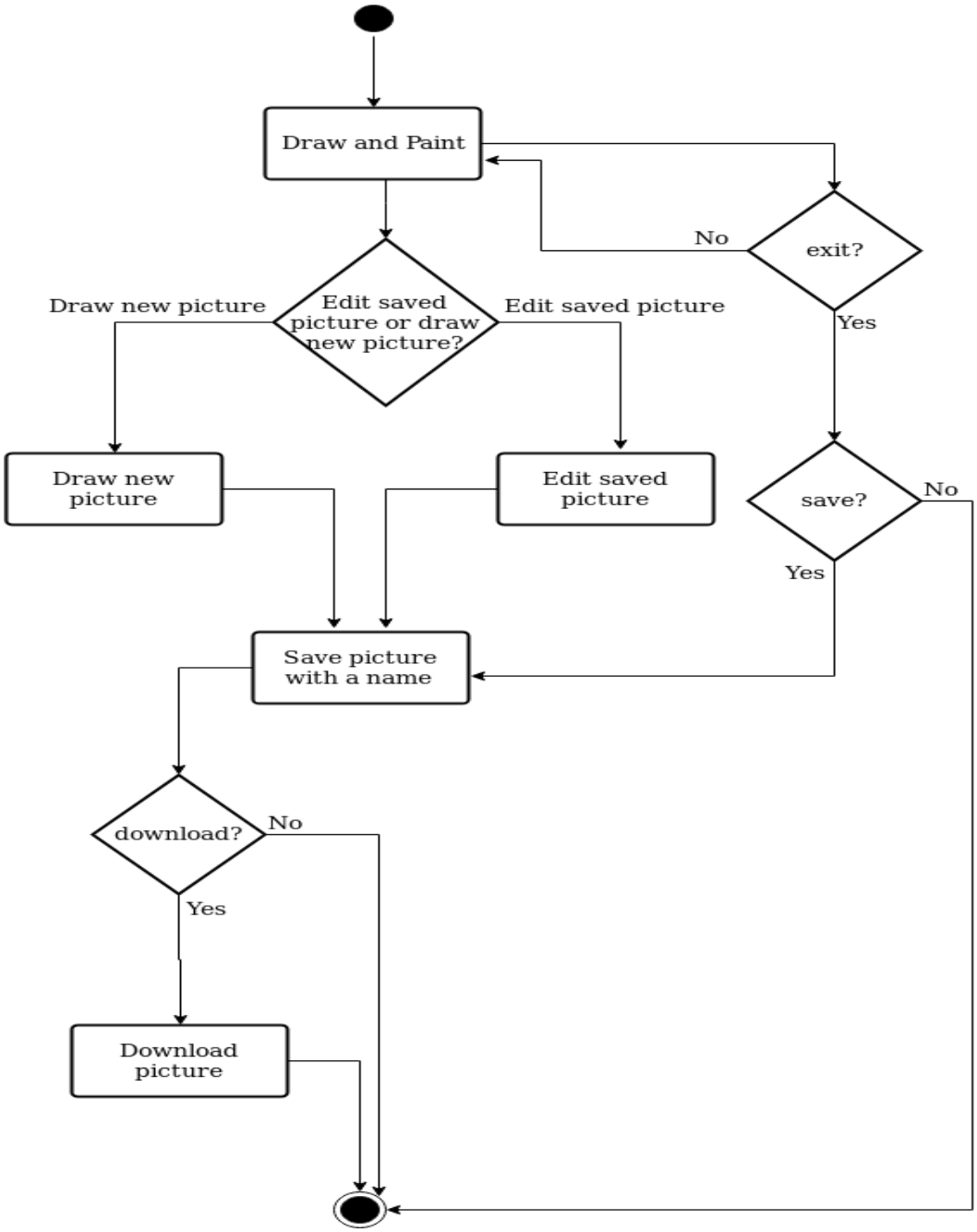


*Figure 15 Activity Diagram of Canvas*

### 4.3.7 Activity Diagram of Search

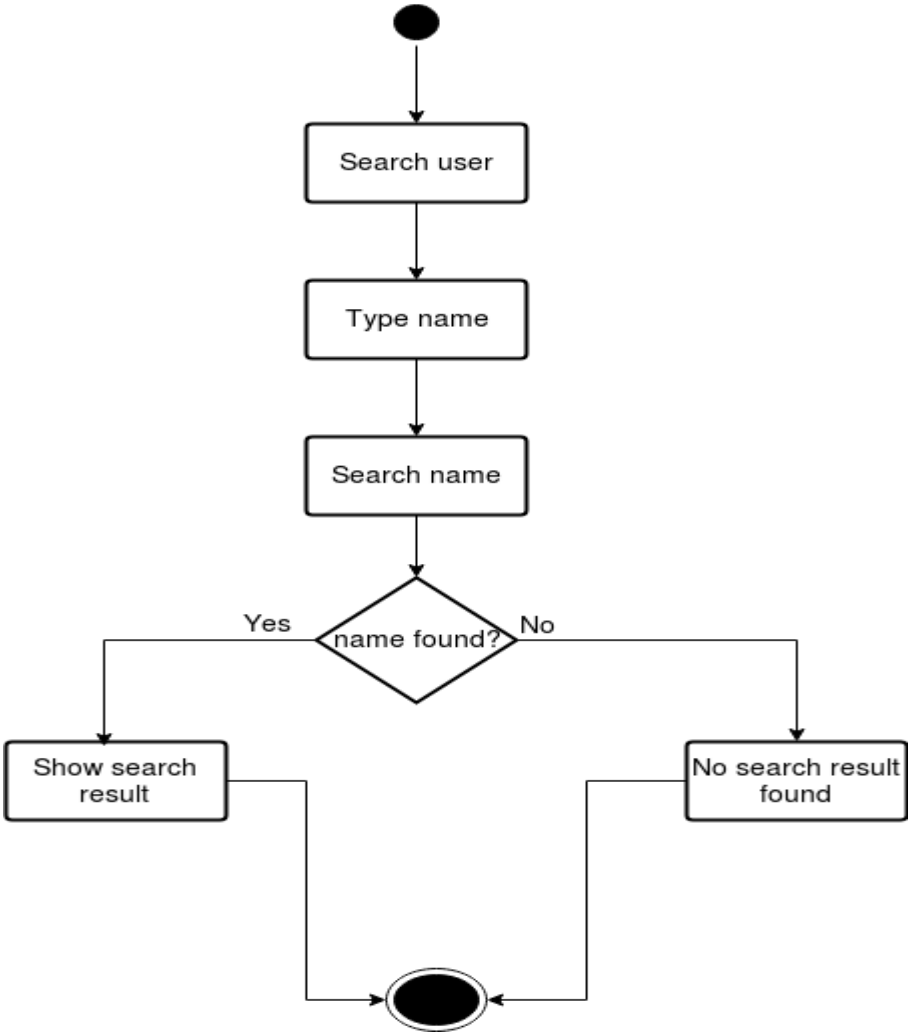


*Figure 16 Activity Diagram of Search*

### 4.3.6 Activity Diagram of Visit Other Member's Account

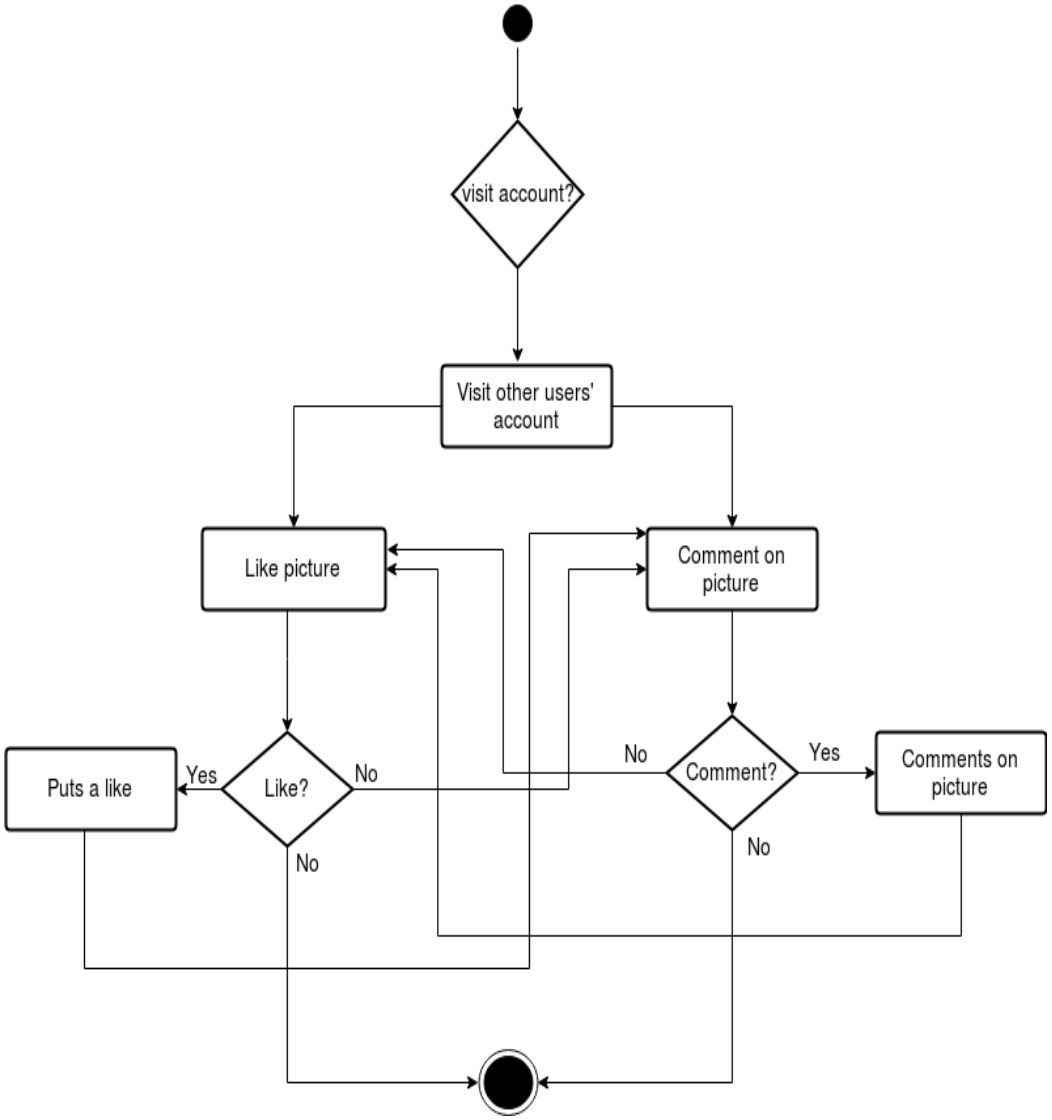


*Figure 17 Activity Diagram of Visit Other Member's Account*

## 4.4 Swimlane Diagrams

A swimlane 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 swimlane diagrams of the modules described in the previous chapter is shown below:

### 4.4.1 Swimlane Diagram of Registration

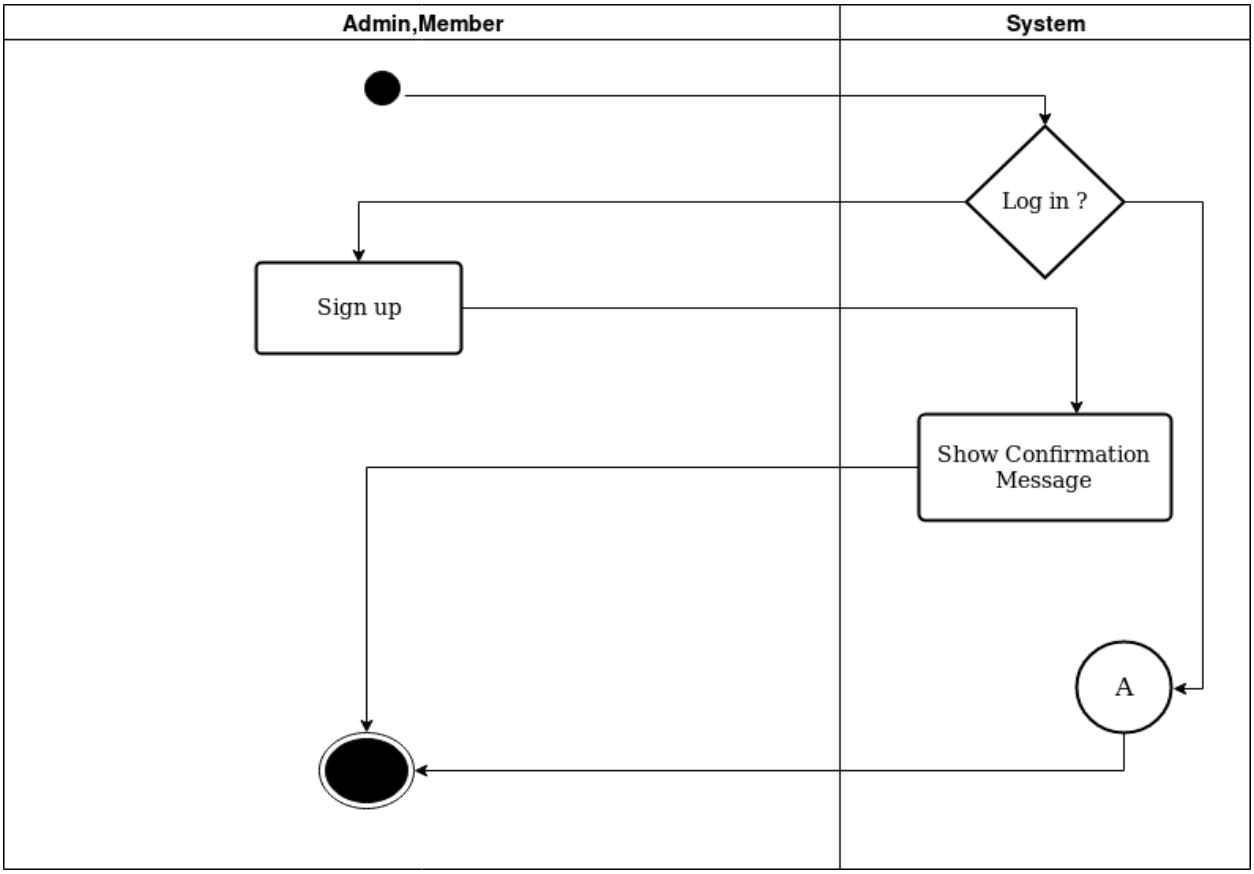


*Figure 18 Swimlane Diagram of Registration*

## 4.4.2 Swimlane Diagram of Authentication

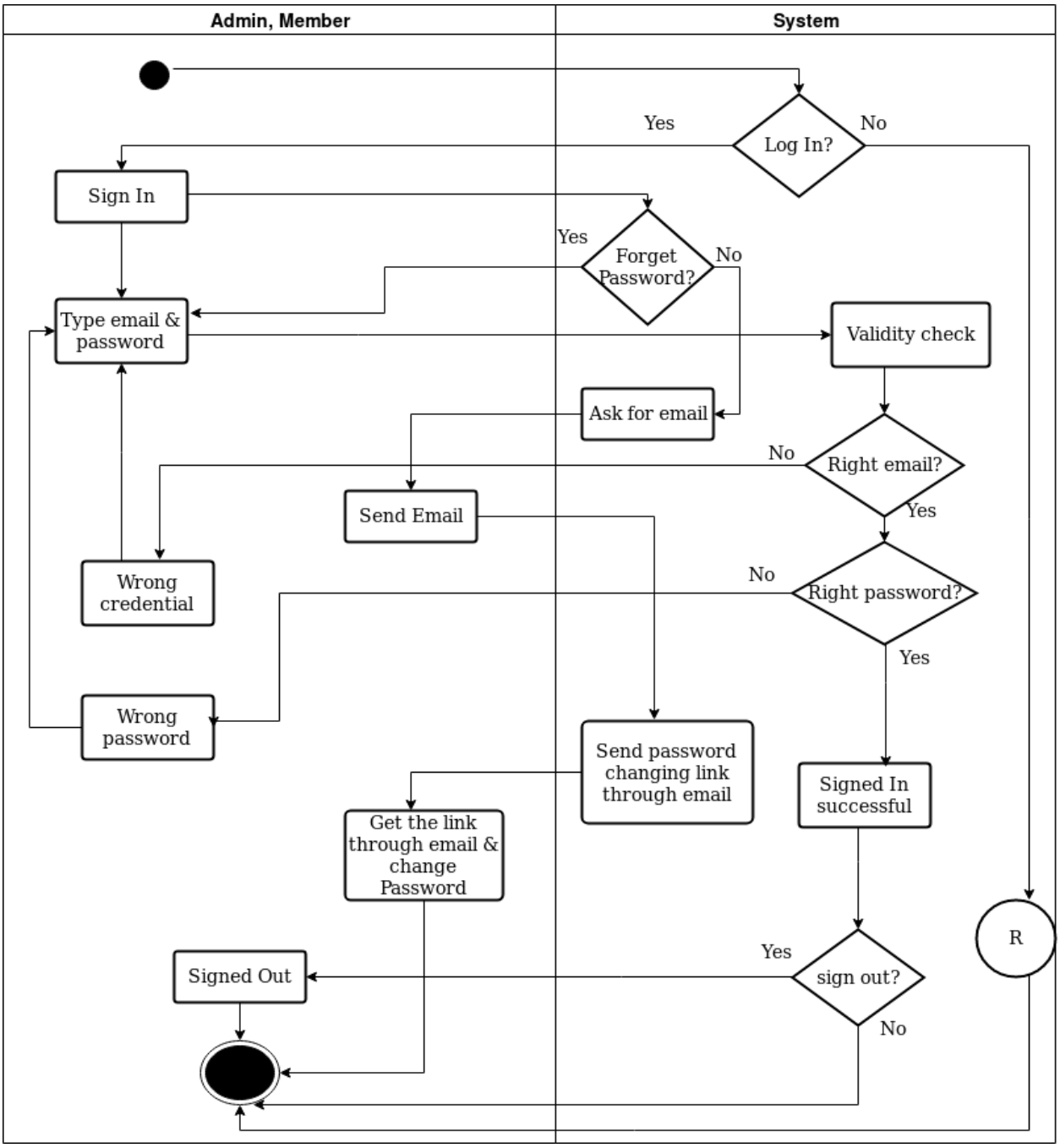


*Figure 19 Swimlane Diagram of Authentication*

### 4.4.3 Swimlane Diagram of Personal Account

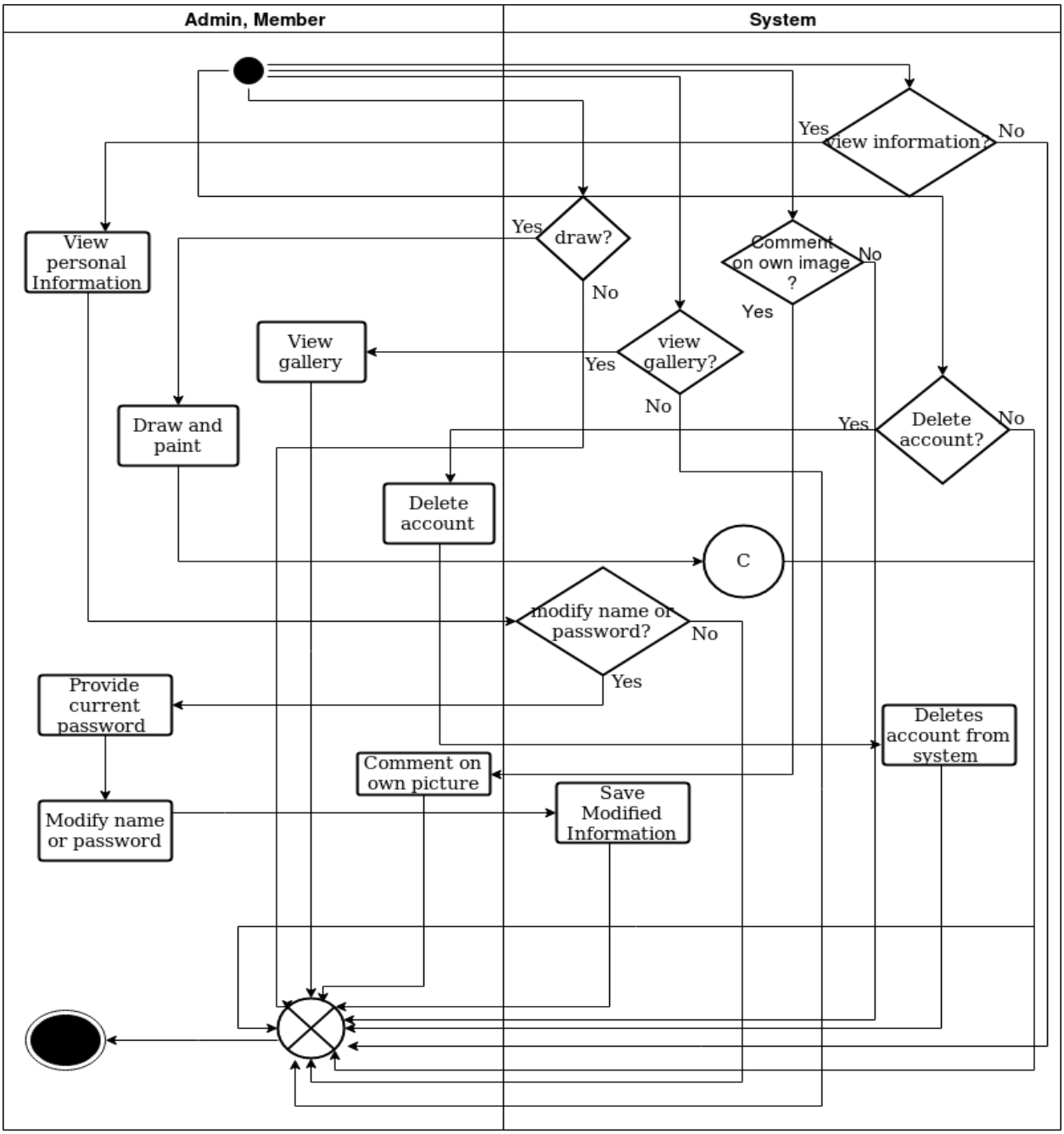


*Figure 20 Swimlane Diagram of Personal Account*

### 4.4.4 Swimlane Diagram of Admin Panel

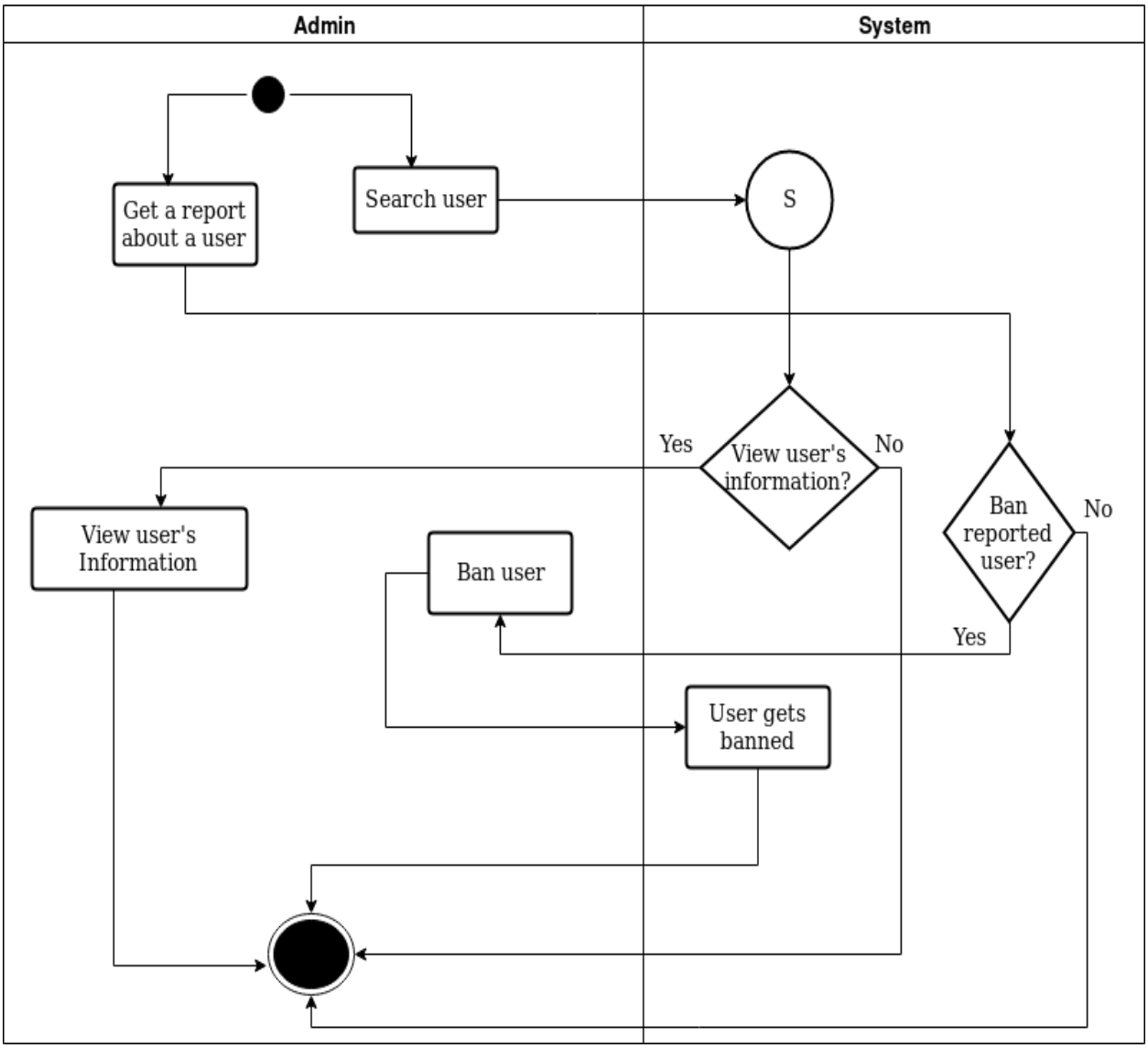


*Figure 21 Swimlane Diagram of Admin Panel*

### 4.4.5 Swimlane Diagram of Canvas

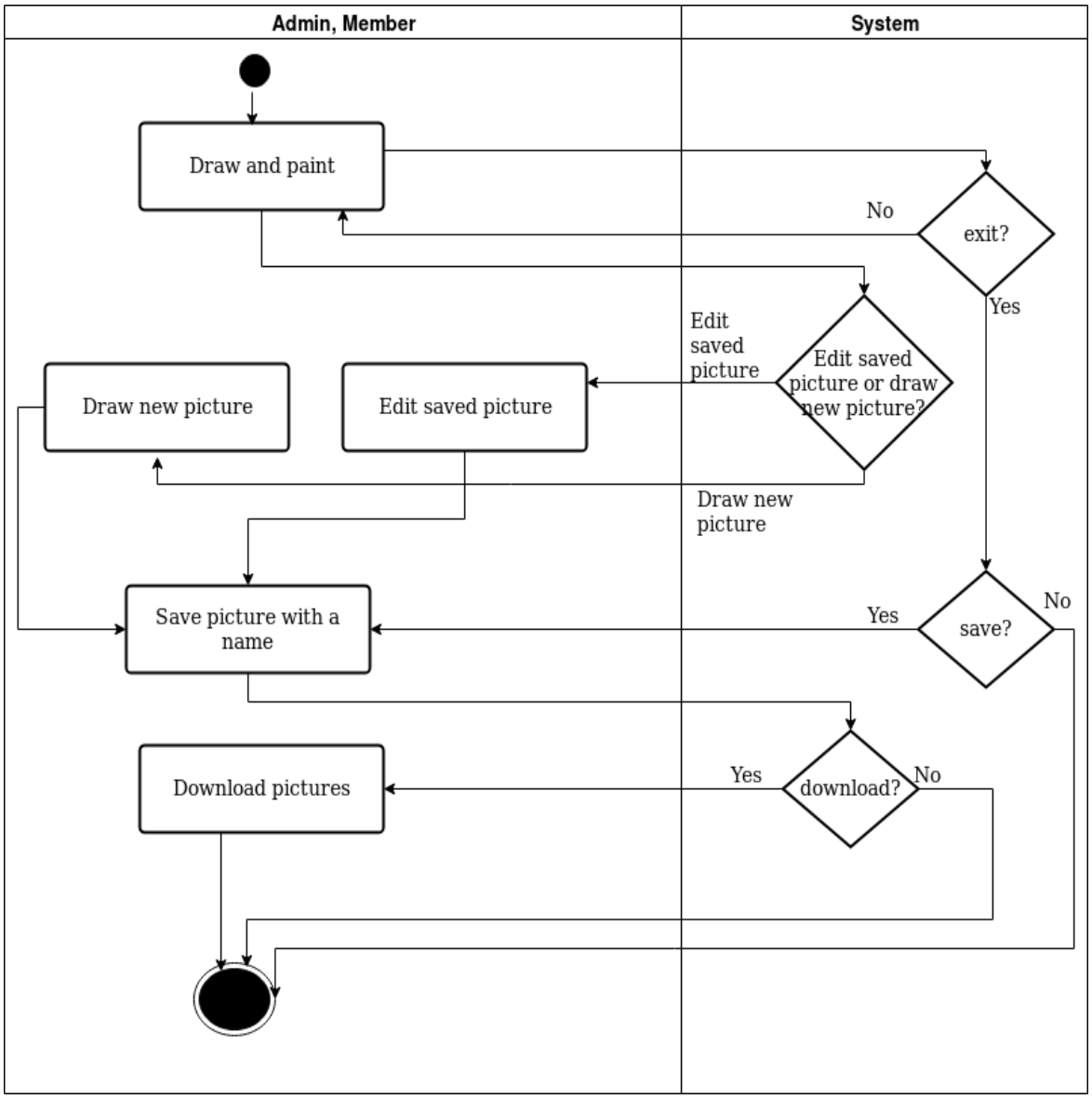


*Figure 22 Swimlane Diagram of Canvas*

### 4.4.6 Swimlane Diagram of Search

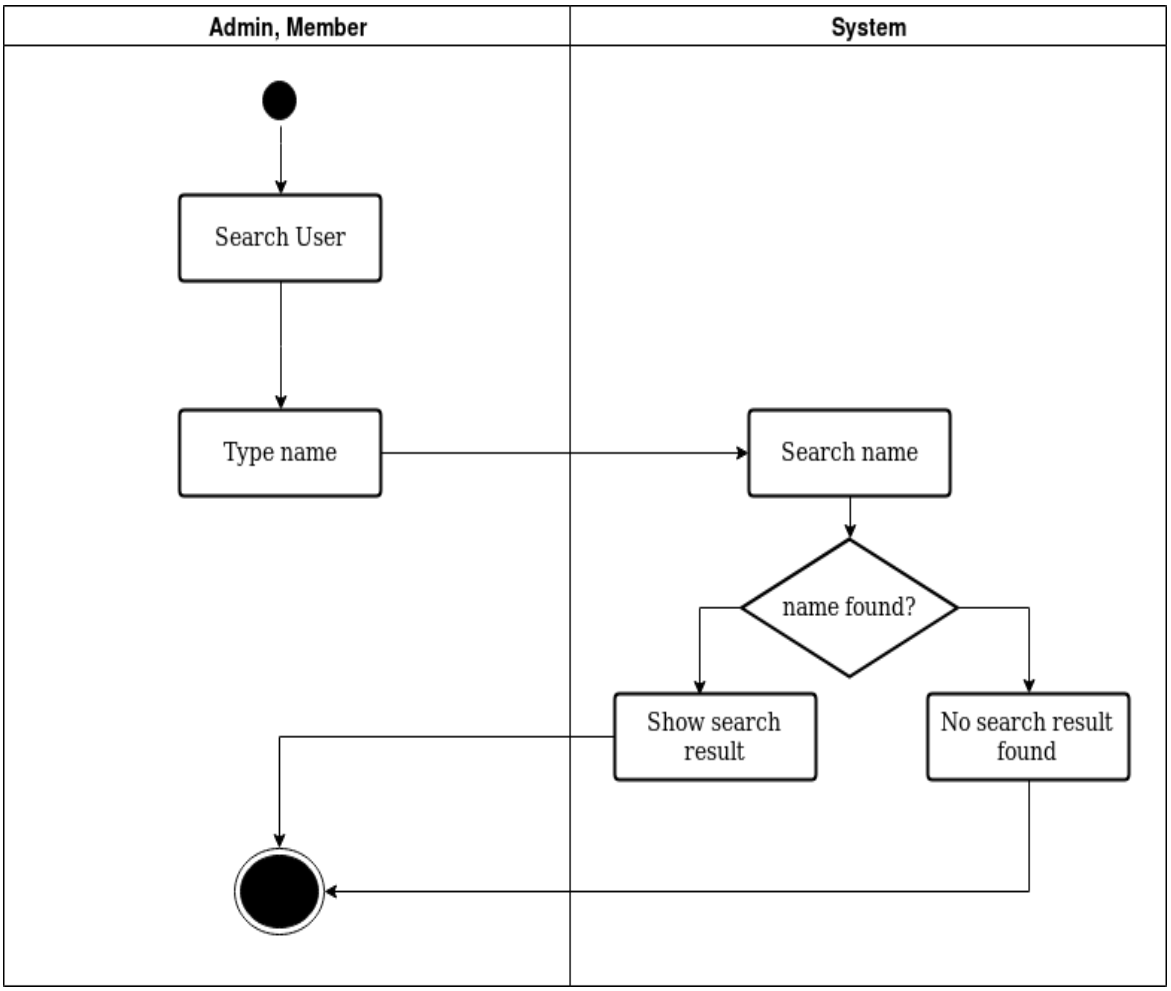


*Figure 23 Swimlane Diagram of Search*

### 4.4.7 Swimlane Diagram of Visit Other Member's Account

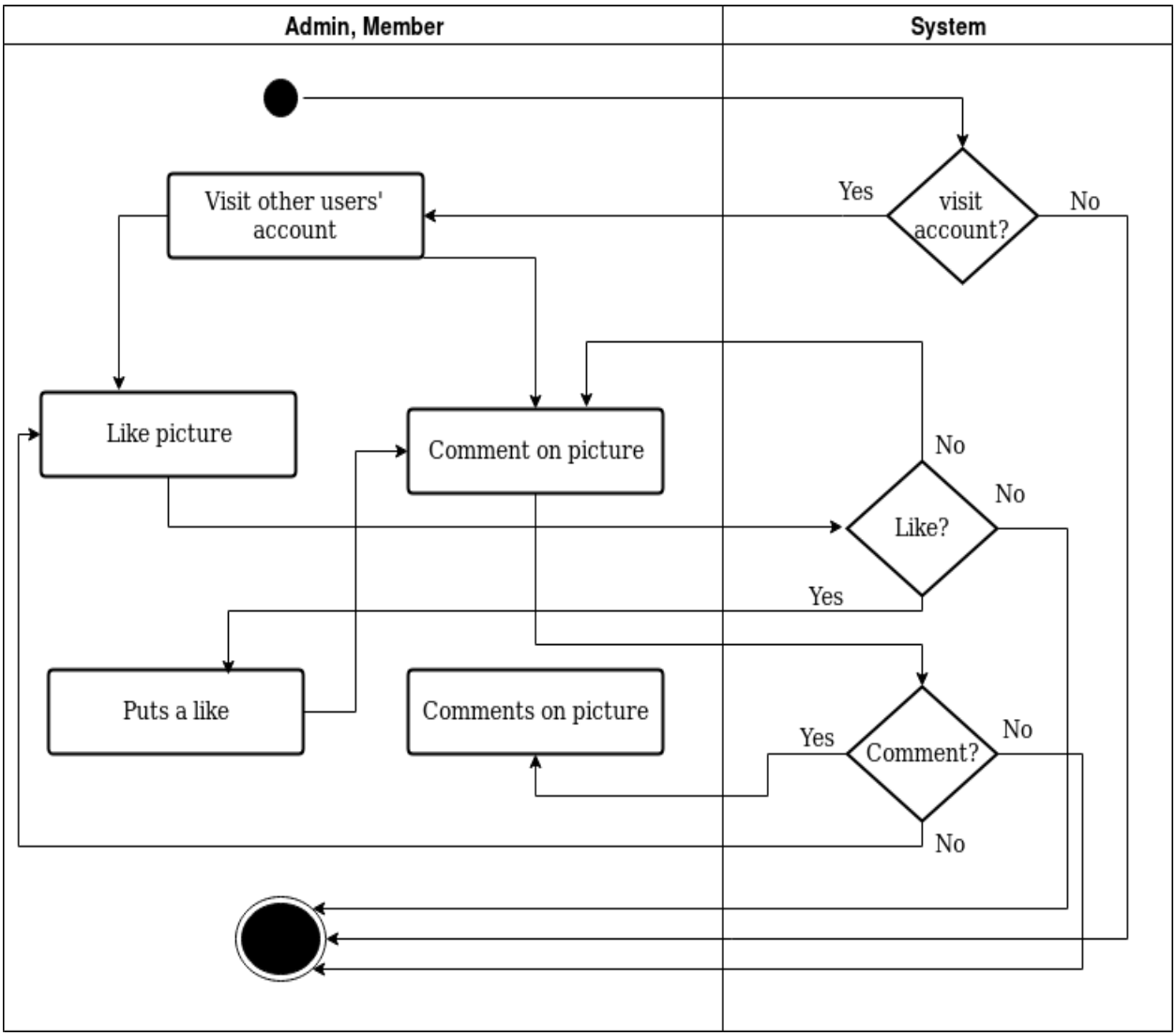


*Figure 24 Swimlane Diagram of Visit Other Member's Account*

# CHAPTER FIVE: DATA MODEL OF PIICHKARI- ONLINE PAINT EDITOR

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 Parsing*

| Serial No | Nouns | P/S | Attributes |
|---|---|---|---|
| 1 | Homepage | p | |
| 2 | Authentication | p | |
| 3 | Registration | p | |
| 4 | User | s | 8,9,10 |
| 5 | member | s | 8,9,10 |
| 6 | software | p | |
| 7 | registration form | p | |

| | | | |
|---|---|---|---|
| 8 | name | s | |
| 9 | email address | s | |
| 10 | password | s | |
| 11 | validation | p | |
| 12 | account | p | |
| 13 | information | p | |
| 14 | confirmation mail | p | |
| 15 | applicant's mail | p | |
| 16 | website | p | |
| 17 | invalid mail | p | |
| 18 | system | p | |
| 19 | invalid credential | p | |
| 20 | wrong password | p | |
| 21 | forget password | p | |
| 22 | link | p | |
| 23 | personal account | p | |
| 24 | images | s | 28 |
| 25 | gallery | p | |

| | | | |
|---|---|---|---|
| 26 | canvas | p | |
| 27 | picture | p | |
| 28 | image name | s | |
| 29 | old password | p | |
| 30 | role | s | |
| 31 | activities | p | |
| 32 | admin | s | 8,9,10 |
| 33 | interface | P | |
| 34 | normal user | p | |
| 35 | user's information | p | |
| 36 | reason | p | |
| 37 | drawing | p | |
| 38 | painting | p | |
| 39 | color palette | p | |
| 40 | colors | p | |
| 41 | paint brushes | p | |
| 42 | different sizes | p | |

| | | | |
|---|---|---|---|
| 43 | pencil | p | |
| 44 | eraser | p | |
| 45 | saved pictures | p | |
| 46 | letters | p | |
| 48 | suggestions | p | |
| 49 | options | p | |
| 50 | like | s | 52 |
| 51 | comment | s | |
| 52 | reply | p | |
| 53 | number of likes | s | |

## 5.2 Potential Data Objects

After applying grammatical parsing in the scenario, following potential data objects were identified.

1. **User:** Name, Email address, Password
2. **Member :** Name, Email address, Password
3. **Admin :** Name, Email address, Password
4. **Image :** Image name
5. **Like :** Number of likes

## 5.3 Analysis of Potential Data Objects

1. **User** object will need **user_id**. We need to know whether the user is an admin or a member.
2. **User**, **Admin, Member** have same attributes. So we can merge these objects into **User**.
3. **Image** will need **image_id** and **save_path**.
4. We can store like count in a relational table. So **like** object will not be needed later.
5. A new **Comment** object will need **comment_id** and **remark** attributes to store comment.
6. For distinguishing **Member** and **Admin** a **Role** table will be needed. The **Role** table will contain **role_id, role_name.**

## 5.4 Final Data Objects

The final data objects are given below-

*Table 2 Finalized Data Objects*

| **Serial no.** | **Data Object** | **Attributes** |
|---|---|---|
| 1 | User | user_id, name, email_address, password |
| 2 | Image | image_id, image_name, save_path |
| 3 | Comment | comment_id, remark |
| 4 | Role | role_id, role_name |

## 5.5 Data Object Relationship

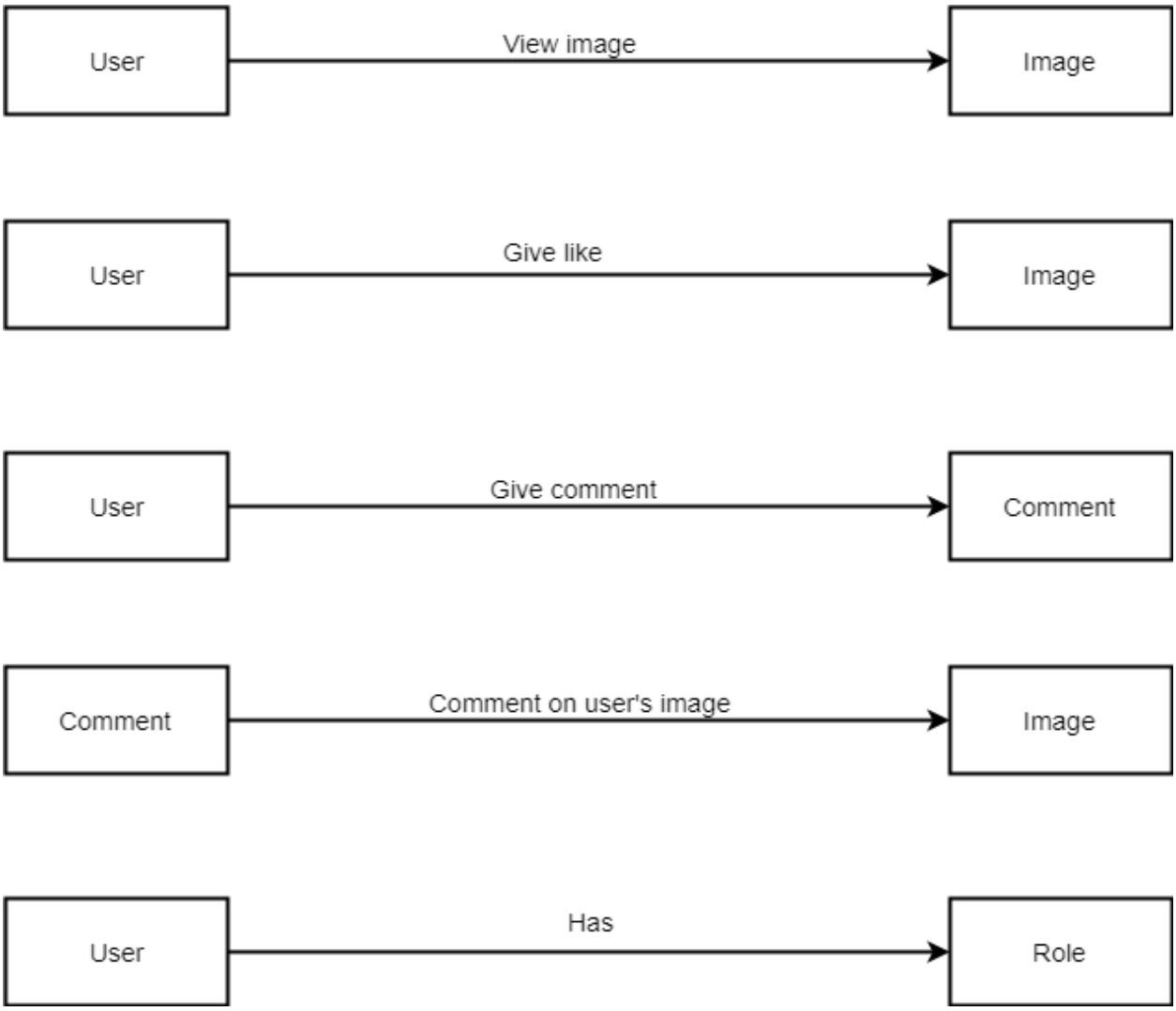


*Figure 25 Data Object Relationship*

## 5.6 Entity Relation Diagram

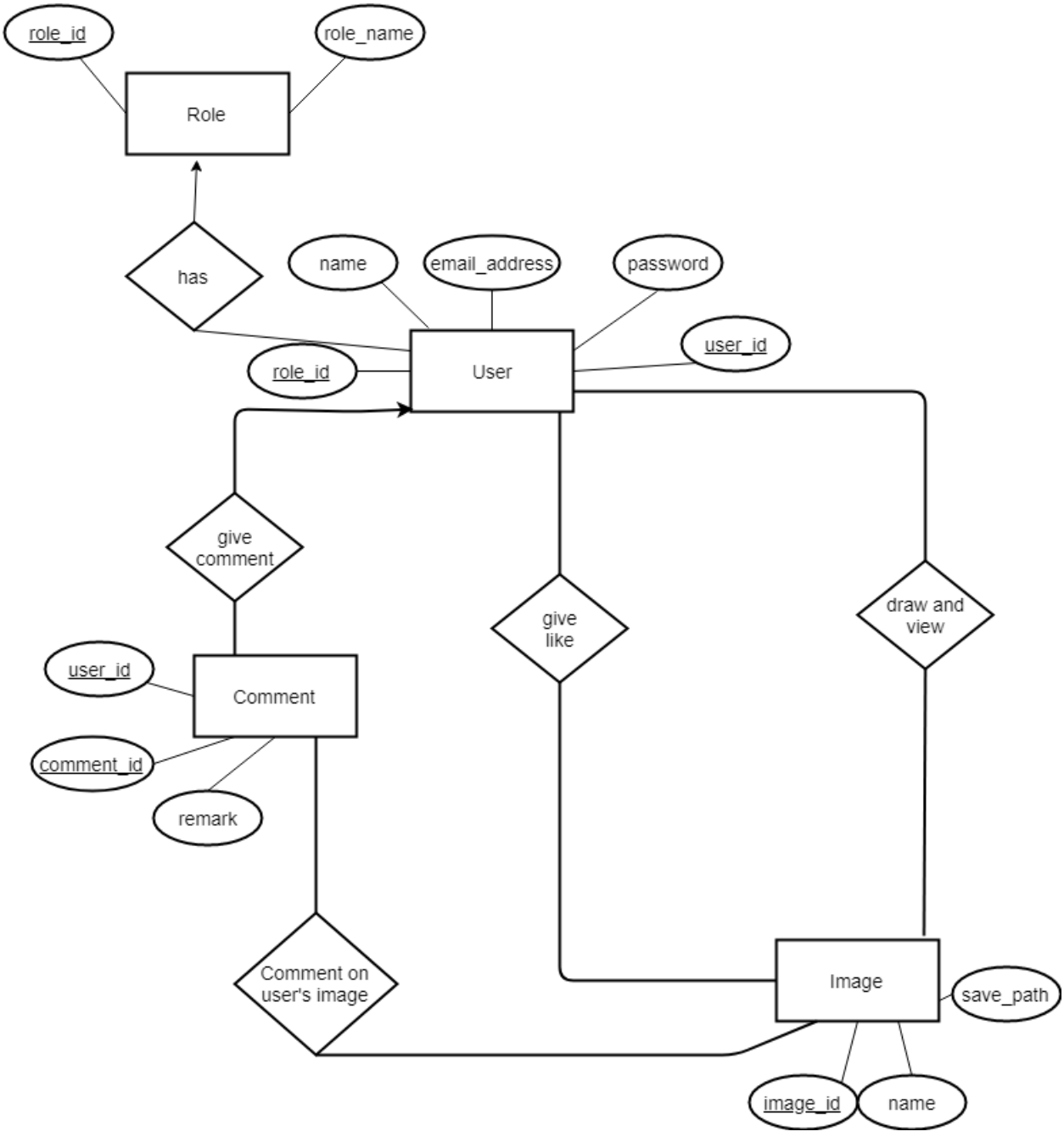


*Figure 26 Entity Relation Diagram*

## 5.7 Table Translation

*Table 3 Table Translation*

| Serial no | Data Table |
|---|---|
| 1 | User(user_id, role_id, name, email_address, password) |
| 2 | Image(image_id, image_name, save_path) |
| 3 | Comment(comment_id, user_id, remark) |
| 4 | View_image(user_id, image_id) |
| 5 | Give_like(user_id, image_id) |
| 6 | Comment_on_user's_image(comment_id, image_id) |
| 7 | Role(role_id, role_name) |

## 5.8 Schema Tables

*Table 4 Schema Table for User*

| Attributes | Type | Size |
|---|---|---|
| role_id | VARCHAR | 20 |
| Name | VARCHAR | 50 |
| email_address | VARCHAR | 50 |
| Password | VARCHAR | 50 |

*Table 5 Schema Table for Image*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| image_id | VARCHAR | 50 |
| image_name | VARCHAR | 50 |
| save_path | VARCHAR | 200 |

*Table 6 Schema Table for Comment*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| comment_id | VARCHAR | 50 |
| user_id | VARCHAR | 20 |
| remark | VARCHAR | 200 |

*Table 7 Schema Table for Give_like*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| user_id | VARCHAR | 20 |
| image_id | VARCHAR | 50 |

*Table 8 Schema Table for View_image*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| user_id | VARCHAR | 20 |
| comment_id | VARCHAR | 50 |

*Table 9 Schema Table for Comment_on_user's_image*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| comment_id | VARCHAR | 20 |
| image_id | VARCHAR | 50 |

*Table 10 Schema Table for Role*

| **Attributes** | **Type** | **Size** |
|---|---|---|
| role_id | VARCHAR | 20 |
| role_name | VARCHAR | 50 |

# CHAPTER SIX: CLASS MODEL OF PIICHKARI- ONLINE PAINT EDITOR

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.

## 6.1 Grammatical Parsing and Analysis

In order to identify the analysis classes, firstly, we grammatically parsed all the nouns and then categorized them according to general classification and selection criteria. We identified potential class by identifying the nouns from the usage scenario. Then we compared those with the following criteria whether they matched or not. We noted down the number of the fulfilled criteria at the right column.

### 6.1.1 Class Identification With General Classification

In table 11, the nouns from the usage scenario are classified by general classification. Also, here, by “P” we meant a noun is in problem domain and by “S” we meant solution space.

**General Classification:**

1. External entities
2. Things
3. Occurrence or events
4. Roles
5. Organizational unit
6. Places
7. Structure

*Table 11 Class Identification with General Classification*

| Serial No | Nouns | P/S | General Classification(GC) |
|---|---|---|---|
| 1 | Homepage | P | |
| 2 | Authentication | S | 3 |
| 3 | Registration | S | 3 |
| 4 | User | P | |
| 5 | member | S | 4 |
| 6 | software | P | |
| 7 | registration form | S | |
| 8 | name | S | |
| 9 | email address | S | |
| 10 | password | S | |
| 11 | validation | S | 3 |
| 12 | account | P | |
| 13 | information | P | |
| 14 | confirmation mail | P | |
| 15 | applicant's mail | P | |
| 16 | website | P | |

| | | | |
|---|---|---|---|
| 17 | invalid mail | P | |
| 18 | system | P | |
| 19 | invalid credential | P | |
| 20 | wrong password | P | |
| 21 | forget password | P | |
| 22 | link | P | |
| 23 | personal account | P | |
| 24 | images | S | 2 |
| 25 | gallery | P | |
| 26 | canvas | S | 4 |
| 27 | picture | P | |
| 28 | image name | S | 2 |
| 29 | old password | P | |
| 30 | role | S | |
| 31 | activities | P | |
| 32 | admin | S | 4 |
| 31 | interface | P | |
| 34 | normal user | P | |

| | | | |
|---|---|---|---|
| 35 | user's information | P | |
| 36 | reason | P | |
| 37 | drawing | S | 3 |
| 38 | painting | S | 3 |
| 39 | color palette | S | |
| 40 | colors | P | |
| 41 | paint brushes | S | |
| 42 | different sizes | P | |
| 43 | pencil | S | |
| 44 | eraser | S | |
| 45 | saved pictures | P | |
| 46 | letters | P | |
| 47 | suggestions | P | |
| 48 | options | P | |
| 49 | like | S | 3 |
| 50 | comment | S | 3 |
| 51 | reply | P | |
| 52 | number of likes | S | |

| | | | |
|---|---|---|---|
| 53 | Database | S | 4 |

### 6.1.2 Class Identification with Selection Criteria

The nouns having two or more than two were selected from the general classification list. After that step, we compared them with the following criteria list. Those are-

1. Retained information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

*Table 12 Class Identification with Selection Criteria*

| Serial No | Noun | Accepted | Remarks |
|---|---|---|---|
| 1 | Authentication | 3,4,5 | Accepted |
| 2 | Registration | 3 | Accepted |
| 3 | Role | 1,2,3,4,5 | Accepted |
| 4 | Member | 1,2,3,4,5 | Accepted |
| 5 | Validation | 3,4,5 | Accepted |
| 6 | Images | 3,4,5 | Accepted |

| | | | |
|---|---|---|---|
| 7 | Canvas | | Rejected |
| 8 | Admin | 1,2,3,4,5 | Accepted |
| 9 | Drawing | | Rejected |
| 10 | Painting | | Rejected |
| 11 | Like | 4,5 | Accepted |
| 12 | Comment | 4,5 | Accepted |
| 13 | Database | 2,5 | Accepted |

## 6.2 Preliminary Classes

- Authentication
- Registration
- Role
- Member
- Admin
- Validation
- Image
- Database

## 6.3 Verb Identification

*Table 13 Verb Identification*

| Serial No | Verbs | P/S |
|---|---|---|
| 1 | Contain | P |
| 2 | Register | S |
| 3 | Provide | P |
| 4 | Checked | S |
| 5 | Used | P |
| 6 | Prompt | P |
| 7 | Created | P |
| 8 | Enter | P |
| 9 | Show | S |
| 10 | Match | S |
| 11 | Try | P |
| 12 | Forget password | S |
| 13 | Send email | S |
| 14 | View image | S |

| | | |
|---|---|---|
| 15 | Go to canvas | S |
| 16 | Draw new picture | S |
| 17 | Modify | S |
| 18 | Change | P |
| 19 | Log out | S |
| 20 | Delete account | S |
| 21 | View user's information | S |
| 22 | Ban user | S |
| 23 | Report user | S |
| 24 | Paint | S |
| 25 | Draw | S |
| 26 | Save picture | S |
| 27 | Search user | S |
| 28 | Type | P |
| 29 | Show suggestions | S |
| 30 | Visit account | S |
| 31 | Visit gallery | P |
| 32 | Open image | P |

| | | |
|---|---|---|
| 33 | See who has given like | P |
| 34 | See who has commented | P |
| 35 | See total number of likes | P |
| 36 | Comment on pictures | S |
| 37 | Like pictures | S |

## 6.4. Attributes and Methods of Preliminary Classes

*Table 14 Attributes and Methods of Preliminary Classes*

| Serial No | Preliminary Class | Nouns | Verbs |
|---|---|---|---|
| 1 | Authentication | | Forget password, send email, log out |
| 2 | Registration | | Register |
| 3 | Role | Name, Email address, password, registration form | view Image, go to canvas, modify, delete account, search user, show suggestions, visit account |
| 4 | Member | Name, Email address, password, registration form | view Image, go to canvas, modify, delete account, search user, show suggestions, visit account, report user |

| | | | |
|---|---|---|---|
| 5 | Admin | Name, Email address, password, registration form | view Image, go to canvas, modify, delete account, search user, show suggestions, visit account, view user's information, ban user |
| 6 | Validation | | Checked, match |
| 7 | Image | Image name, canvas, drawing, painting, color palette, paint brushes, pencil, eraser, like, comment, number of likes | draw new picture, paint, draw, save picture, Comment on pictures, like Pictures |
| 8 | Database | | Update, retrieve, store, delete |

## 6.5 Analysis of Potential Classes

1. The Role class has similarities with Member and Admin class. So, Member and Admin class can extend Role class.
2. Authentication and registration classes can be merged into one class Authentication.
3. Database class can be removed since the methods of Database class can be fulfilled by other classes.

## 6.6 Final Classes

The final classes are:

1. Role
    1.1. Member
    1.2. Admin
2. Authentication
3. Image
4. Validation

## 6.7 Attributes and Methods of Final Classes

*Table 15 Attributes and Methods of Validation*

| Attributes | Methods |
|---|---|
| | isEmailExisting()<br>checkPasswordLength()<br>checkEmail&Password() |

*Table 16  Attributes and Methods of Authentication*

| Attributes | Methods |
|---|---|
| | viewOption()<br>signUp()<br>signIn()<br>signOut()<br>forgetPassword() |

Table 16: Attributes and Methods of Role

| Attributes | Methods |
|---|---|
| Name<br>Email Address<br>Password | viewPersonalInformation()<br>viewOwnPicture()<br>searchOtherUser()<br>deleteAccount()<br>modifyInformation()<br>viewOthersPicture() |

*Table 17 Attributes and Methods of Admin*

| Attributes | Methods |
|---|---|
| | viewOtherUsersInformation()<br>banUser() |

*Table 18 Attributes and Methods of Member*

| Attributes | Methods |
|---|---|
| | reportUser() |

*Table 19 Attributes and Methods of Image*

| Attributes | Methods |
|---|---|
| imageName<br>imagePath | convertPictureToImage()<br>saveImage()<br>downloadImage()<br>editImage()<br>convertImageToPicture()<br>likePicture()<br>commentPicture() |

## 6.8 Class Responsibilities and Collaborators

*Table 20 Class Card of Validation*

| Attributes | Methods |
|---|---|
| | isEmailExisting()<br>checkPasswordLength()<br>checkEmail&Password() |
| Responsibility | Collaborators |
| Check for existing email | Role |
| Check Password length | |
| Check Email And Password | Role |

*Table 21 Class Card of Authentication*

| Attributes | Methods |
|---|---|
| | viewOption()<br>signUp()<br>signIn()<br>signOut()<br>forgetPassword() |
| Responsibilities | Collaborators |
| Viewing options | |
| Signing up | Role, Validation |
| Signing in | Role, Validation |
| Signing out | Role |
| Changing password if user forgets | Role |

*Table 22 Class Card of Role*

| Attributes | Methods |
|---|---|
| Name<br>Email Address<br>Password | viewPersonalInformation()<br>viewOwnPicture()<br>searchOtherUser()<br>deleteAccount()<br>modifyInformation()<br>viewOthersPicture() |
| Responsibilities | Collaborators |
| Viewing personal information | |
| Viewing Own Picture | Image |
| Search other user | |
| Deleting account | |
| Modifying Information | |
| Viewing Others pictures | Image |

*Table 23 Class Card of Admin*

| Attributes | Methods |
|---|---|
| | viewOtherUsersInformation()<br>banUser() |
| Responsibilities | Collaborators |
| Viewing Other users personal information | |
| Banning user | Member |

*Table 24 Class Card of Image*

| Attributes | Methods |
|---|---|
| imageName<br>imagePath | convertPictureToImage()<br>saveImage()<br>downloadImage()<br>editImage()<br>convertImageToPicture()<br>likePicture()<br>commentPicture() |
| Responsibilities | Collaborators |
| Saving Image | Role |
| Downloading image | |
| Editing image | |
| Like Image | Role |
| Comment on an Image | Role |
| Conversion between image and picture | |

*Table 25 Class Card of Member*

| Attributes | Methods |
|---|---|
| | reportUser() |
| Responsibilities | Collaborators |
| Reporting user | |

## 6.9 Class Diagram

Class diagram is a diagram where dynamics of object interaction and collaboration are represented through UML diagrams and their networks.

Here composition, association and inheritance of the classes are shown in the diagram. The notations are in figure 25-

*Table 26 Notations*

| Relationship | Notation |
|---|---|
| Association |  |
| Composition |  |

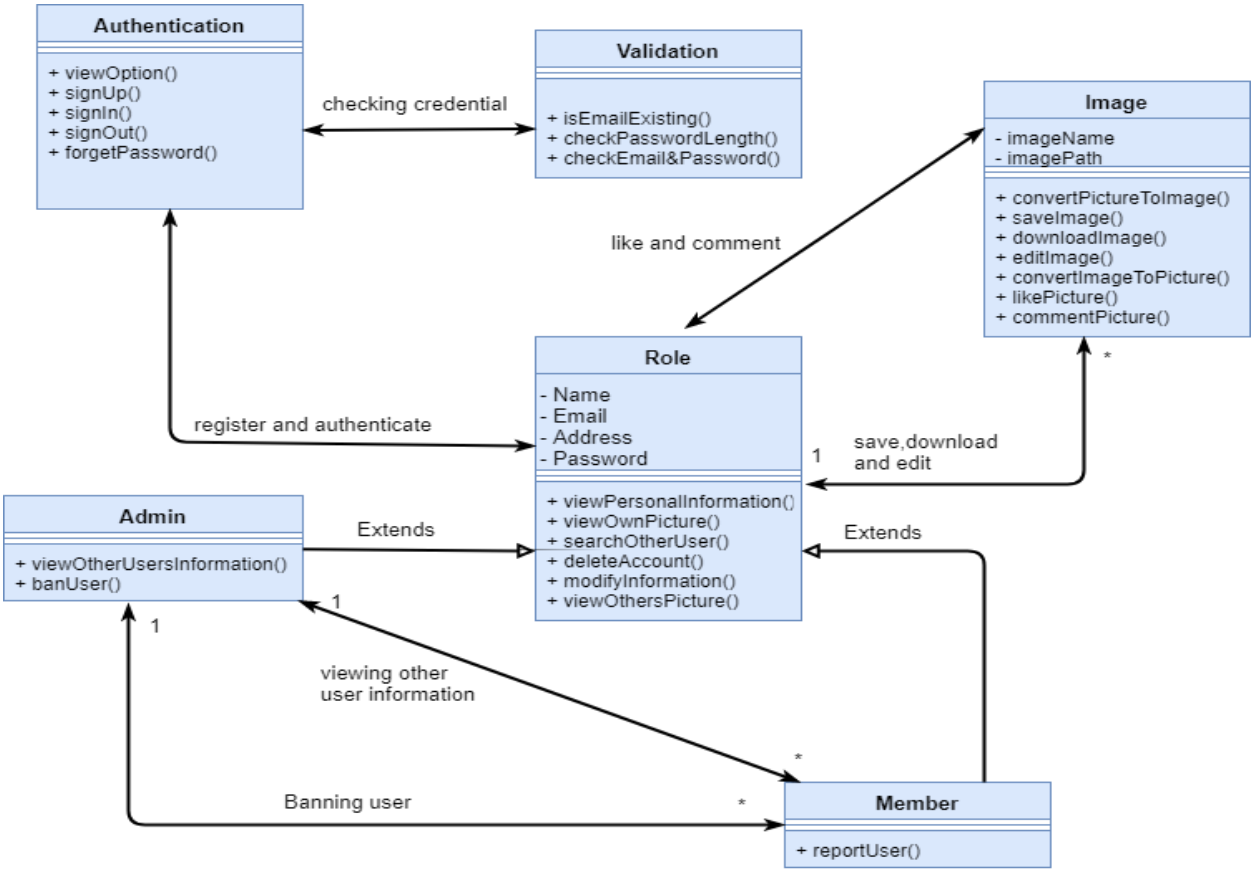


*Figure 27 CRC Diagram*

# CHAPTER SEVEN: BEHAVIORAL MODEL OF PIICHKARI- ONLINE PAINT EDITOR

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.

## 7.1 Event Identification

From the usage scenario, we have identified the events, their initiators and collaborators.
They are shown in Table 28-

*Table 27 Event Identification*

| Serial No | Events | Initiator | Collaborator | Invoked Method |
|---|---|---|---|---|
| 1 | Sign up | Authentication | Validation | **Authentication:** ViewOption(), signUp() **Validation:** IsEmailExisting(), checkPasswordLength() |
| 2 | Sign in | Authentication | Validation | **Authentication:** ViewOption(), signIn() **Validation:** checkEmailAndPassword() |
| 3 | Forget password | Authentication | | **Authentication:** ViewOption(), forgetPassword() |
| 4 | Check for existing email | Validation | Role | **Validation:** isEmailExisting() |
| 5 | Check password length | Validation | | **Validation:** checkPasswordLength() |

| | | | | |
|---|---|---|---|---|
| | Check email & password | Validation | Role | **Validation:** CheckEmailAndPassword() |
| 7 | Search other user | Role | | **Role:** searchOtherUser() |
| 8 | Delete account | Role | | **Role:** deleteAccount() |
| 9 | View personal information | Role | | **Role:** ViewPersonalInformation() |
| 10 | Modifying information | Role | | **Role:** modifyInformation() |
| 11 | View image | Role | Image | **Role:** viewOwnImage(), ViewOthersImage() |
| 12 | Save image | Image | Role | **Image:** saveImage() |
| 13 | Edit image | Image | Role | **Image:** editImage() |
| 14 | Download image | Image | Role | **Image:** downloadImage() |
| 15 | Like image | Image | Role | **Image:** likeImage() |
| 16 | Comment on image | Image | Role | **Image:** commentOnImage() |
| 17 | Ban user | Admin | | **Admin:** banUser() |
| | View other user's | Admin | Member | **Admin:** viewOtherUser'sInformation() |

| | | | | |
|---|---|---|---|---|
| | information | | | |
| 19 | Report user | Member | | **Member:** reportUser() |
| 20 | Sign out | Authentication | Role | **Authentication:** signOut() |
| 21 | View option | Authentication | | **Authentication**: viewOption() |

## 7.2 State Diagram

State-transition diagrams describe all of the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions). State-transition diagrams are very useful for describing the behavior of individual objects over the full set of use cases that affect those objects. State-transition diagrams are not useful for describing the collaboration between objects that cause the transitions.

### 7.2.1 Role

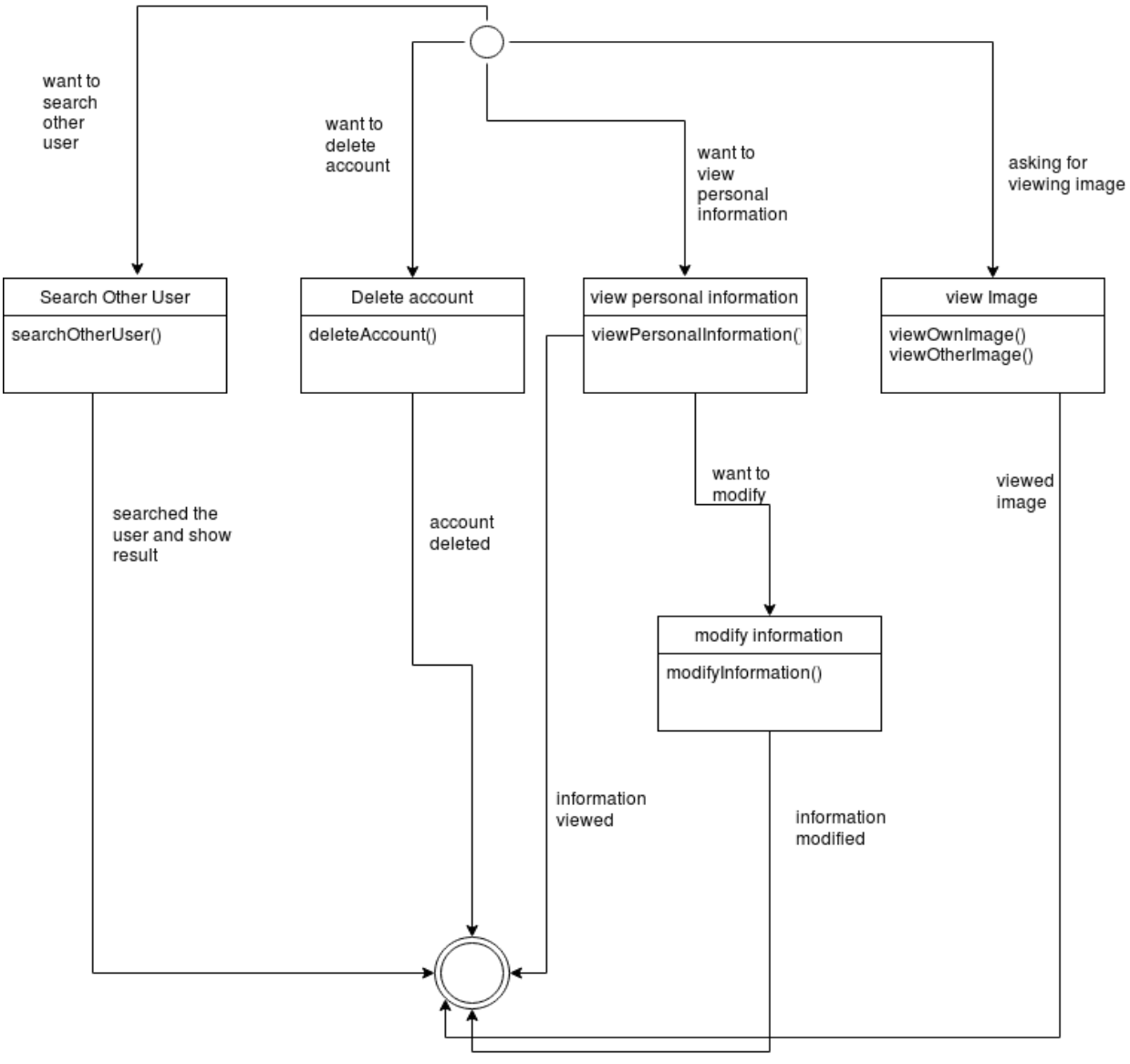


*Figure 28 State Diagram of Role*

### 7.2.2 Member

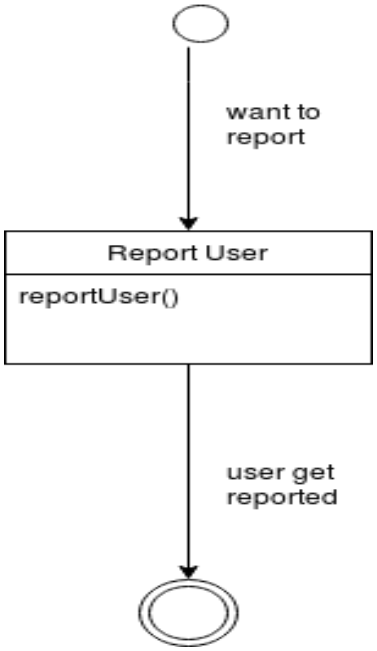


*Figure 29 State Diagram of Member*

### 7.2.3 Admin

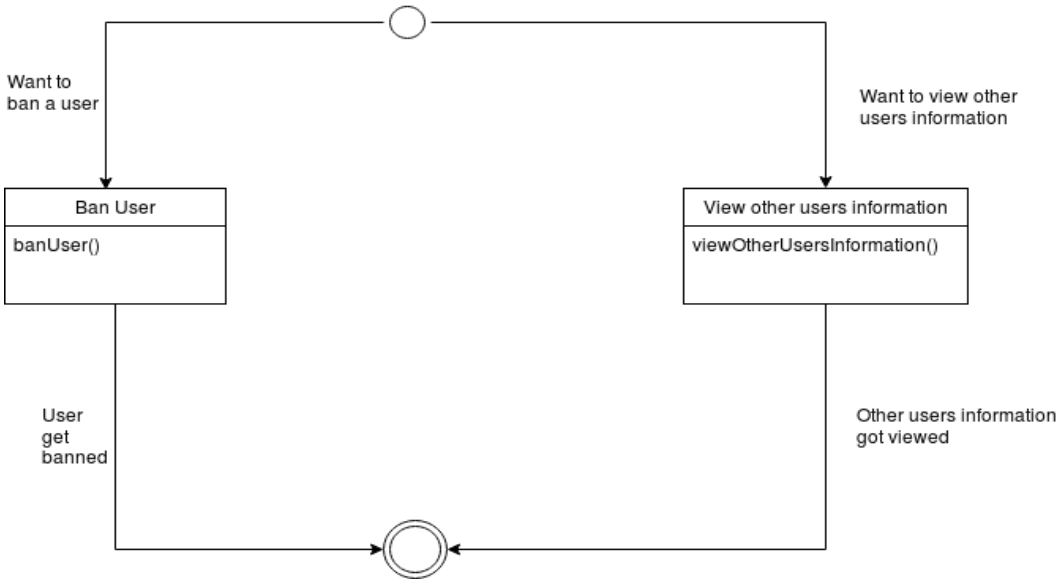


*Figure 30 State Diagram of Admin*

### 7.2.4 Validation

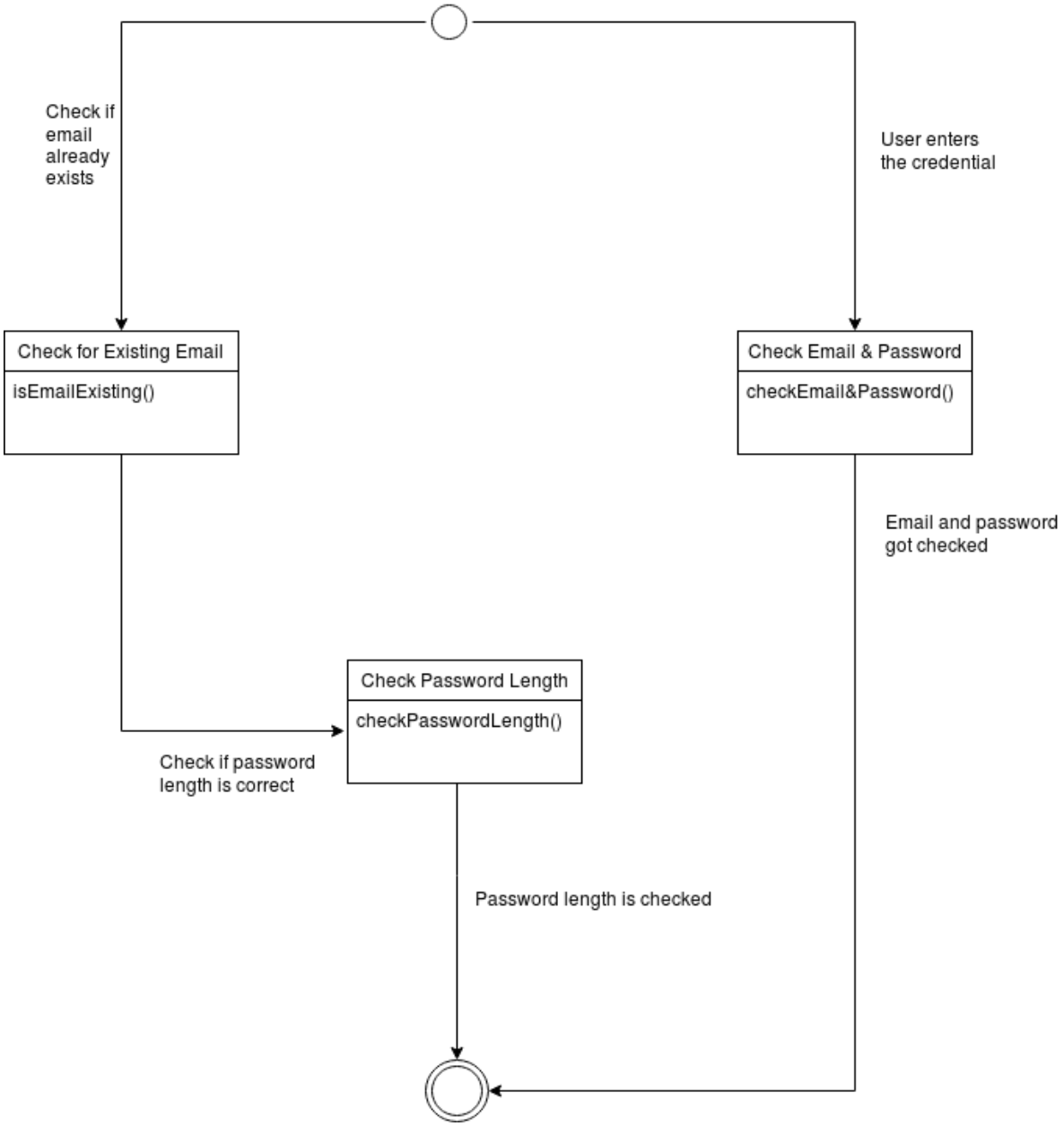


*Figure 31 State Diagram of Validation*

## 7.2.5 Image

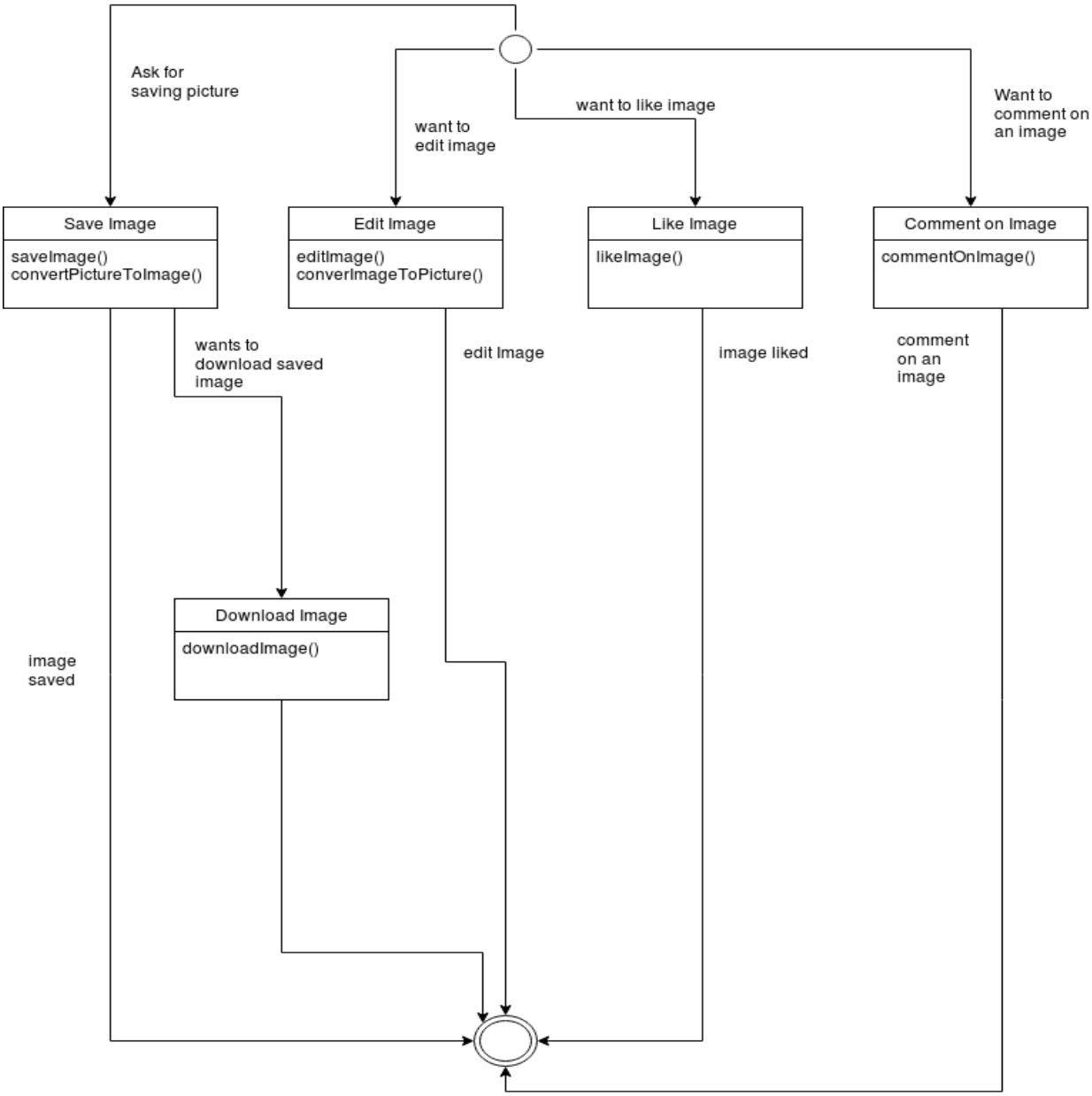


*Figure 32 State Diagram of Image*

### 7.2.6 Authentication

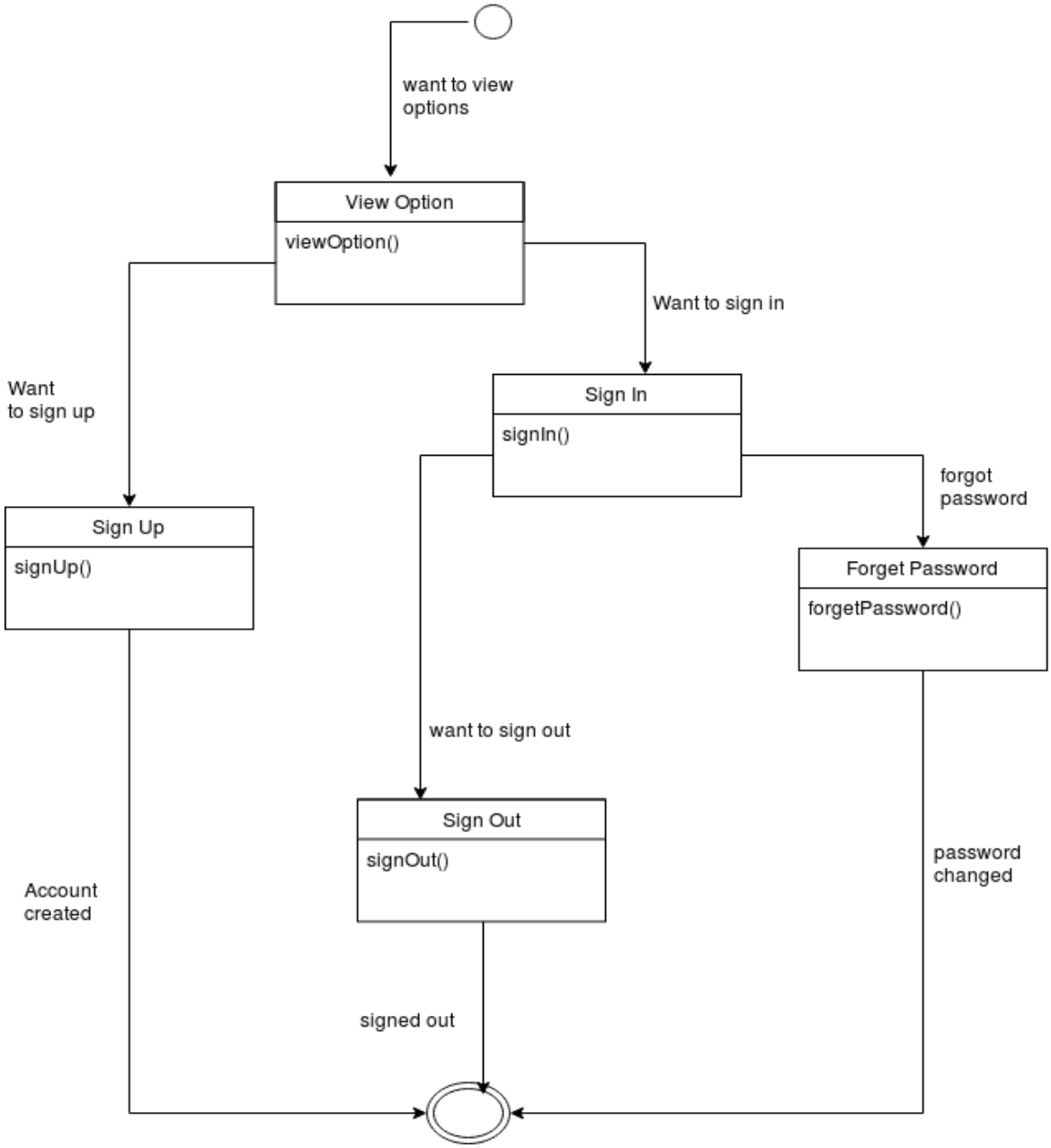


*Figure 33 State Diagram of Authentication*

## 7.3 Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

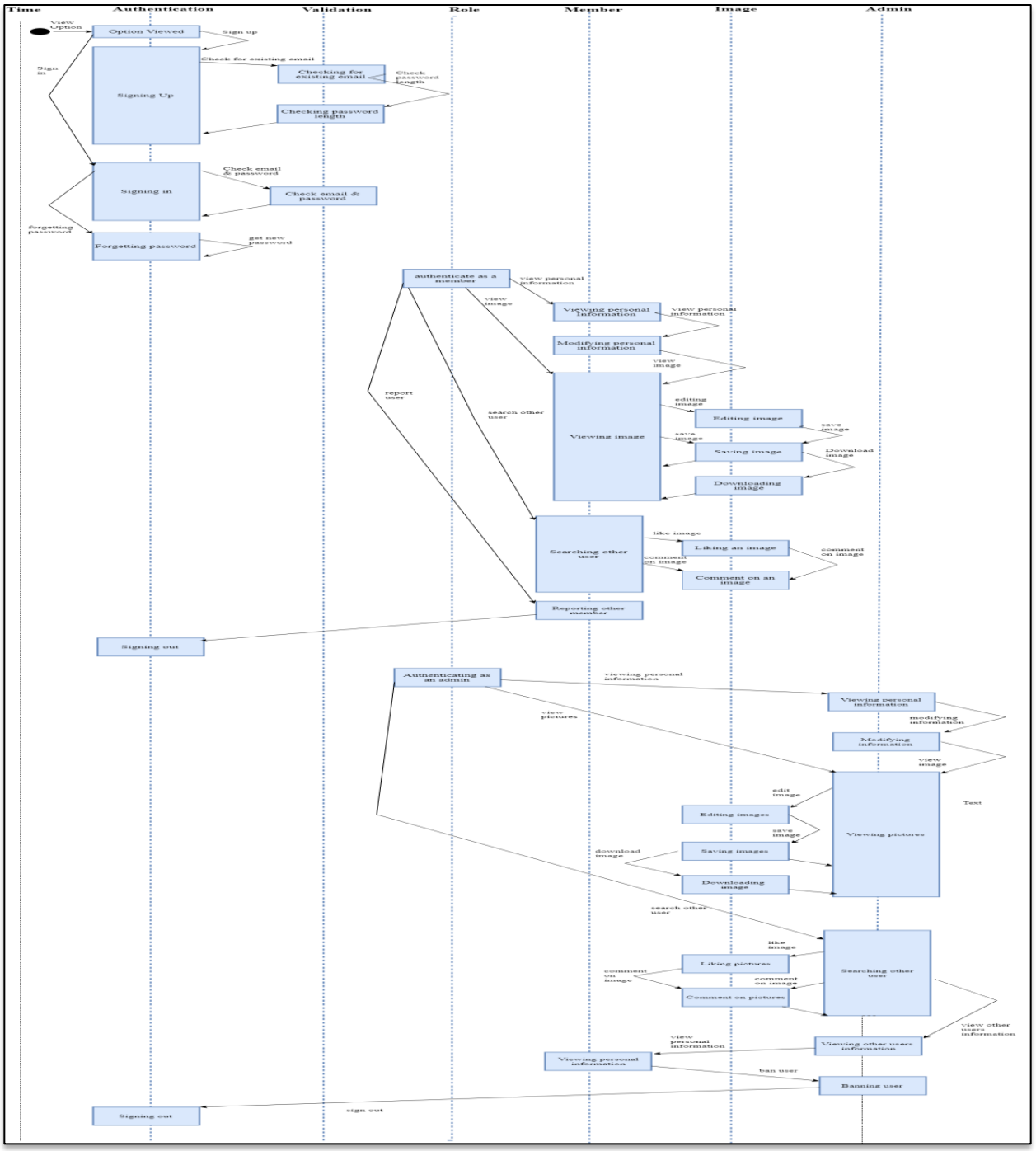


*Figure 34 Sequence Diagram of Piichkari*

# CHAPTER EIGHT: DATA FLOW DIAGRAM of PIICHKARI

A data flow diagram (DFD) maps out the flow of information for any process or system. DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. There are four components of DFD. They are:

1. External entity
2. Process
3. Data store
4. Data flow

## 8.1 Level 1 Data Flow Diagram

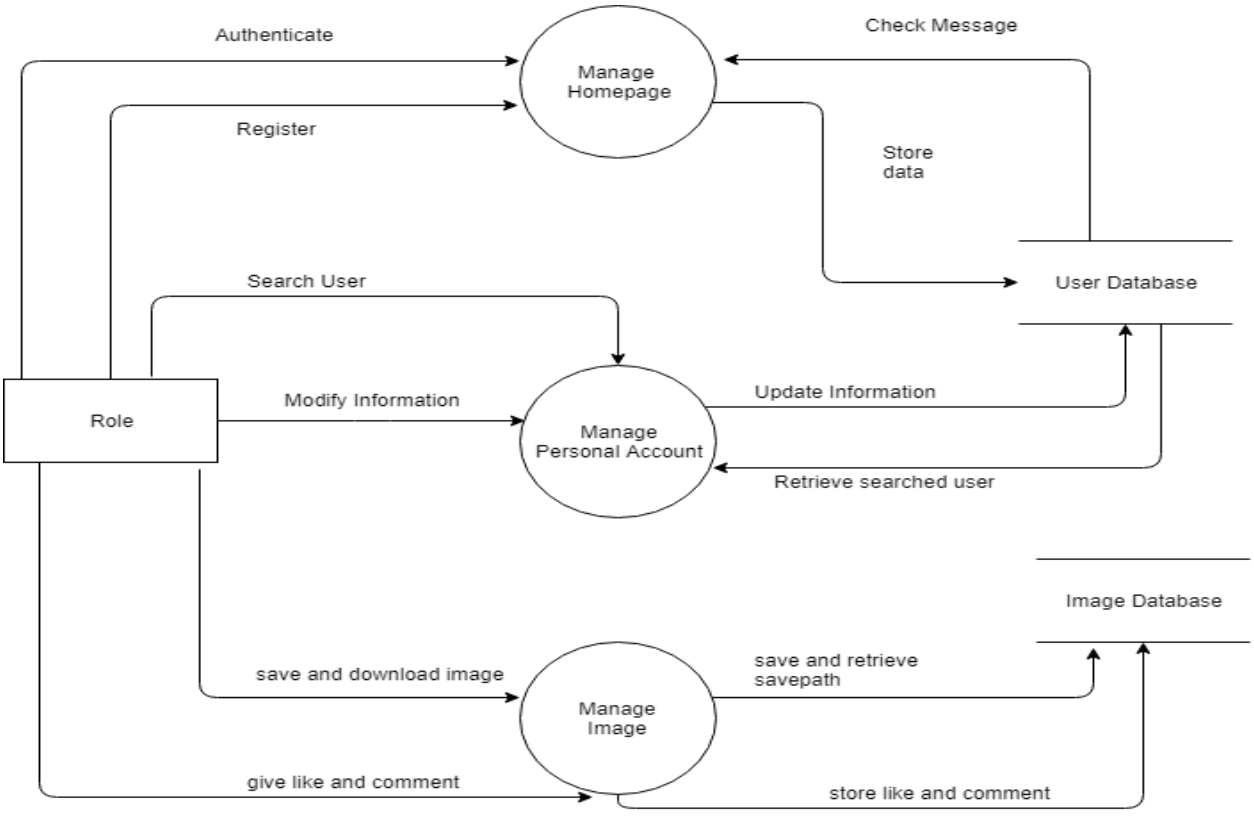


*Figure 35 Level 1 Data Flow Diagram*

## 8.2 Level 1.1 Data Flow Diagram

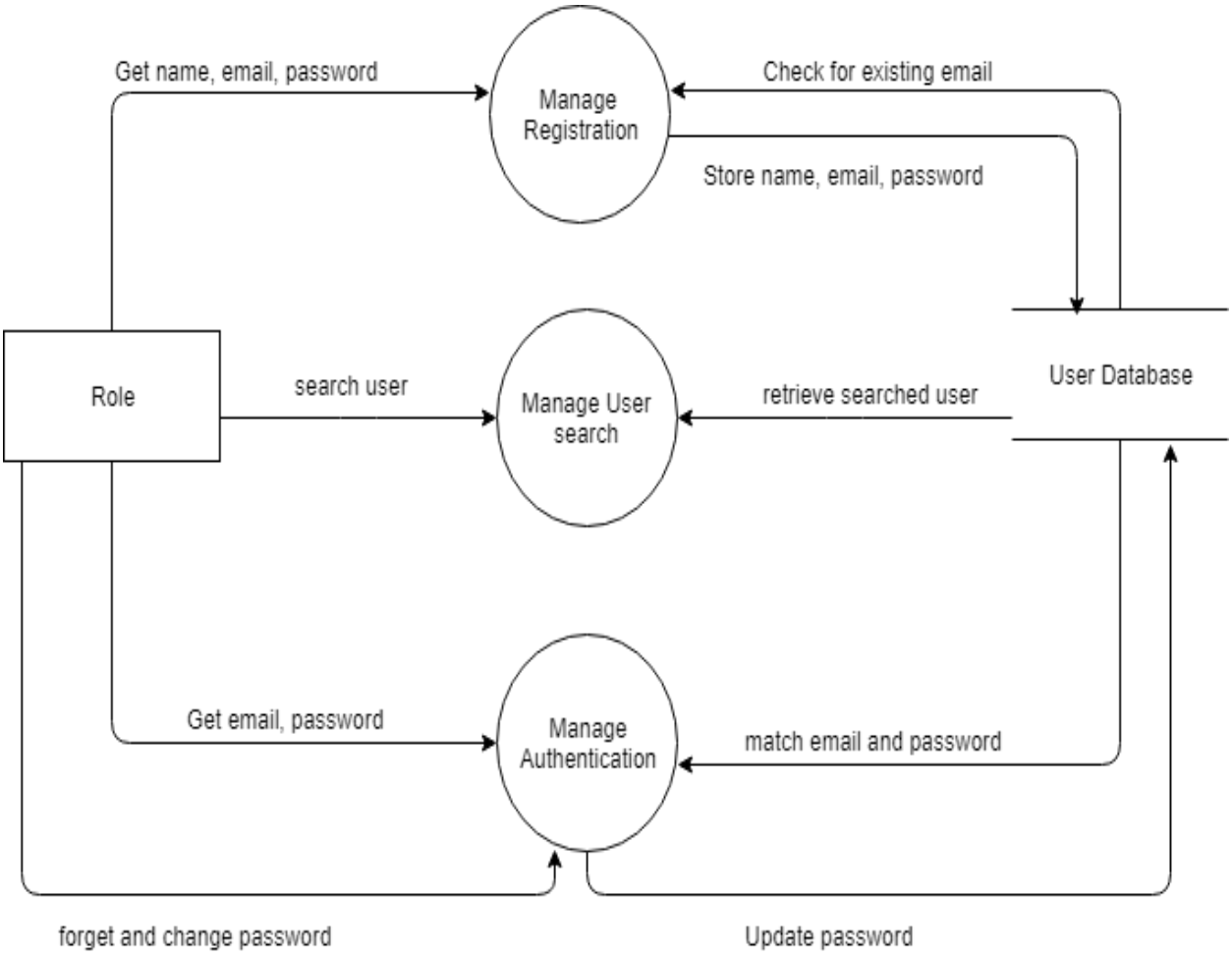


*Figure 36 Level 1.1 Data Flow Diagram*

## 8.3 Level 1.2 Data Flow Diagram

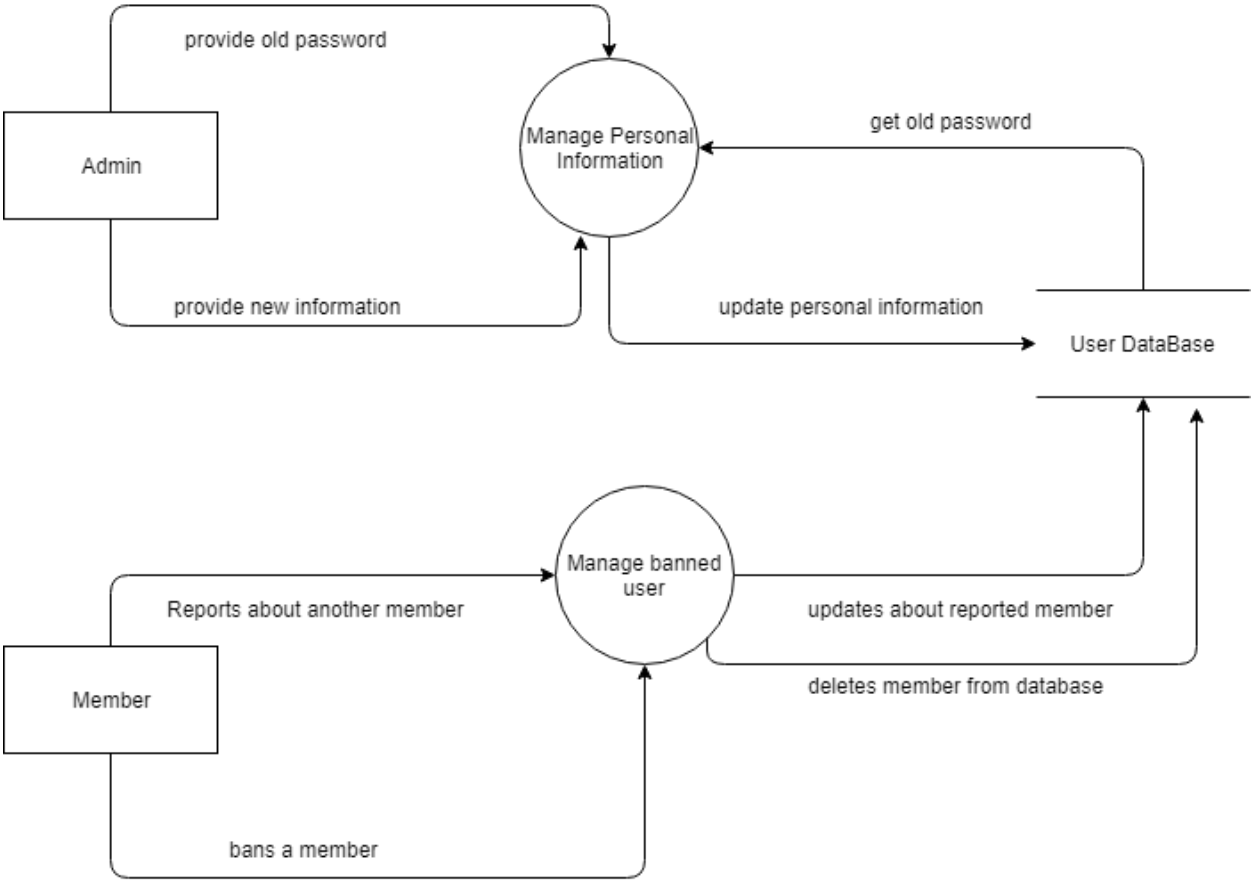


*Figure 37 Level 1.2 Data Flow Diagram*

## 8.4 Level 1.3 Data Flow Diagram

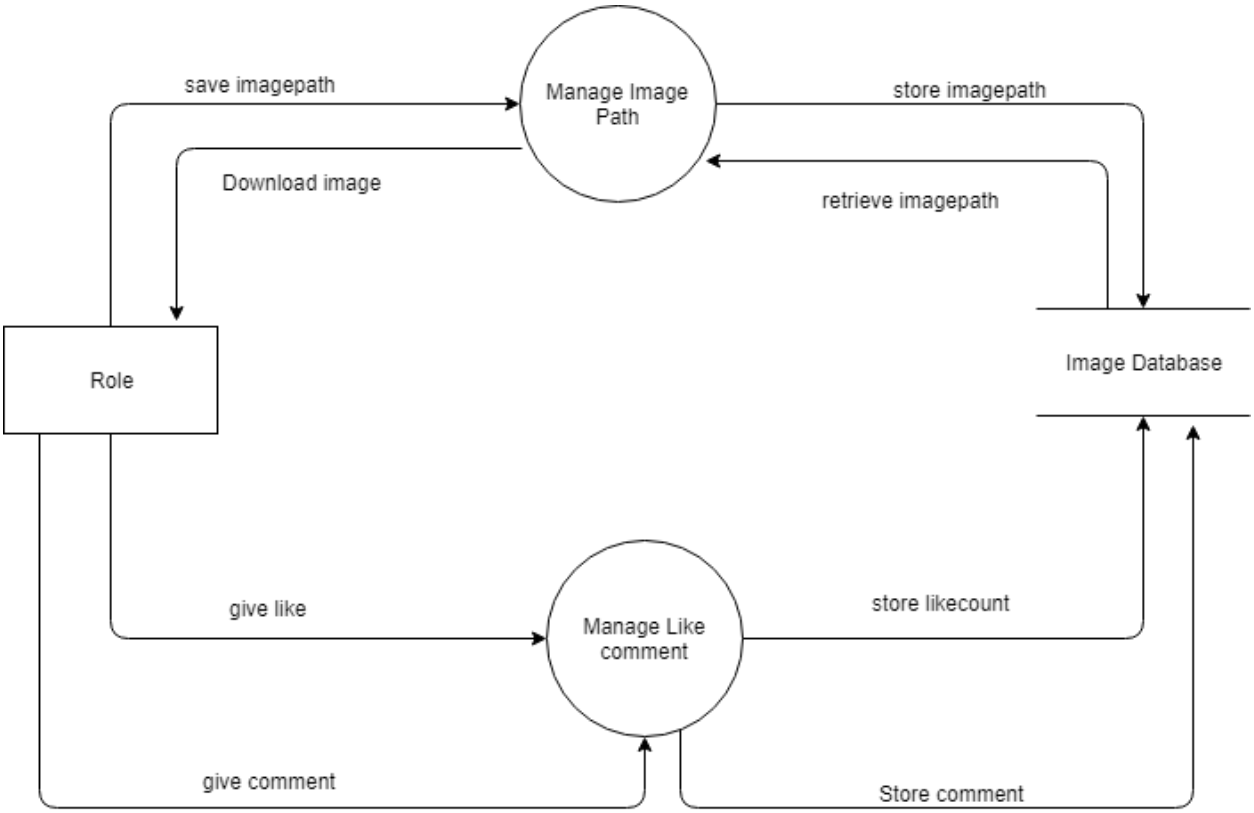


*Figure 38 Level 1.3 Data Flow Diagram*

# CHAPTER NINE: CONCLUSION

Documentation of SRS is the key activity to elicit and understand, elaborate and illustrate the business requirements and forms a crucial building block in the process of software engineering. We are extremely happy to submit our SRS report of PIICHKARI. Hopefully this document will help the readers to get a clear view of the entire system. The document will help the developers to maintain a workflow and will help them to keep track of the work. The SRS report will be helpful to maintain the software development cycle.

# REFERENCES

[1] “http://discovery.ucl.ac.uk/744/1/1.7_stake.pdf” [Last accessed: 18-02-2018 10 am]

[2] “Book- Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th Edition)”