

Camera Synchronization

An android app

**Software Requirement
Specification and Analysis**

**Camera Synchronization
An Android App**

Course: SE 505

Submitted by

S.M. Shahriyar (BSSE 0902)

Ishrat Jahan Emu (BSSE 0927)

BSSE Session: 2016-2017

Supervised by

Dr. Naushin Nower

Assistant Professor



**Institute of Information Technology
University of Dhaka**

Date: 20/03/2019

Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on F- Commerce based Client Management System. We would like to thank whole-heartedly our supervisor Dr. Naushin Nower ,,,Accossiate Professor, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. We are also thankful of those people who gave requirements to do this job.

Abstract

At present, we are living in a world where technology rules almost every place and every person. Internet offers us to grab anything from any corner of the world now. In almost twinkling of an eye, people enjoy a wide range of facilities given by internet through a small smartphone , tab or laptop. Following the stream, we have developed a camera synchronization application for android smartphones. When using this app, a user will be able to sync his and his friends' camera through creating hotspot.. This will help to capture perfect photo.

Table of Contents

Chapter One	7
Introduction	7
1.1 Purpose	7

1.2 Intended Audience	7
Chapter Two	9
Inception.....	9
2.1 Introduction	9
2.2 Conclusion	10
Chapter Three.....	11
Elicitation.....	11
3.1 Introduction	11
3.2 Eliciting requirements	11
3.3 Requirements gathering	12
3.4 Quality function deployment	12
3.5 User scenario	13
3.6 Elicitation work product	14
Chapter Four	15
Scenario Based Model	15
4.1 Introduction	15
4.2 Use case diagram	15
4.2 Use Case Scenario.....	16
Figure 1: Level 0 Use Case Diagram	17
Figure 2: Level 1 Use Case Diagram	18
Figure 3: Level 1.1 Use Case Diagram of Authentication Subsystem	19
Figure 4: Level 2.2 Use Case Diagram of Group Creation Subsystem	20
Figure 5: Level 2.3 Use Case Diagram of Search Subsystem	21
Figure 6: Level 1.3 Use Case Diagram of View Subsystem.....	22
Figure 7: Level 1.4 Use Case Diagram of Settings Subsystem	23
4.3 Activity Diagrams of Camera Synchronization	24
Figure 9: Level 4.3.1 Activity Diagram of Sign Up	25
Figure 10: Level 4.3.2 Activity Diagram of Sign In.....	26
Figure 10: Level 4.3.2 Activity Diagram of An-authentic	27
Figure 11: Level 4.3.3 Activity Diagram of Sign out	28
Figure 12: Level 4.3.4 Activity Diagram of Send Request.....	29
Figure 13: Level 4.3.4 Activity Diagram of Cancel Request.....	31
Figure 14: Level 4.3.5 Activity Diagram of Accept Request	31
Figure 15: Level 4.3.4 Activity Diagram of Remove from group	32
Figure 16: Level 4.3.4 Activity Diagram of profile view	33
Figure 17: Level 4.3.4 Activity Diagram of Camera View.....	35

Figure 18: Level 4.3.4 Activity Diagram of Change profile photo.....	36
Figure 18: Level 4.3.4 Activity Diagram of Change user name	37
4.4 Swim-lane diagram of Camera Synchronization.....	37
4.4.1 Swim-lane diagram of Sign Up	37
4.4.2 Swim-lane diagram of Sign In.....	39
4.4.3 Swim-lane diagram of Un-authentic.....	40
4.4.4 Swim-lane diagram of Sign out	41
4.4.5 Swim-lane diagram of Send Request.....	42
4.4.7 Swim-lane diagram of Cancel Request	43
4.4.8 Swim-lane diagram of Accept Request.....	44
4.4.9 Swim-lane diagram of Remove from group.....	45
4.4.10 Swim-lane diagram of Profile View.....	46
4.4.11 Swim-lane diagram of Camera View.....	47
4.4.12 Swim-lane diagram of Change profile photo.....	48
4.4.13 Swim-lane diagram of Change User name.....	49
Chapter Five	50
Data Model	50
5.1 Introduction.....	50
5.2 Data object.....	50
5.3 ER diagram.....	54
5.4 Data schema table	55
Chapter Six	57
Class Based Model Of Camera Synchronization Android Application	57
6.1 Grammatical Parsing and Analysis	57
6.1.1 CLASS IDENTIFICATION WITH GENERAL CLASSIFICATION.....	57
6.1.2 CLASS IDENTIFIED WITH SELECTION CRITERIA.....	60
6.2.1 Attributes and Methods of Preliminary Classes.....	63
6.2.2 Analysis on potential classes:	64
6.2.3 Final Classes.....	64
6.3 Class Responsibility Collaborator modeling.....	66
6.3.1 Class cards	67
6.3.2 UML Diagram.....	69
BEHAVIORAL MODEL OF 'CAMERA SYNCHRONIZATION' A MOBILE APPLICATION	70
7.1 State diagram of User :	70
7.2 State diagram of Member:	71
7.3 State diagram of Gallery:.....	71

7.4 State diagram of System:.....	72
7.5 State diagram of Database:.....	73
DATA FLOW MODEL OF 'CAMERA SYNCHRONIZATION' A MOBILE APPLICATION	74
8.1.1 level 1:	75
8.1.2: Level 2.1	75
8.2.1 Level 2.2.1.....	76
8.2.1 Level 2.2.2.....	77
8.2.1 Level 2.3.....	78
8.2.1 Level 2.4.....	79
8.2 Sequence diagram.....	79

Chapter One

Introduction

This chapter describes the objectives of this report as well as the audiences who should have to go through this report for individual purposes.

1.1 Purpose

This document is based on the Software Requirement Analysis (SRS) for Camera Synchronization android application. It includes all necessary requirements to develop this application no matter whether they are functional or non-functional. The information about the requirements here have been organized systematically. so that everyone can easily figure out a summarized concept about Camera Synchronization . This SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer and stakeholder community. It will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This report is intended for several audiences, including the customer, as well as the project managers, designers, developers, and testers.

- The customer will use this document to ensure that whatever he requires has been fulfilled by the project teams.
- The project managers of the developer team will use this SRS to fix a milestone and time to deliver the software and to ensure that the teams working on this project are on the right path.
- The designers will use this document 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.
- The developers will use this report as a basis for developing the system's functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created software that will fulfill all of the customer's documented requirements.
- The testers will use this SRS to derive test plans and test cases for each documented requirement. 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. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

Chapter Two

Inception

In this chapter, we will discuss about the first step of Software Requirements Specifications Analysis, that is, Inception.

2.1 Introduction

Requirement Engineering comprises several sequential steps. Inception is the first one among them. Inception creates the entrance to the project for the requirements analysts. It refers them how the project should get started. It also provides a basic idea to the engineers about the problems ahead which are needed to be solved and how critical obstacles may come during the project. The main target of Inception phase is to identify the people related to the project and their needs. In order to complete this phase, we have focused on---

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

2.1.1 Identifying stakeholders

Stakeholders are entities that have an interest in a given project. These stakeholders may be inside or outside an organization which:

- Sponsor a project, or
- Have an interest or a gain upon a successful completion of a project,
- May have a positive or negative influence in the project completion

There is one stakeholder for this system. That is an android app supplier who requested for the app to develop and design as well as specified the requirements for the app.

2.1.2 Requirements:

The requirements that have to be fulfilled in the application are:

- The app should be able to launch on any version of android smartphone.
- User can activate or deactivate the app from functioning.
- User can create hotspot for connecting
- User can connect by group sharing or adding account .
- The app should maintain a database for faster query.

2.1.3 Asking first questions

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

2.2 Conclusion

Inception phase helped us to establish basic understanding about the Camera Synchronization application, identify the people who will be benefited if the application is implemented, define the purpose of the project and establish a preliminary communication with the stakeholder.

Chapter Three

Elicitation

In this chapter, we will briefly discuss about the Elicitation phase of Camera Synchronization application.

3.1 Introduction

Requirements elicitation is recognized as one of the most critical, knowledge-intensive activities of software development; poor execution of elicitation will almost guarantee that the final project is a complete failure. Since project failures are so rampant, it is quite likely that improving how the industry performs elicitation could have a dramatic effect on the success record of the industry. Improving requirements elicitation requires us to first understand it. Although many papers have been written that define elicitation, or prescribe a specific technique to perform during elicitation, nobody has yet defined a unified model of the elicitation process that emphasizes the role of knowledge.

3.2 Eliciting requirements

Earlier we have seen that the methodology used in Inception phase is Question and Answer approach. But Elicitation is quite different in this point of view. The elicitation phase follows a format of eliciting requirements which combines the other four phases namely problem solving, elaboration, negotiation and specification. In order to elicit requirements, we have followed four steps:

- Collaborative Requirements gathering
- Quality Function Deployment (QFD)
- Usage Scenarios
- Elicitation work product

3.3 Requirements gathering

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

- Meetings have been conducted with the android app supplier and he was questioned about his requirements and expectations from the Camera Synchronization application.
- He was asked about the existing problems that are being faced by users without the application.
- Based on the meetings and response of him, we finally selected the requirements.

3.4 Quality function deployment

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

3.4.1 Normal requirements

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

- Joining or creating own group to share
- Allowing users to create hotspot
- 2 types connection require (Online & Offline)
- No editor is needed for this camera .

3.4.2 Expected requirements

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

dissatisfaction. The expected requirements of our app are:

- No editor is needed for this camera
- Color must be in RGB
- Not need to show the whole profile (Contact no only)
- Users can be joined to the group by requesting or adding

3.4.3 Exciting requirements

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying when present. The exciting requirements for our app are following:

- For online connection , there must have 4G connection
- Can be able to connect with other channels (for TV journalist)

3.5 User scenario

The aim of the app is to synchronize camera..

Authentication:

At first user must provide name, unique id, and password to create account. System will create a unique account for the user with the given information.

To use the app user does not need to enter own id and password to login each time. If any other user wants to login, the logged in user has to log out at first.

Connection:

This app is basically made for connecting two or more devices. User can connect with each other via online or offline. In offline , user can create hotspot and other can join the group . In online , user can be connected by using wi-fi.

Group Creation:

When logged in, a user can create and update own group. A user can add other users by searching with id. If searched account is available, it will show the users' profile photo. If the user is interested to add the searched user to group he can send request to the user. If a user accepts request from other users, both of them will be join the group and can view each other's profile .A user can also remove someone from the group .

View:

In view option, a user can view all friends' profile .. From the group , the user can select someone to view his camera view, where home position. If any user's on his camera mode , other group members can get the access of camera view.

Settings:

A user can change profile photo, username, and mode from settings.

3.6 Elicitation work product

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

- Set of usage scenarios.
- Description of the system's technical environment.
- Make a bounded statement of scope for our system.
- Make a list of user and other stakeholder who participated in requirements.
- Make a statement of the requirements for Camera Synchronization.

Scenario Based Model

This chapter is about the scenario based model of Camera Synchronization.

4.1 Introduction

Scenario based modelling is an inexpensive rapid prototyping technique. This method is effective when systems are being built with the requirements vaguely known at the outset. Users are involved right from the start, to build prototypes evolving towards the final product. The users are also involved with the testing of the prototypes which is essential for the validation of requirements and help the users to gain an initial experience of the final system during the development itself. This method involves techniques which are applied by one or more professionals working alongside users who are expected to provide and specify their requirements at the beginning as well as evaluate and approve the system upon completion. The user (in a passive capacity) and the designer/builder (an active partner) cooperate to reach a working model where the means of communications are by the examination of preliminary models such as the initial narratives, paper models and graphical representations built to represent the final system functions

4.2 Use case diagram

A use case diagram is a graphic depiction of the interactions among the elements of a system .

The purposes of use case diagrams are:

- Gathering requirements of a system.
- Getting an outside view of a system.
- Identifying external and internal factors influencing the system.
- Showing the interaction among actors.

The first step in writing a use case is to define the set of actors that will be involved in the story.

Actors are of two types. They are:

- 1. Primary Actors:** Primary actors are the actors using the system to achieve a goal. They both consume data and produce information.
- 2. Secondary Actors:** Secondary actors are the actors that the system needs assistance from to achieve the primary actor's goal. They either consume data or produce information.

Once actors have been identified, use cases can be developed.

4.2 Use Case Scenario

Level 0	Level 1	Level 2	Level 3
Camera Synchronization	Authentication	Sign Up (Registration)	
		Sign In	
		Sign Out	
		Offline(Unauthentic)	
	Group Creation	Search	Send Request
			Cancel Request
			Accept Request
			Remove from group
	View	Profile View	
		Camera View	
	Settings	Edit Account	Change Profile Photo
			Change Username

4.2.1 Use case diagrams and description of subsystems

There are two actors in the system- user and system. The use case diagram of each subsystem and their description will be discussed in this section.

Level 0: Camera Synchronization

Primary actor: User, System.

There are no secondary actors in Camera Synchronization application.

Goal in context: The diagram in Figure 1 represents the whole Camera Synchronization .

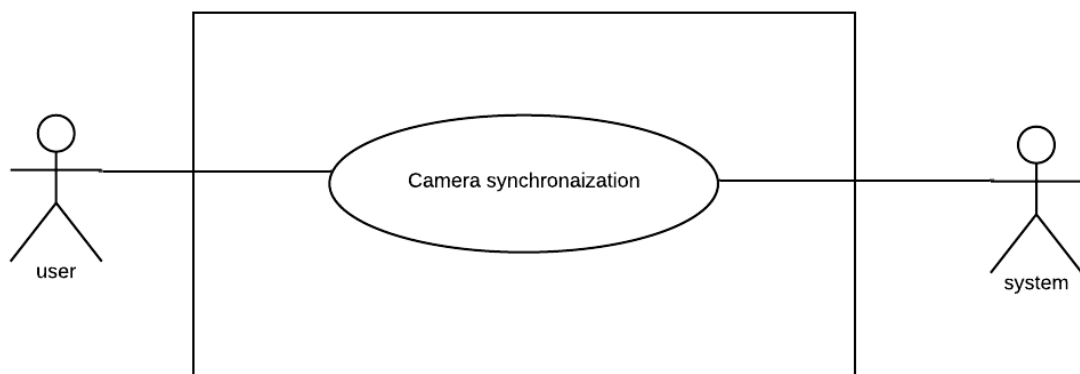


Figure 1: Level 0 Use Case Diagram

Level 1: Camera Synchronization

There are four subsystems in Camera Synchronization application. These are- Authentication, Group Creation, View and Settings. Figure 2 shows these subsystems.

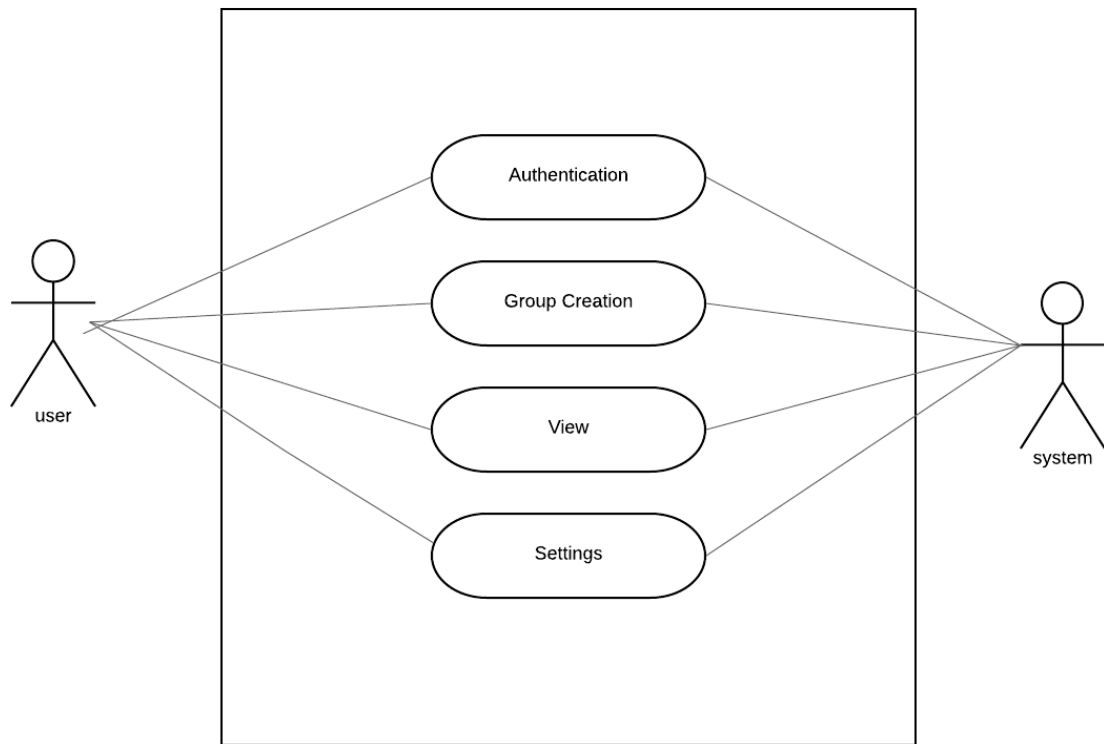


Figure 2: Level 1 Use Case Diagram

4.2.1.1 Authentication:

In this subsystem, the user inputs name, id and password where system verifies this information .. Once registration is completed, user needs to login to own account. Here, both user and system are primary actors. Figure 3 shows Authentication subsystem.

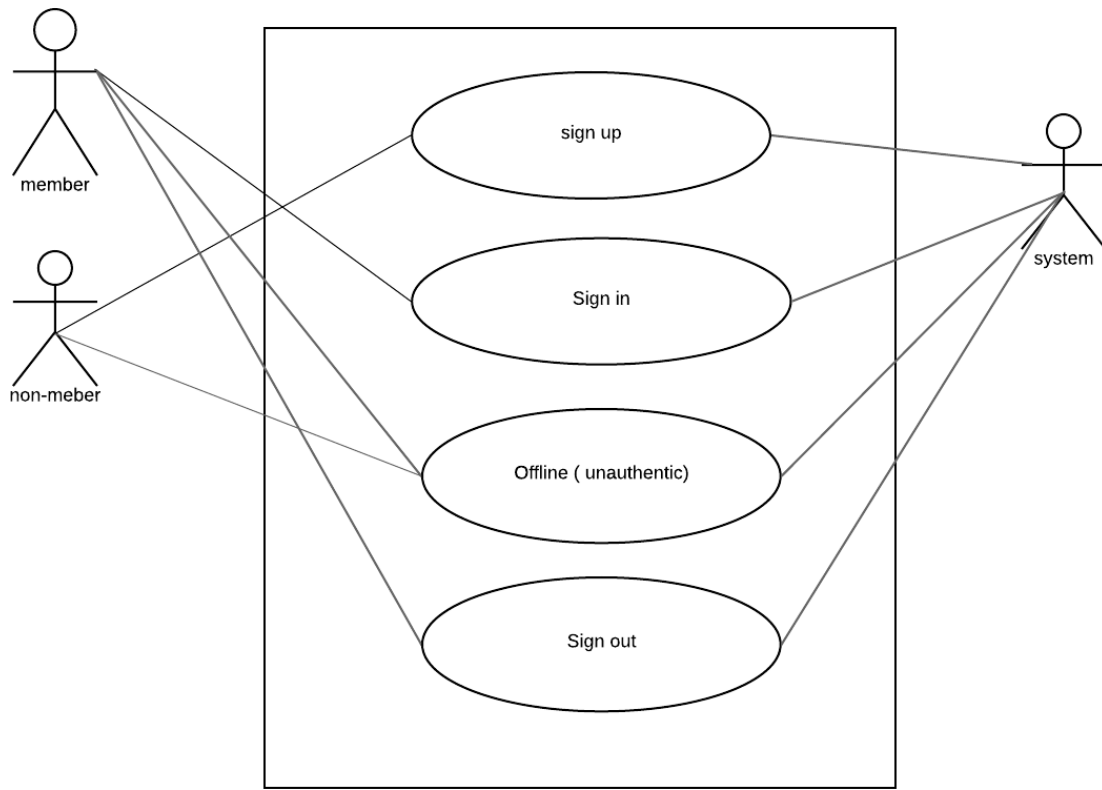


Figure 3: Level 1.1 Use Case Diagram of Authentication Subsyte

4.2.1.2 Group Creation:

In this subsystem, the user creates group by searching user id. In group , user can send request ,cancel request , accept request and remove anyone from group .

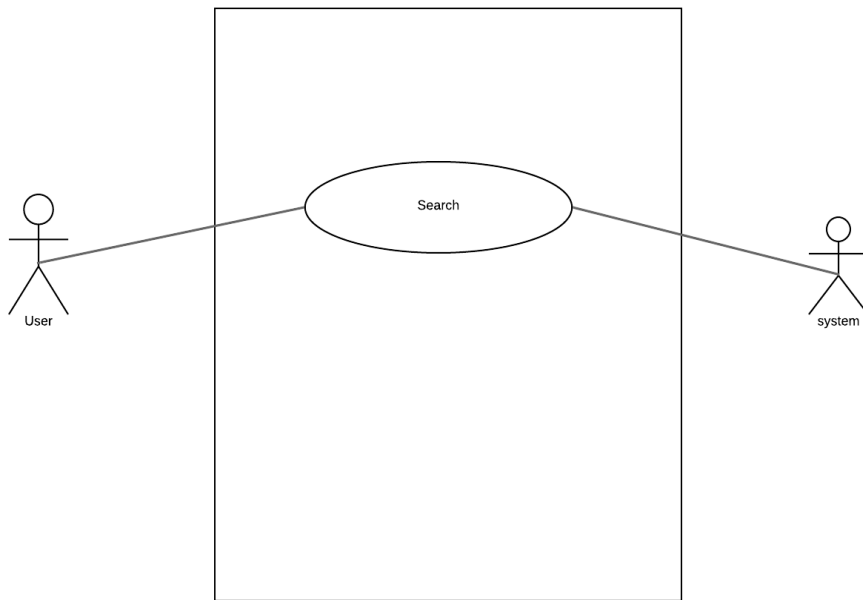


Figure 4: Level 2.2 Use Case Diagram of Group Creation Subsystem

4.2.1.2.1 Search:

In this subsystem, the user creates group by searching user id. In group , user can send request ,cancel request , accept request and remove anyone from group .

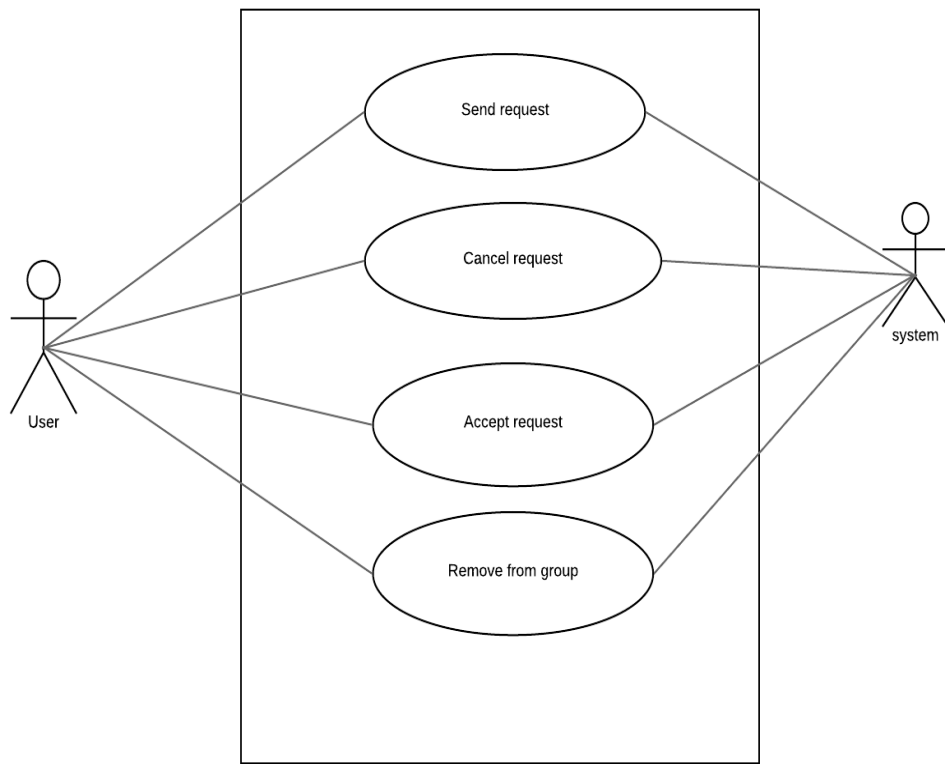


Figure 5: Level 2.3 Use Case Diagram of Search Subsystem

4.2.1.3 View:

In View subsystem, the users can view a map of locations which contains each user's current or last position. User can also view other user's profile as well as own profile. Following use case diagram in Figure 8 shows view subsystem works –

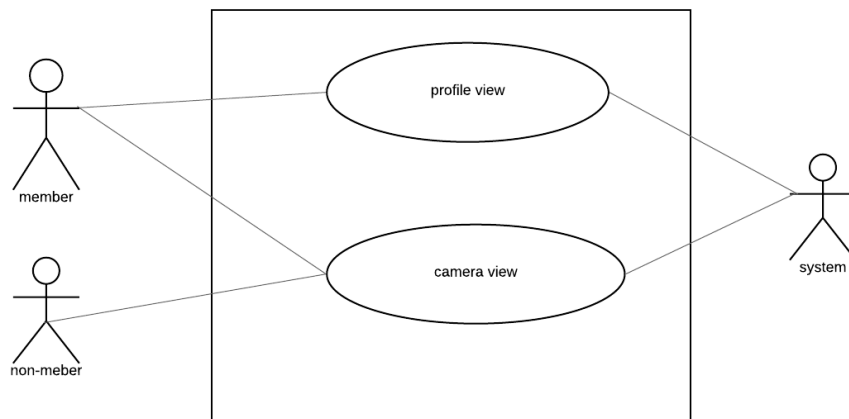


Figure 6: Level 1.3 Use Case Diagram of View Subsystem

4.2.1.4 Settings:

Settings subsystem lets user update own profile information along with check in mode - manual

or auto. In manual mode, after reaching a place user check in on own where in auto mode, system checks it automatically.

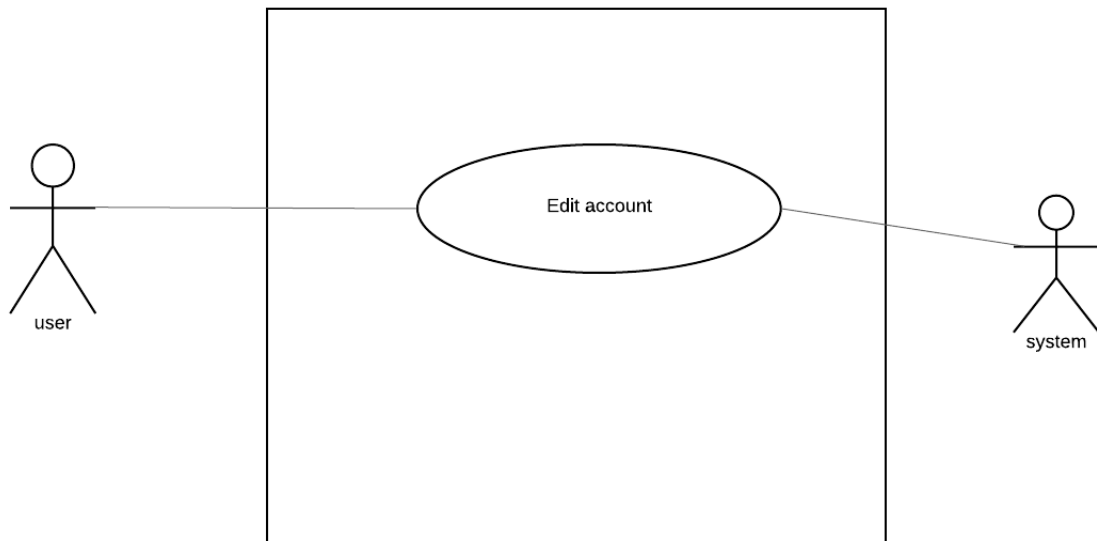


Figure 7: Level 1.4 Use Case Diagram of Settings Subsystem

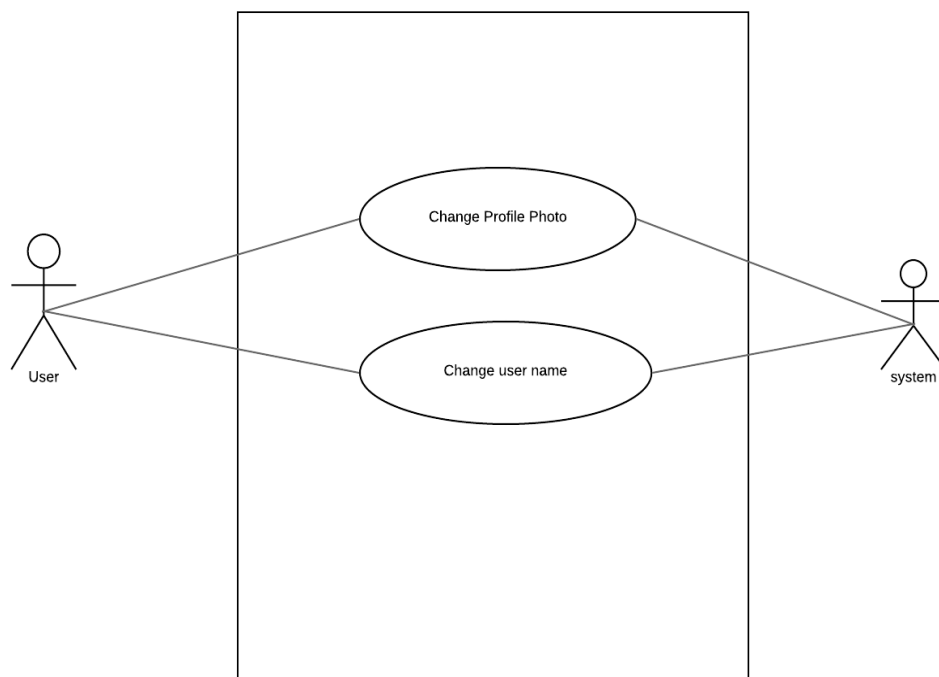


Figure 8: Level 2.4 Use Case Diagram of Edit Account Subsystem

4.3 Activity Diagrams of Camera Synchronization

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 of Sign Up:

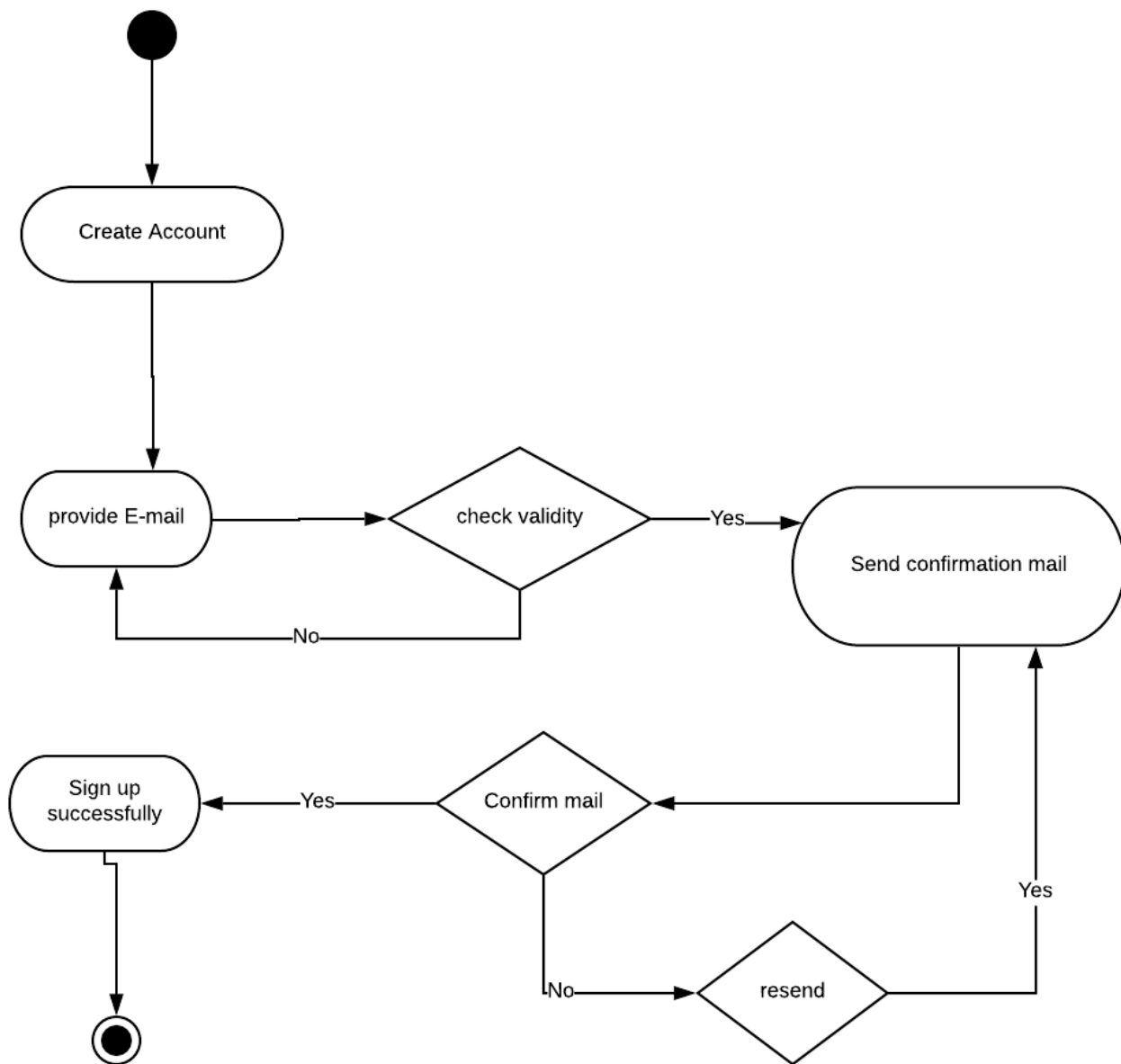


Figure 9: Level 4.3.1 Activity Diagram of Sign Up

4.3.2 Activity diagram of Sign In:

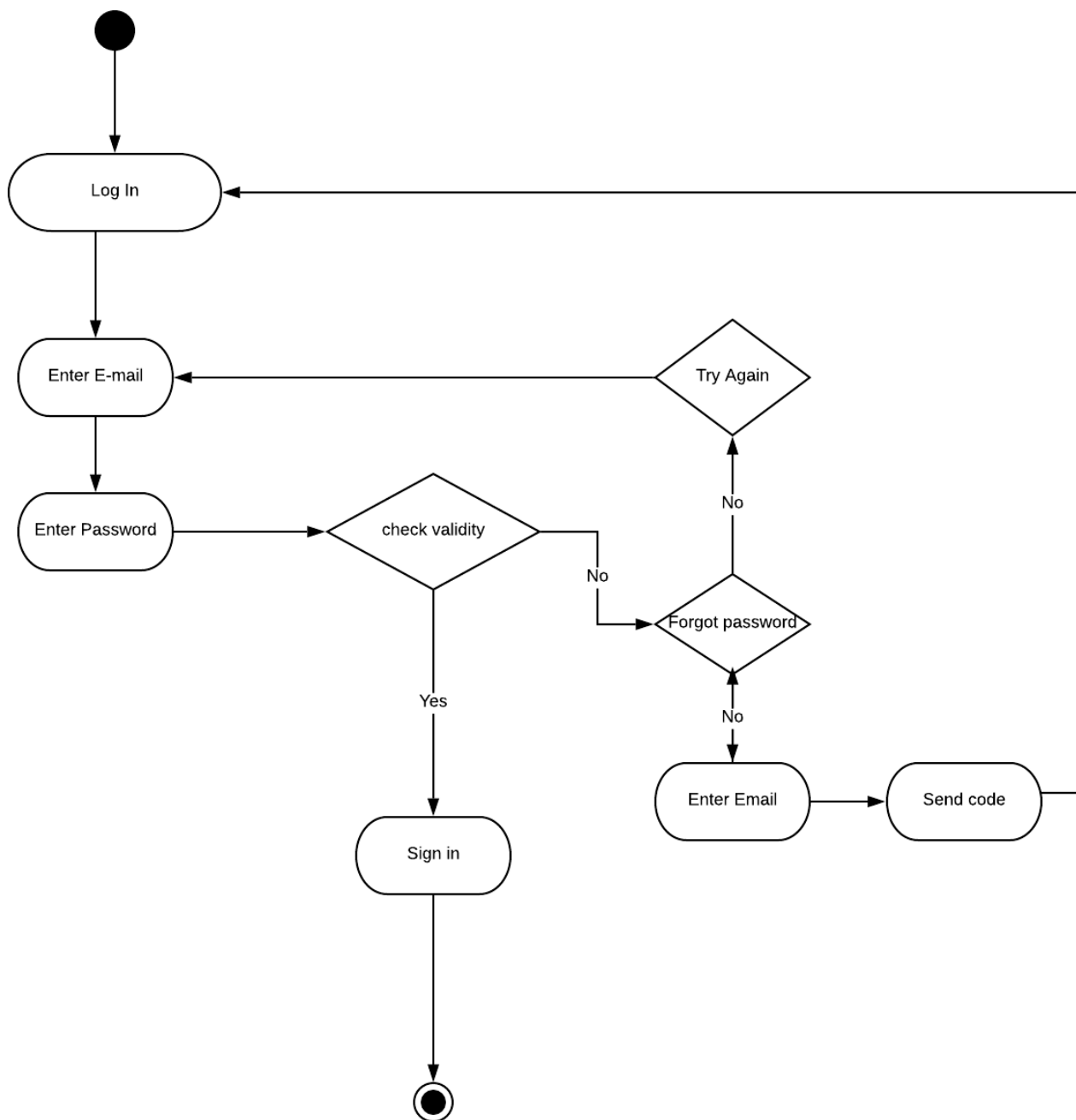


Figure 10: Level 4.3.2 Activity Diagram of Sign In

4.3.3 Activity diagram of Un-authentic:

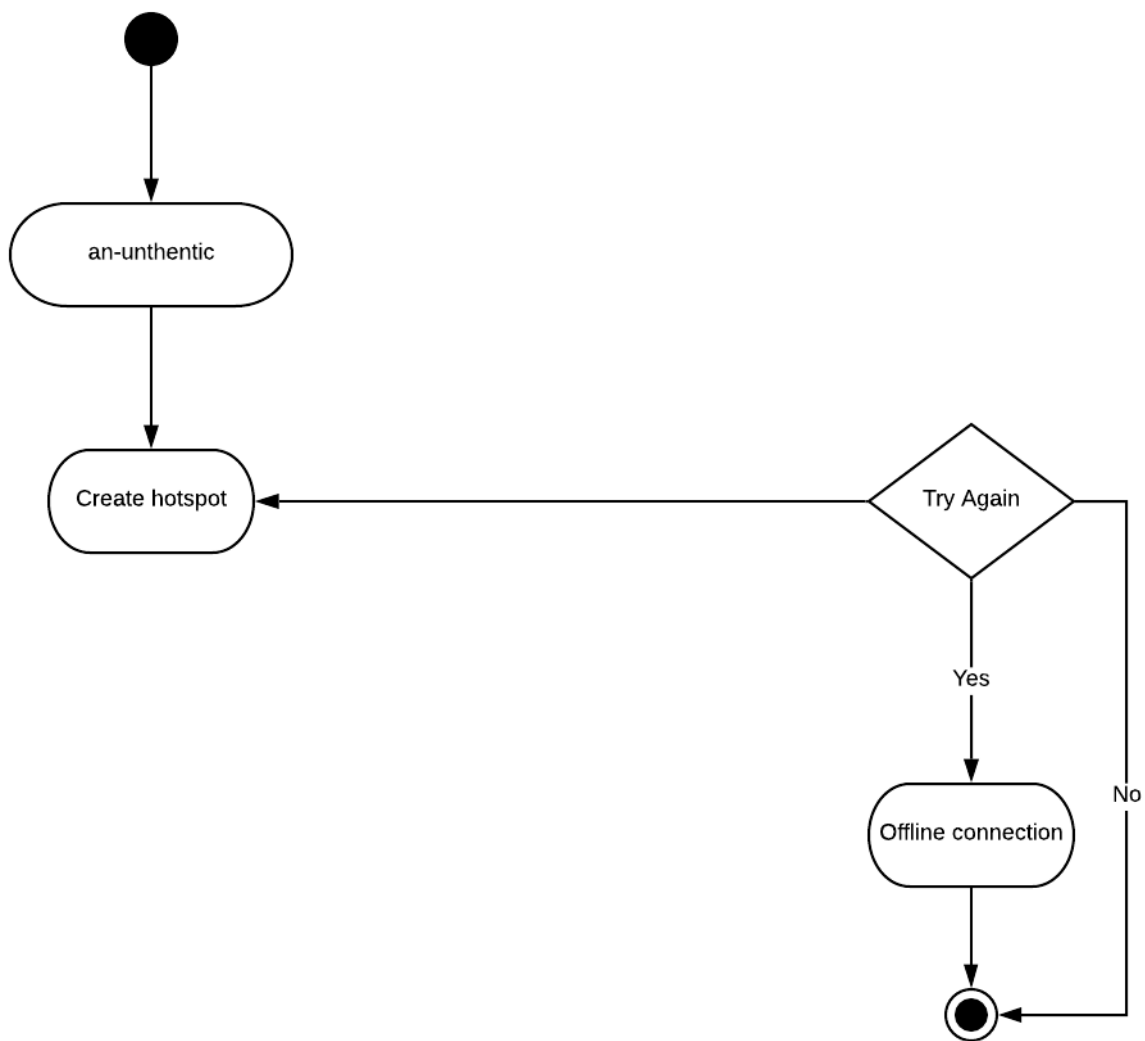


Figure 10: Level 4.3.2 Activity Diagram of An-authentic

4.3.4 Activity diagram of Sign Out:

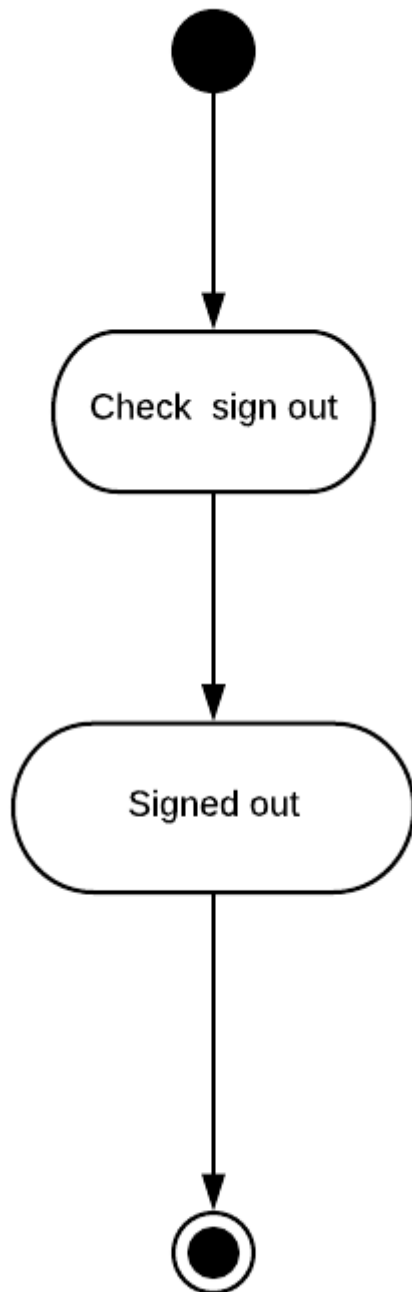


Figure 11: Level 4.3.3 Activity Diagram of Sign out

4.3.4 Activity diagram of Send Request:

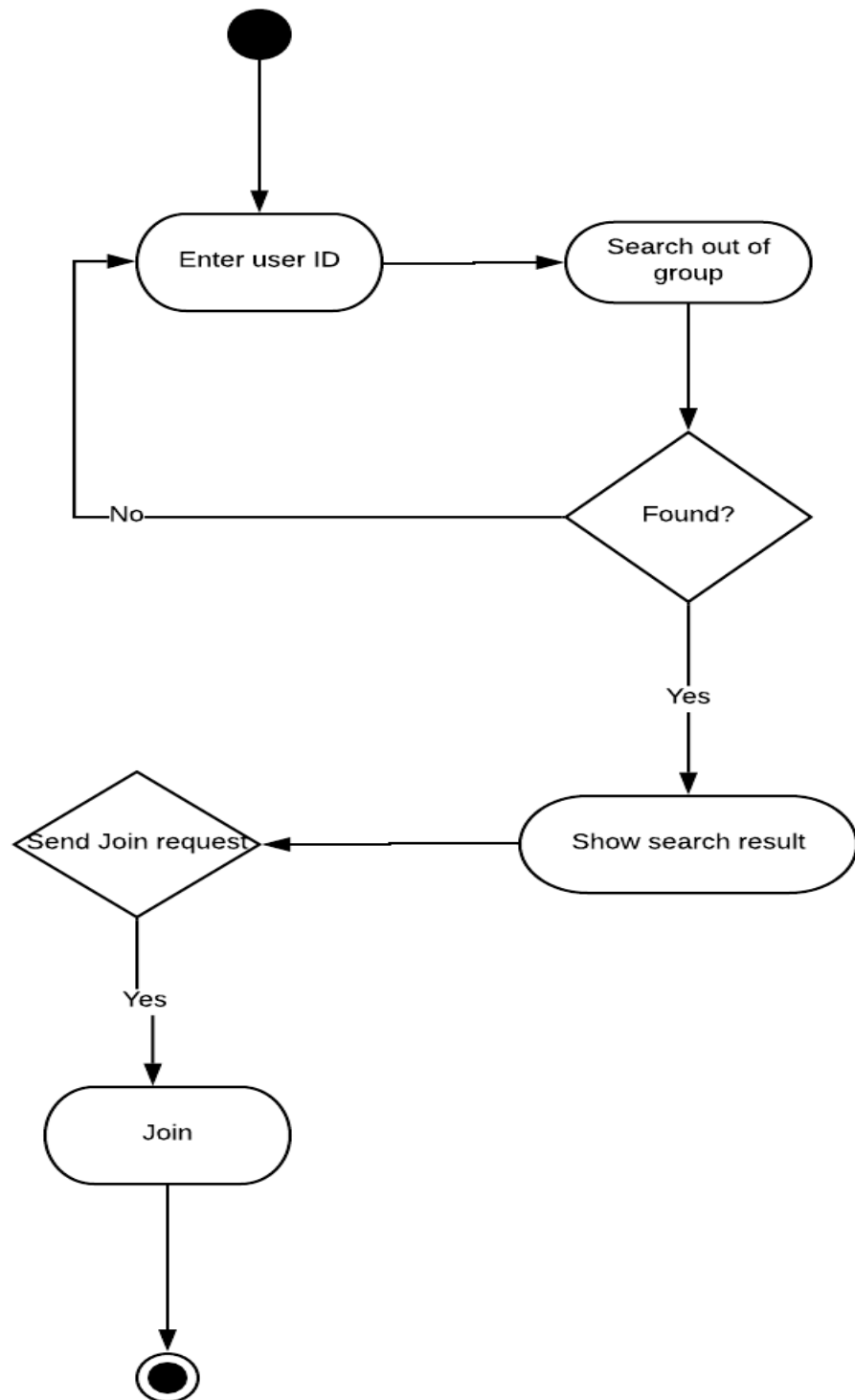


Figure 12: Level 4.3.4 Activity Diagram of Send Request

4.3.5 Activity diagram of Cancel Request:

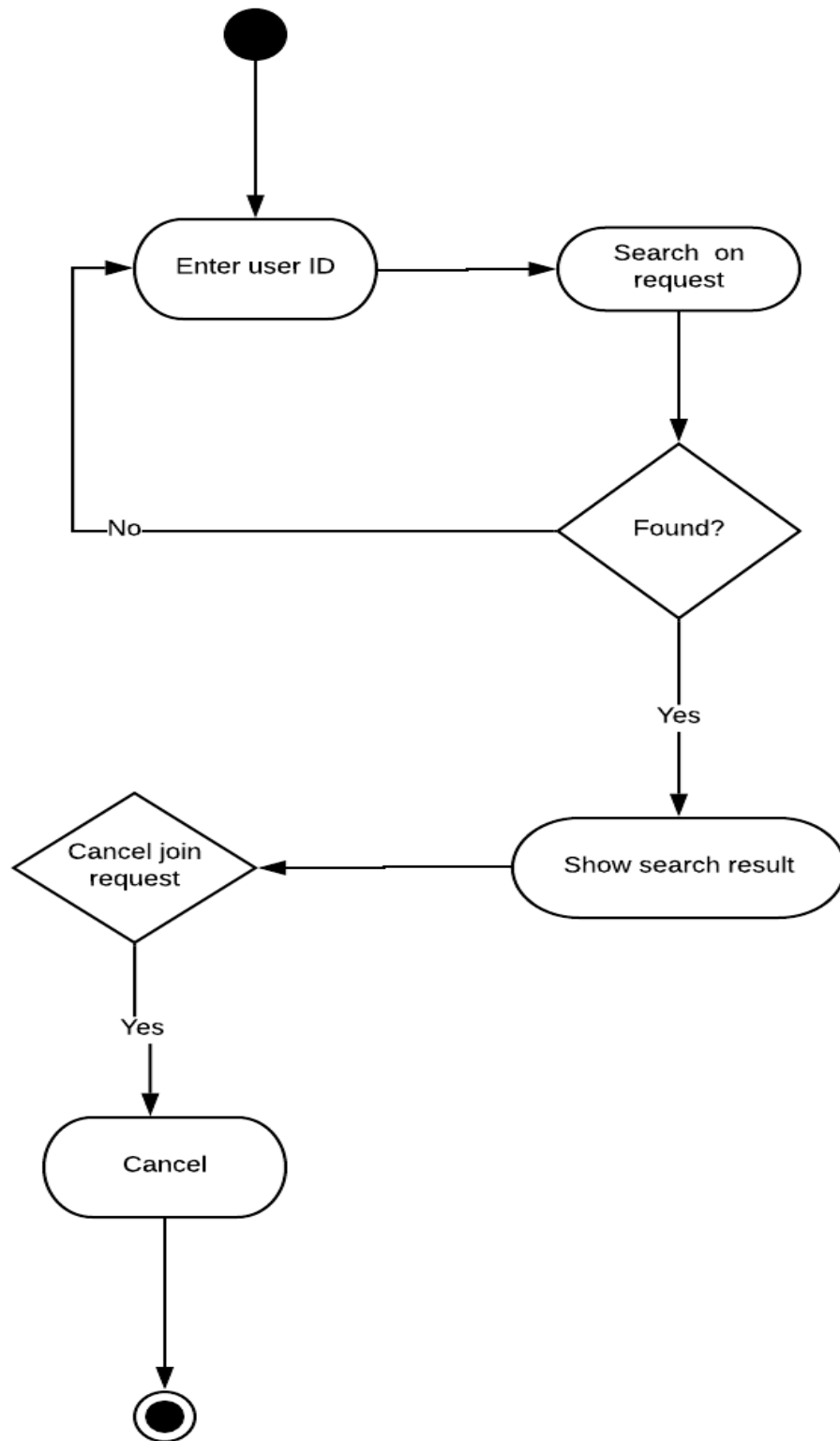
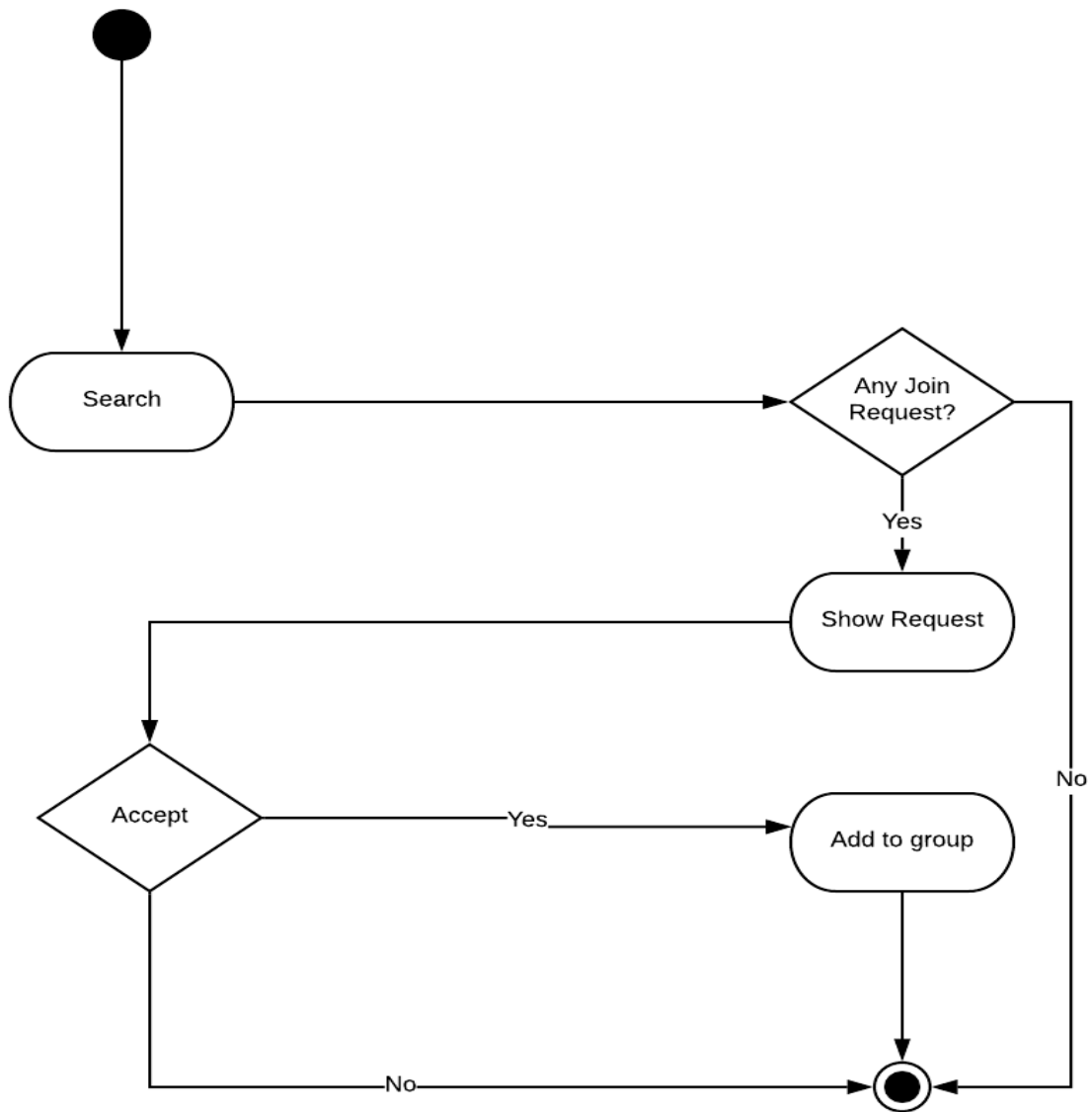


Figure 13: Level 4.3.4 Activity Diagram of Cancel Request

4.3.6 Activity diagram of Accept Request:



4.3.5 Activity Diagram of Accept Request

Figure 14: Level

4.3.7 Activity diagram of Remove from group:

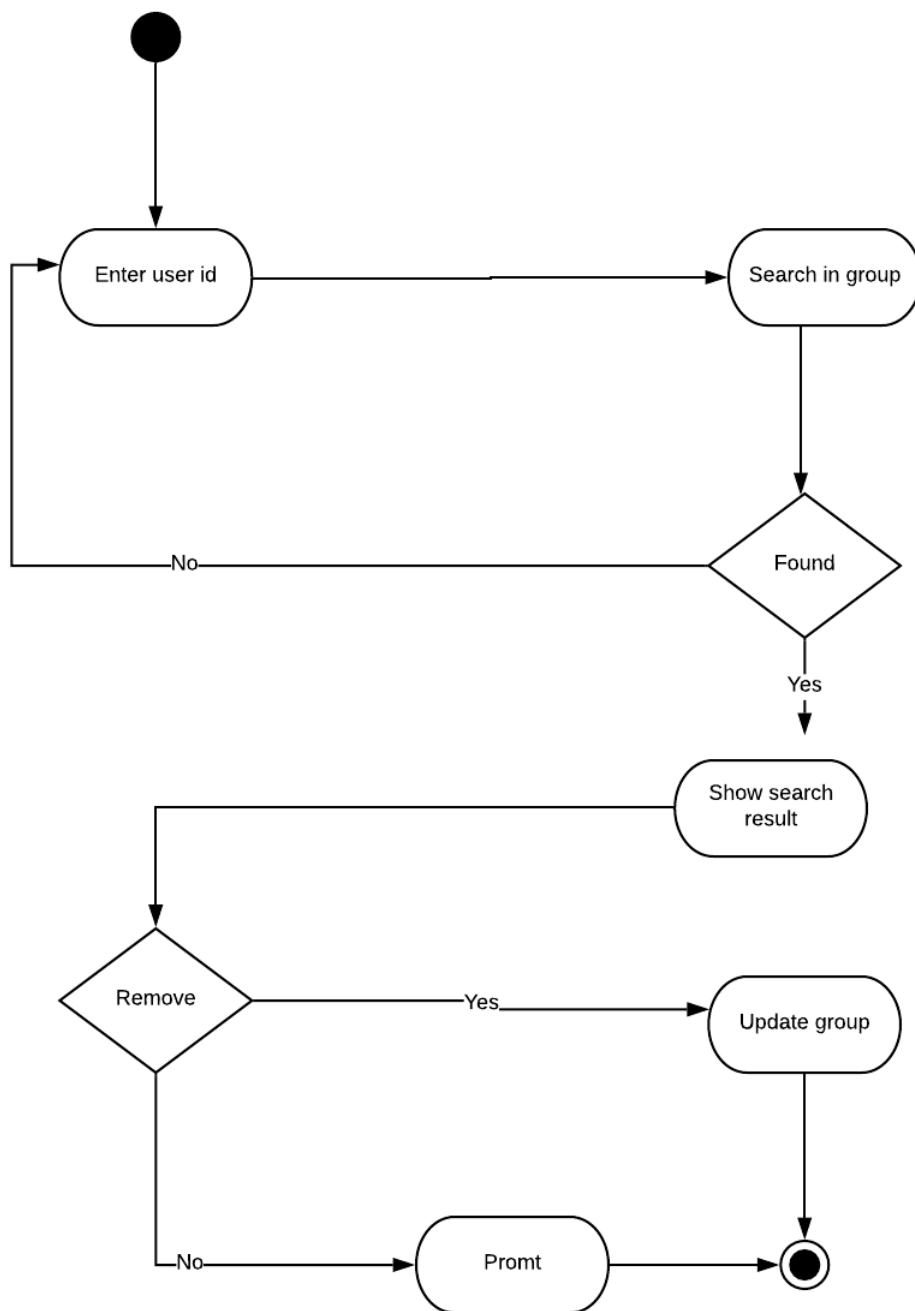


Figure 15: Level 4.3.4 Activity Diagram of Remove from group

4.3.8 Activity diagram of Profile View:

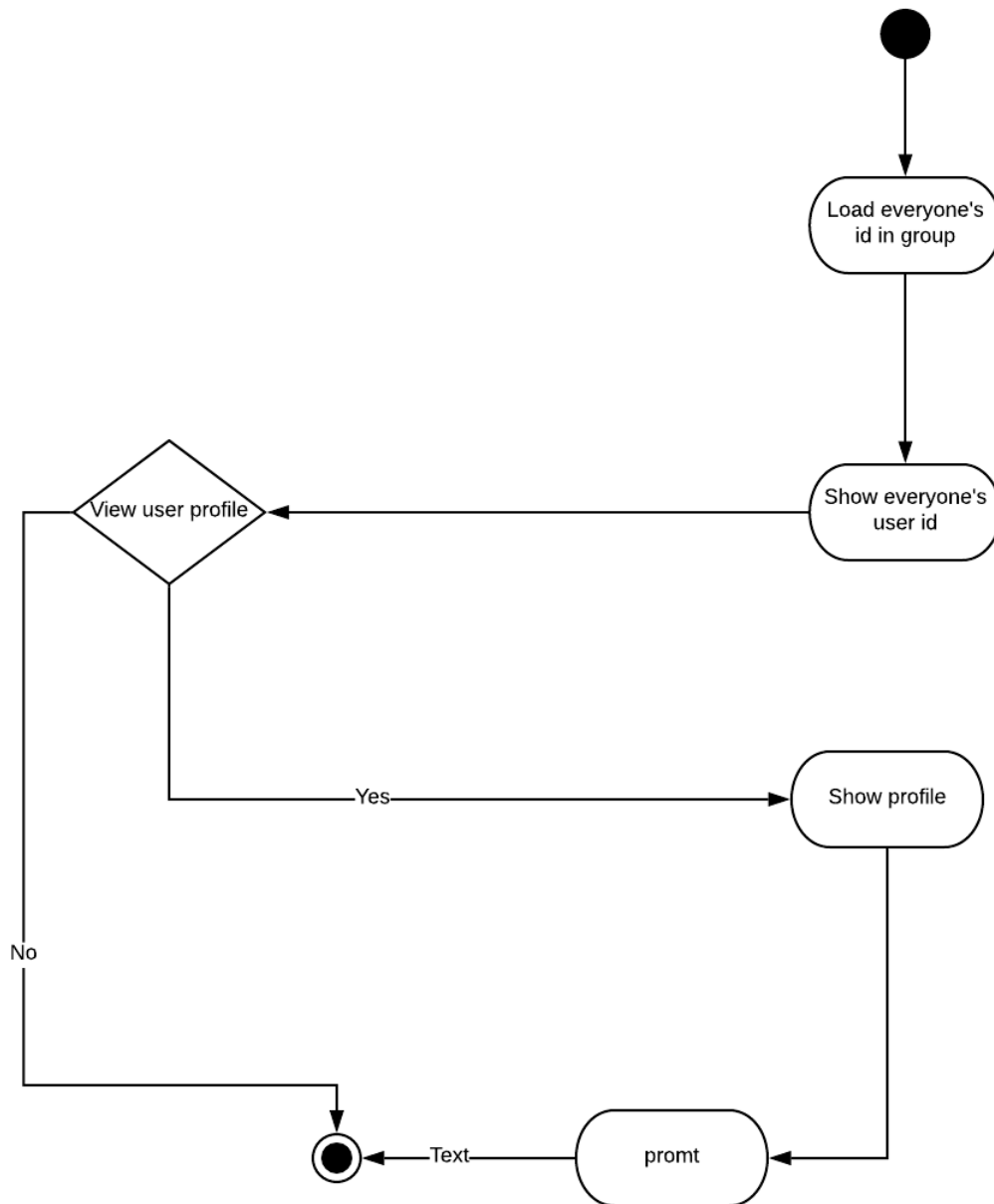


Figure 16: Level 4.3.4 Activity Diagram of profile view

4.3.9 Activity diagram of Camera View:

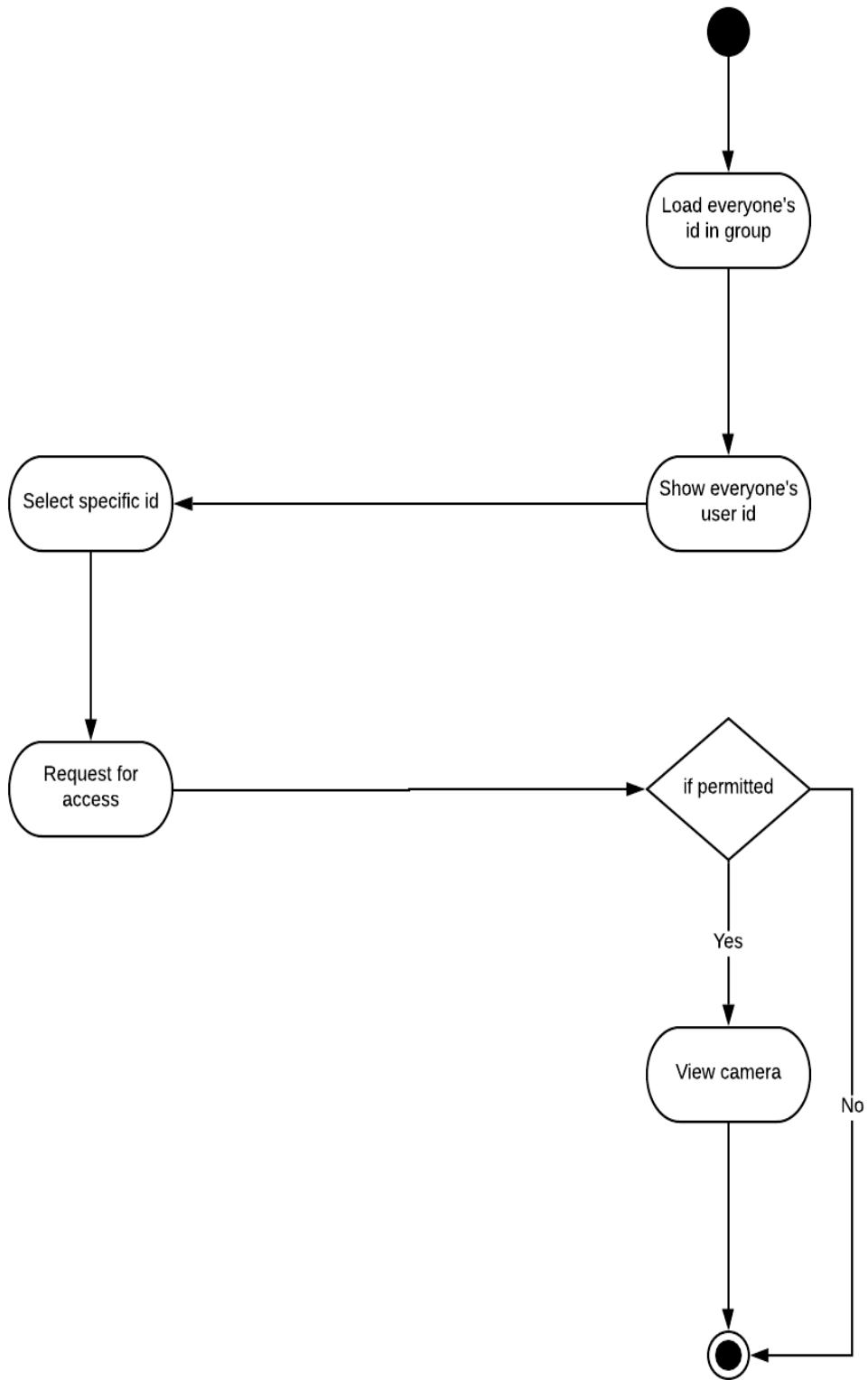


Figure 17: Level 4.3.4 Activity Diagram of Camera View

4.3.10 Activity diagram of Change Profile Photo:

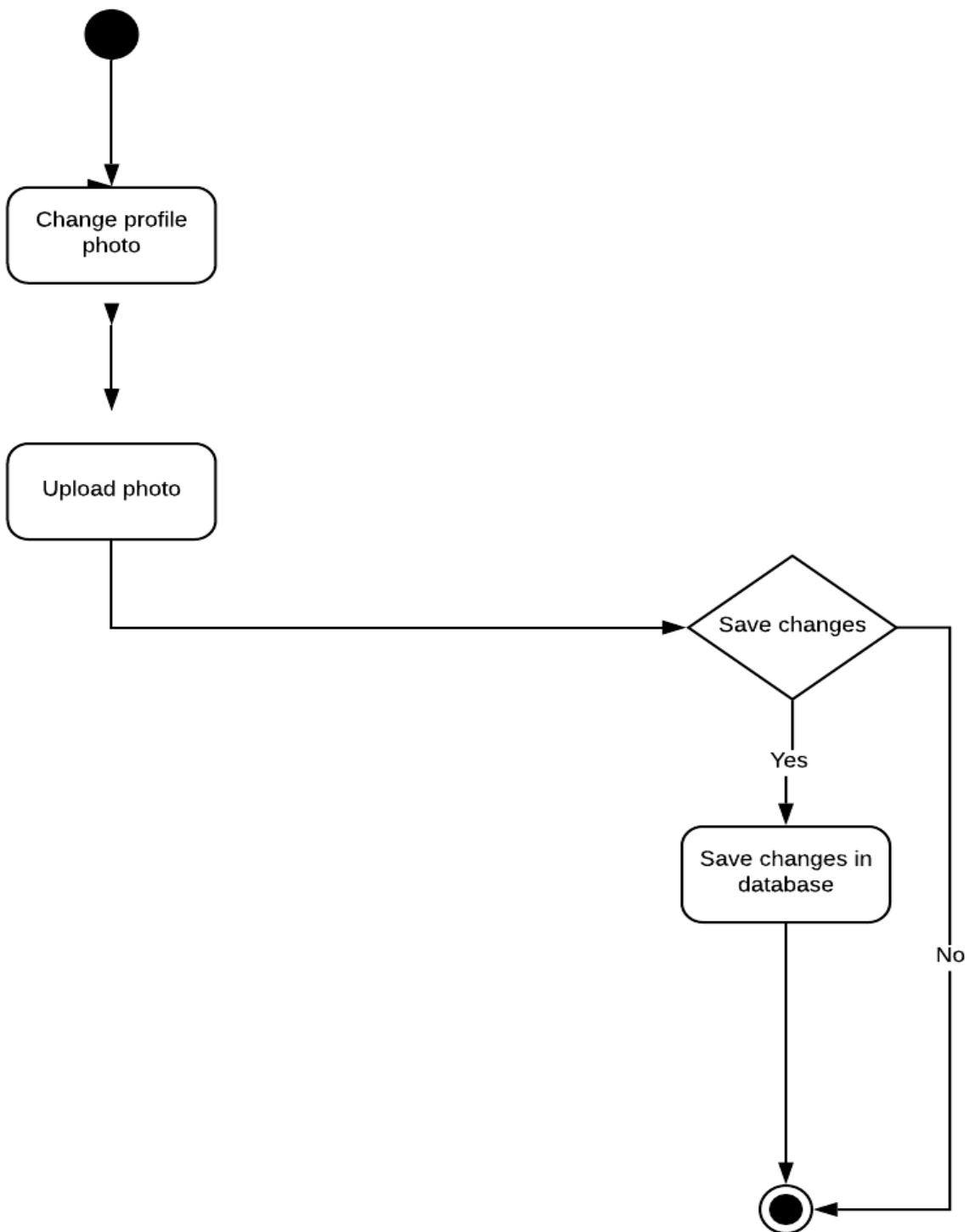


Figure 18: Level 4.3.4 Activity Diagram of Change profile photo

4.3.11 Activity diagram of Change User Name:

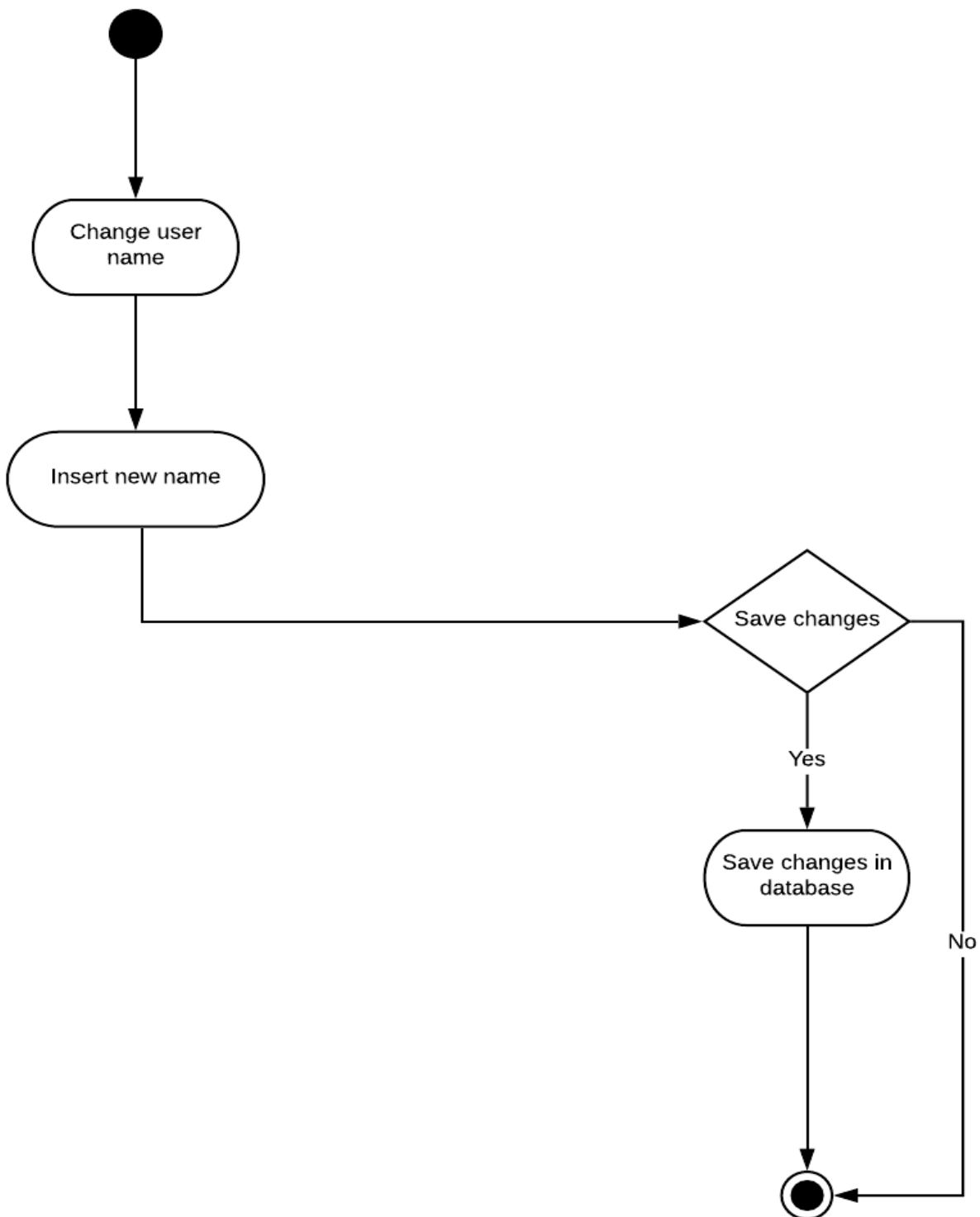
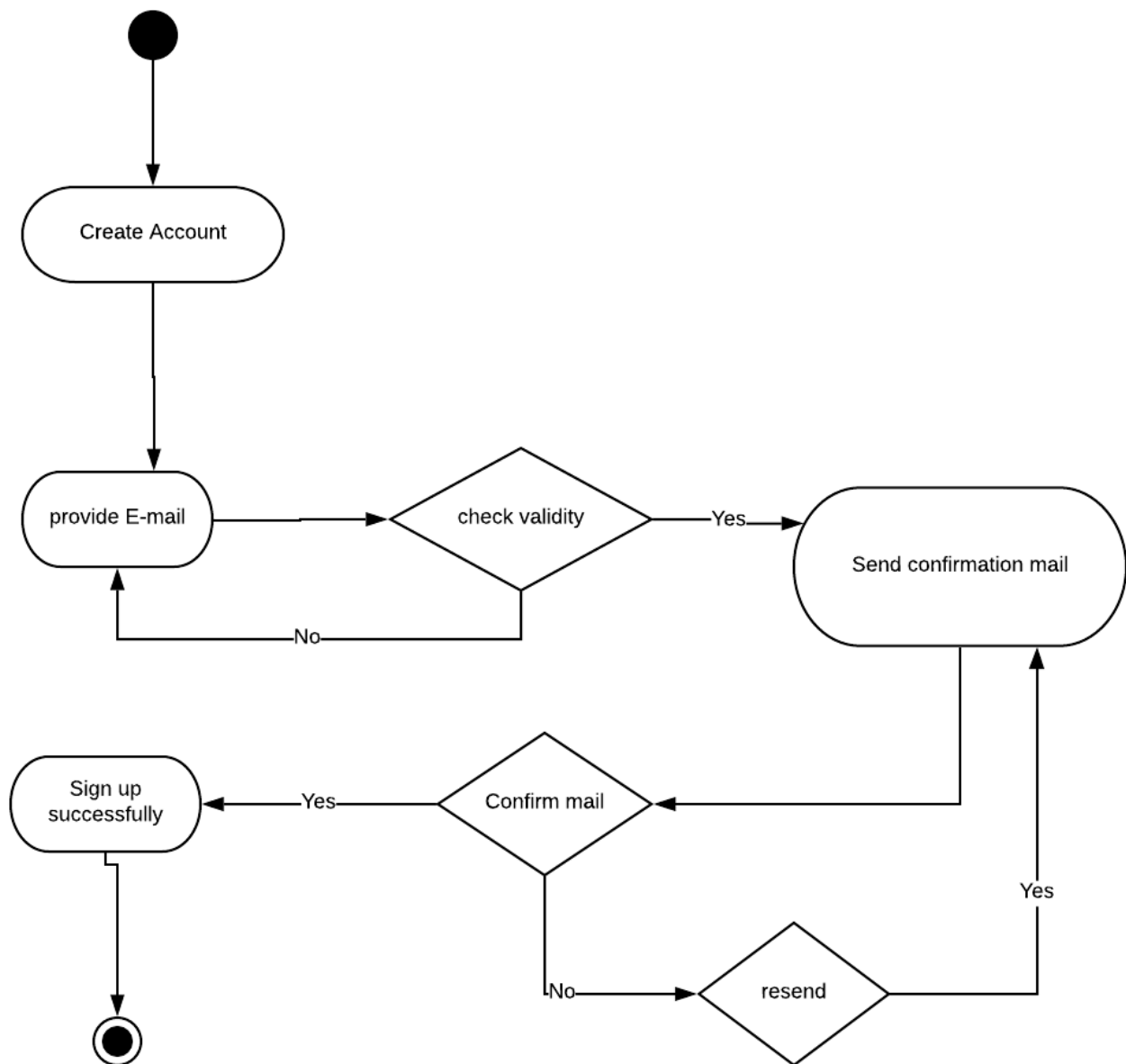


Figure 18: Level 4.3.4 Activity Diagram of Change user name

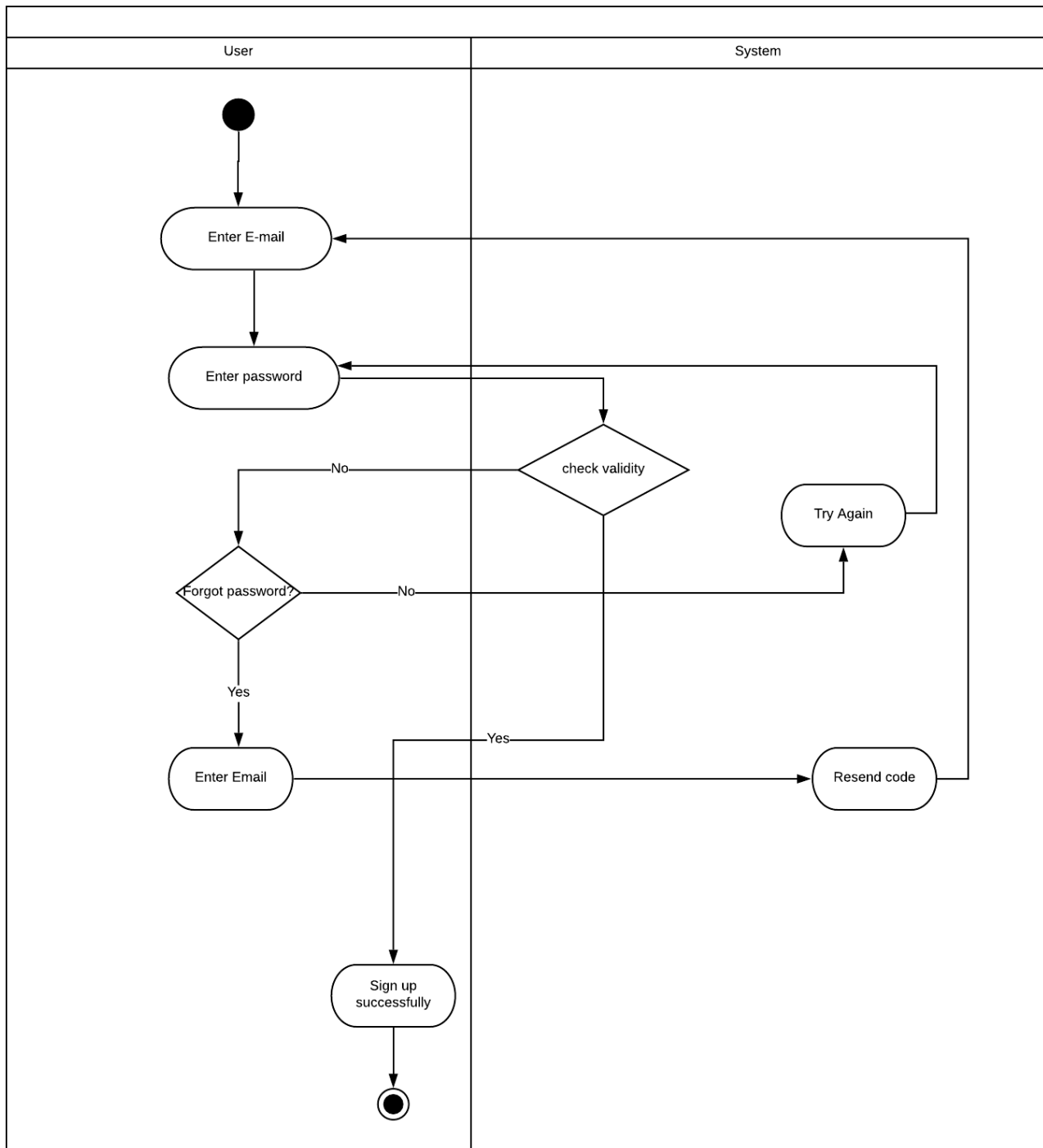
4.4 Swim-lane diagram of Camera Synchronization

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:

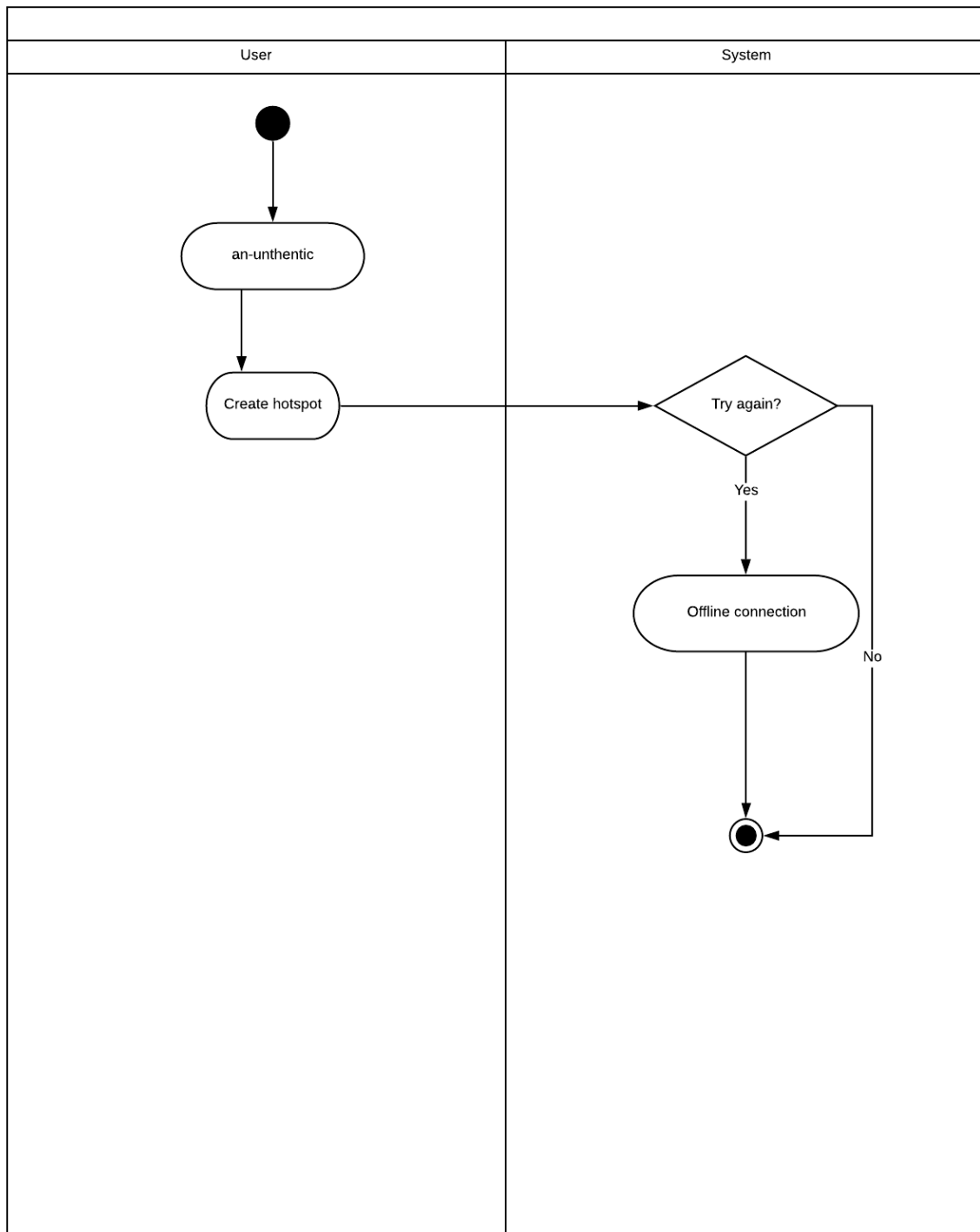
4.4.1 Swim-lane diagram of Sign Up



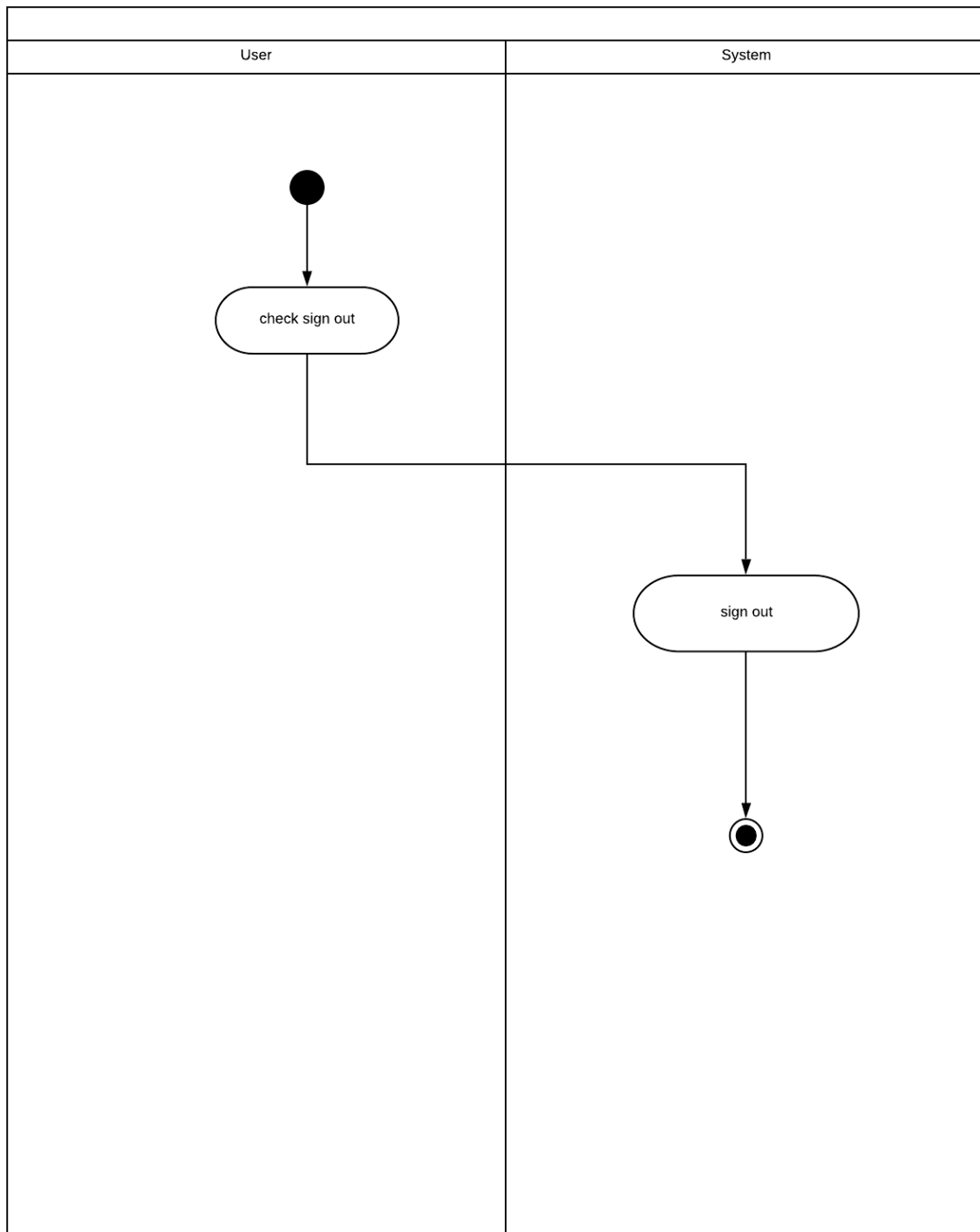
4.4.2 Swim-lane diagram of Sign In



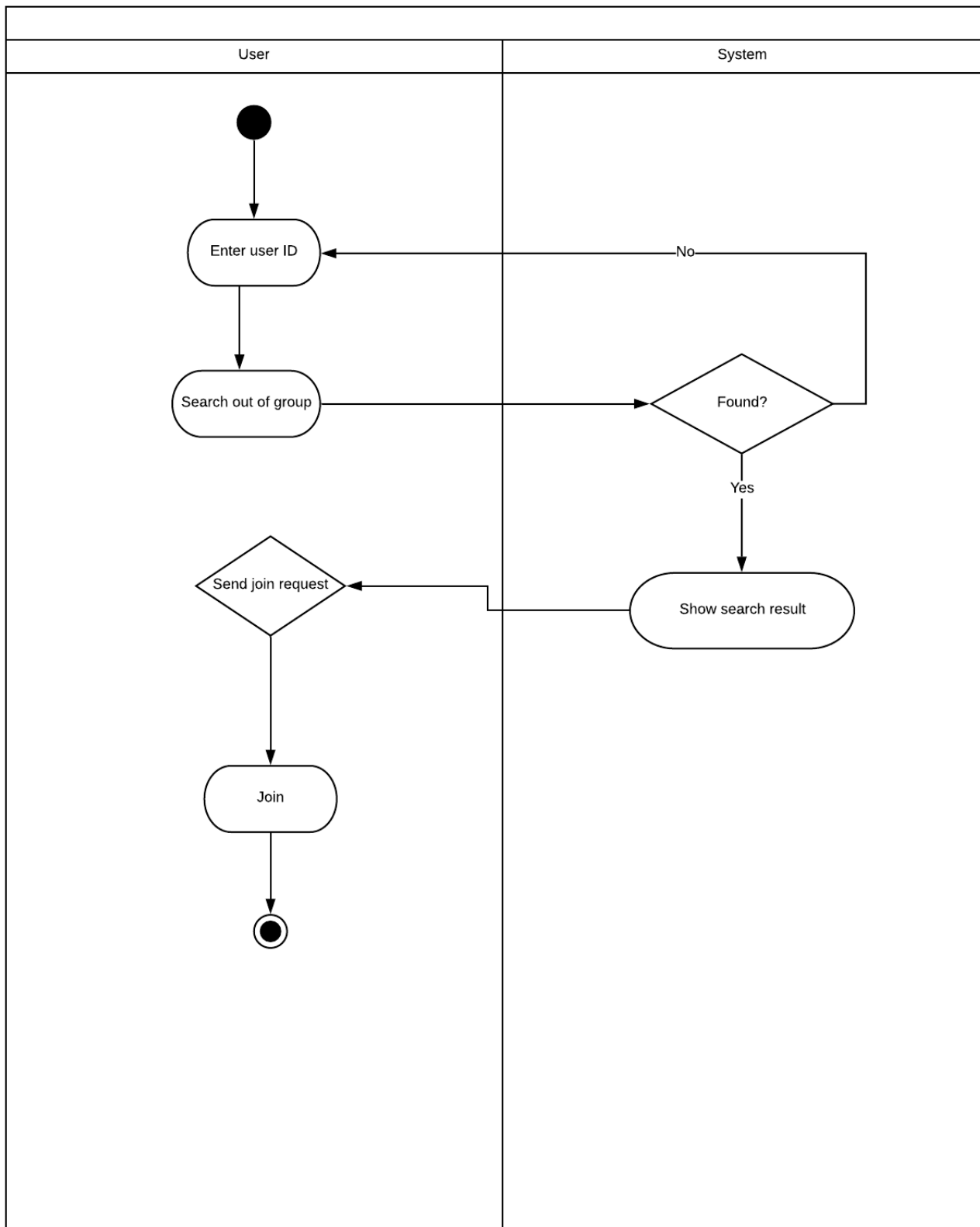
4.4.3 Swim-lane diagram of Un-authentic



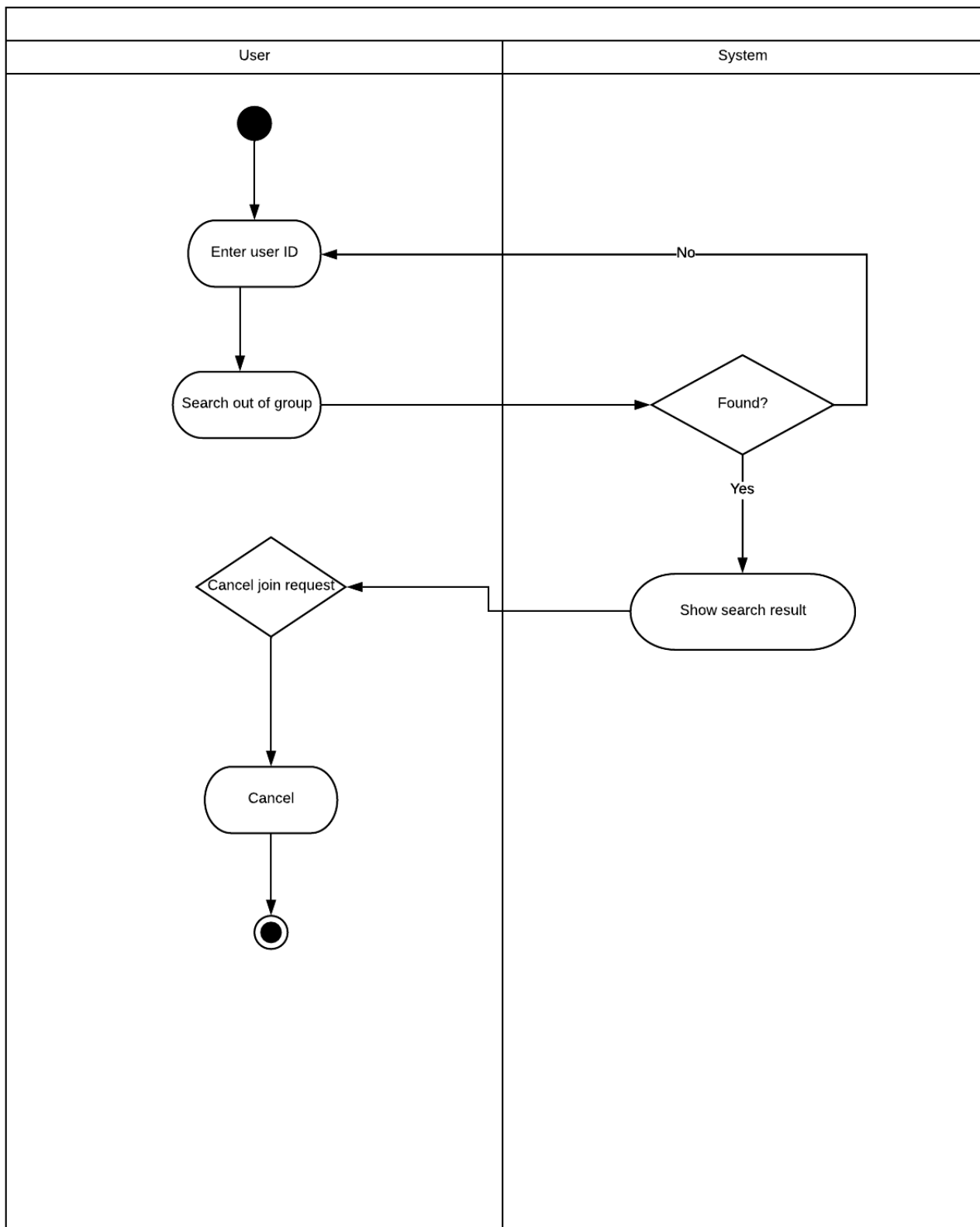
4.4.4 Swim-lane diagram of Sign out



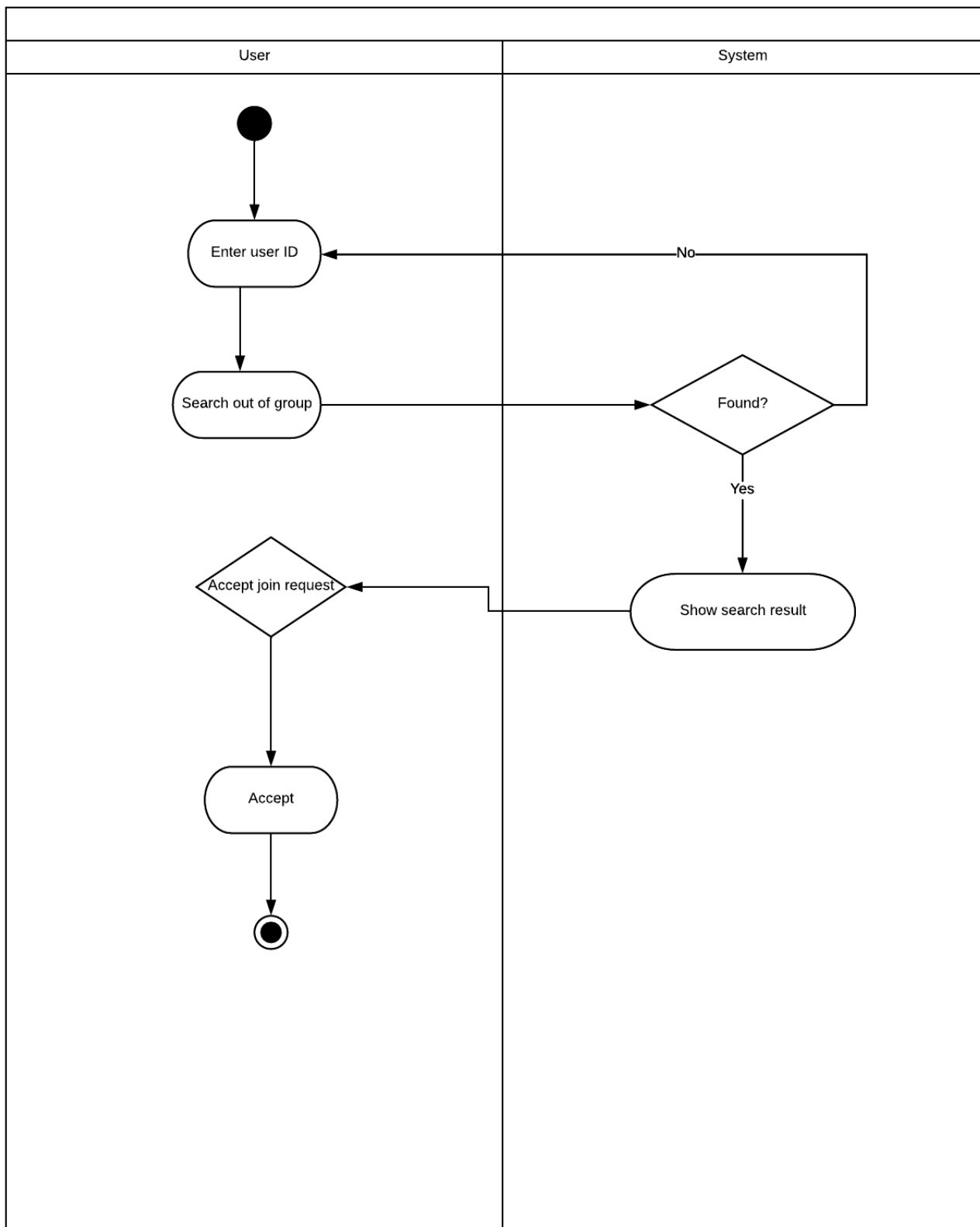
4.4.5 Swim-lane diagram of Send Request



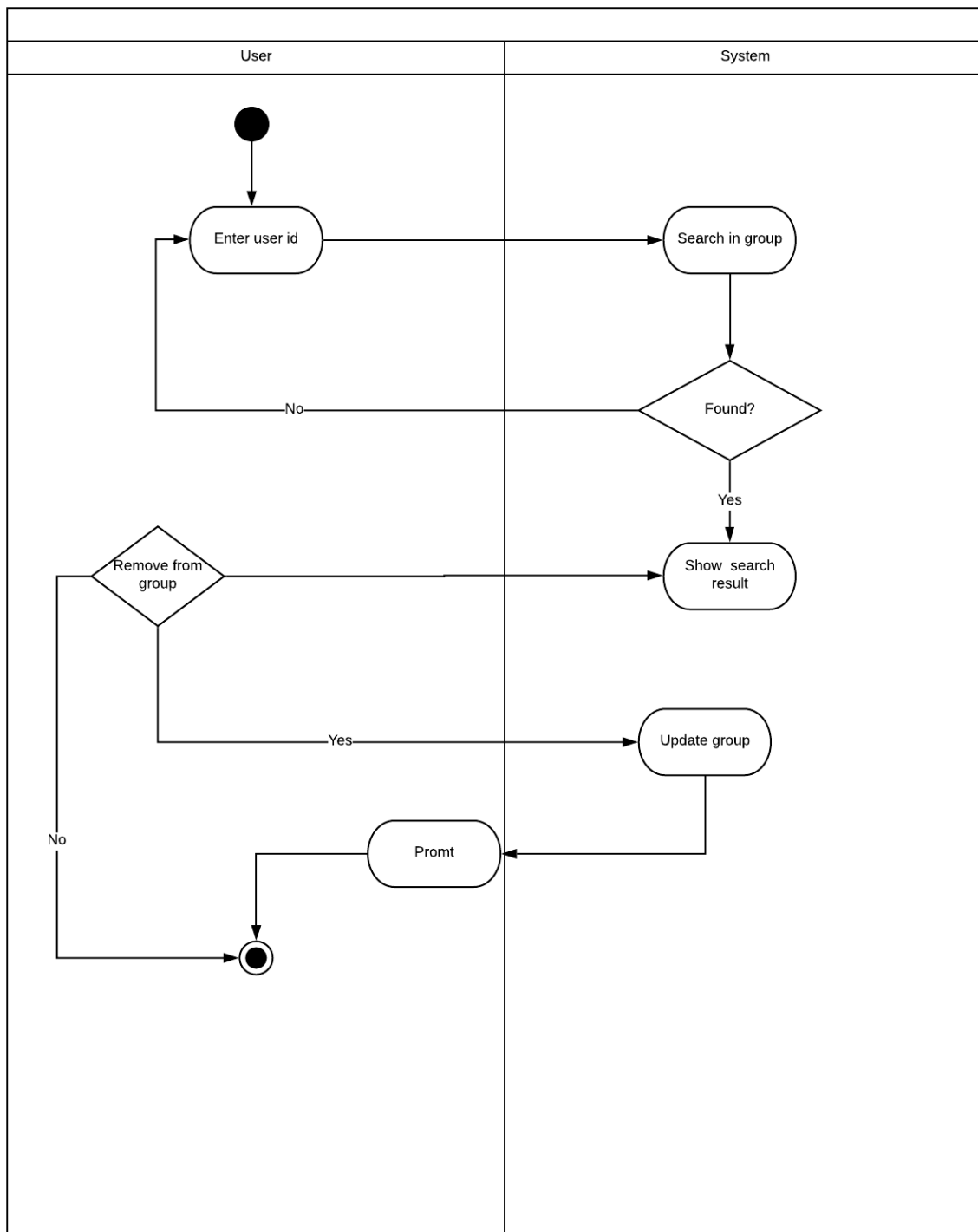
4.4.7 Swim-lane diagram of Cancel Request



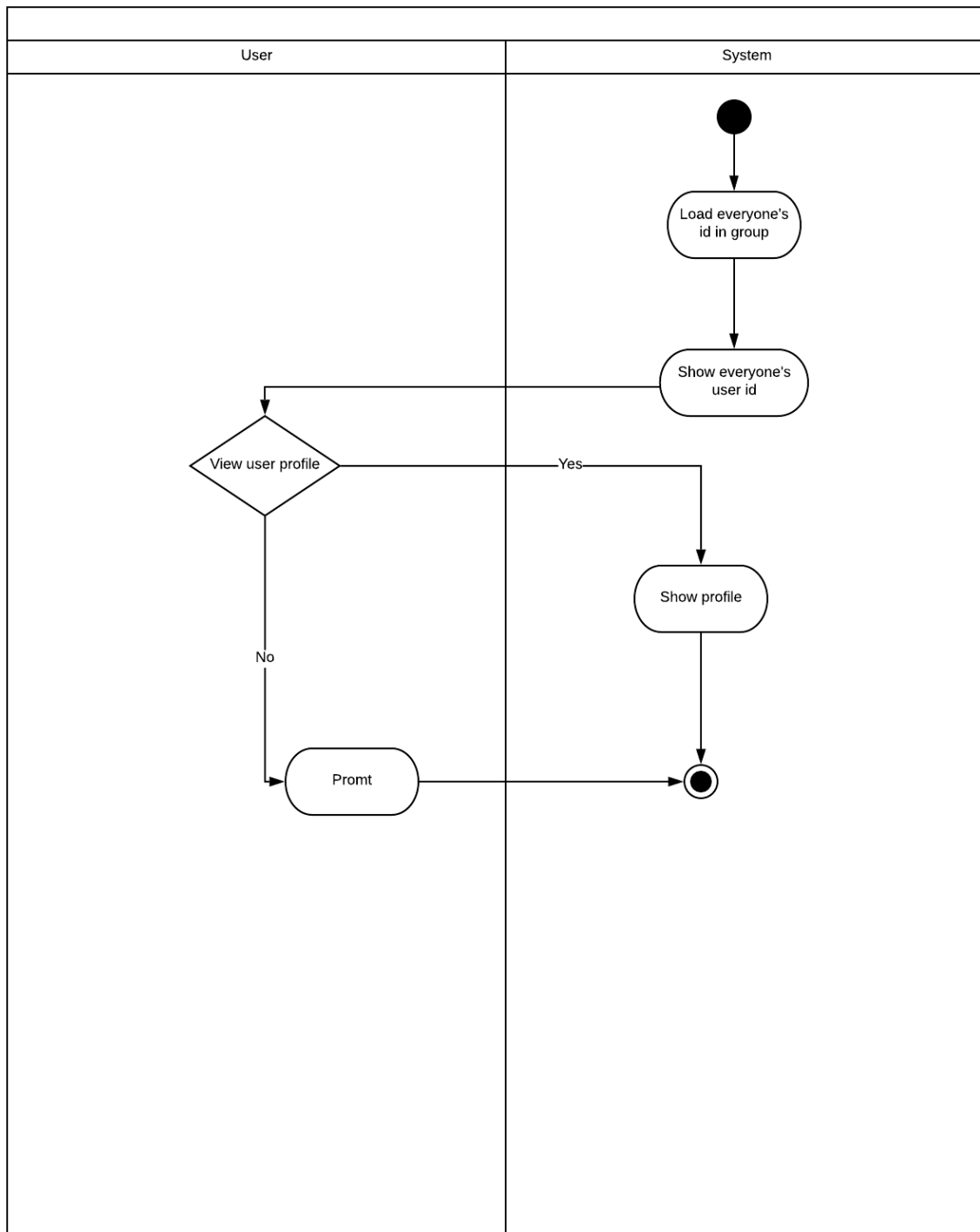
4.4.8 Swim-lane diagram of Accept Request



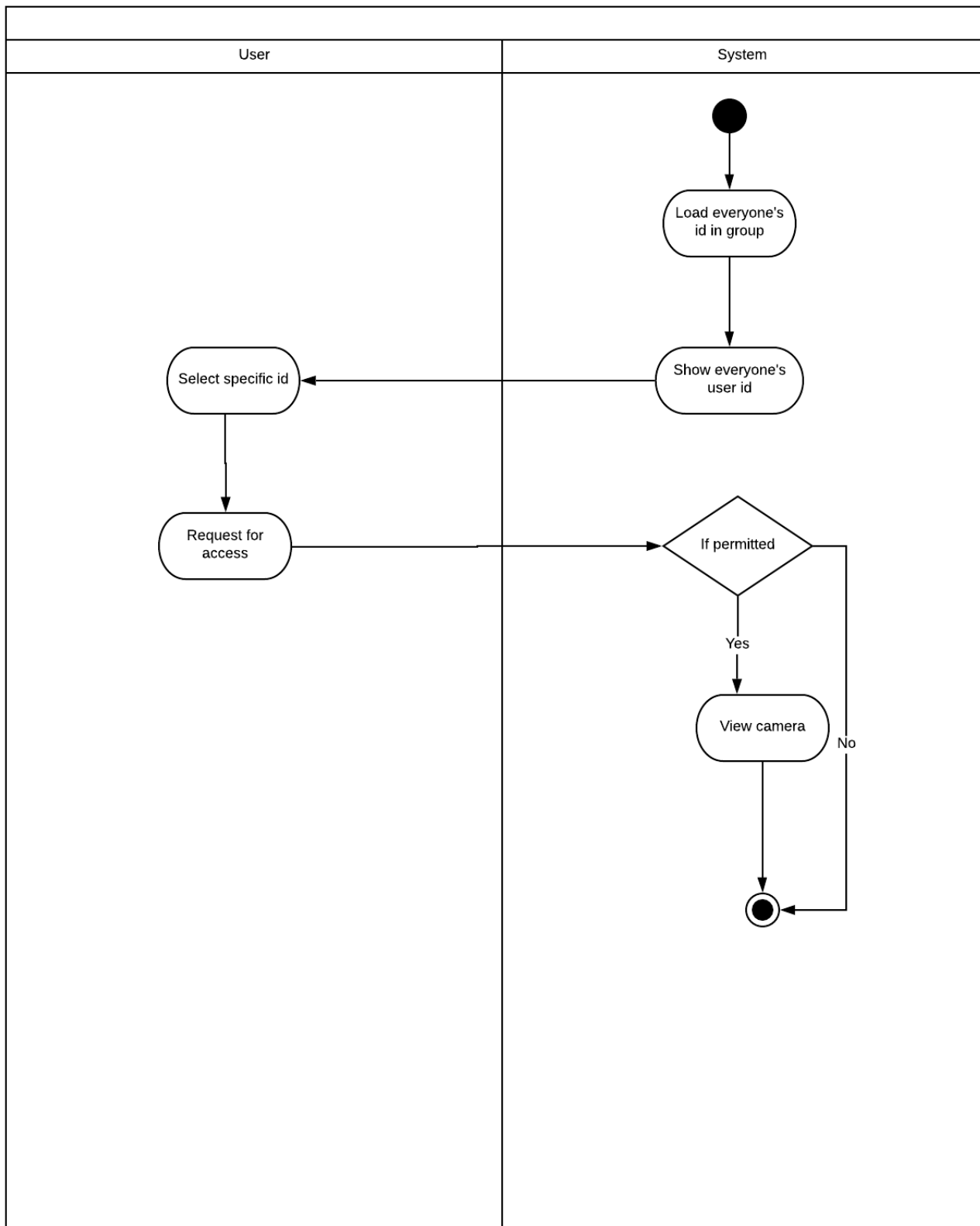
4.4.9 Swim-lane diagram of Remove from group



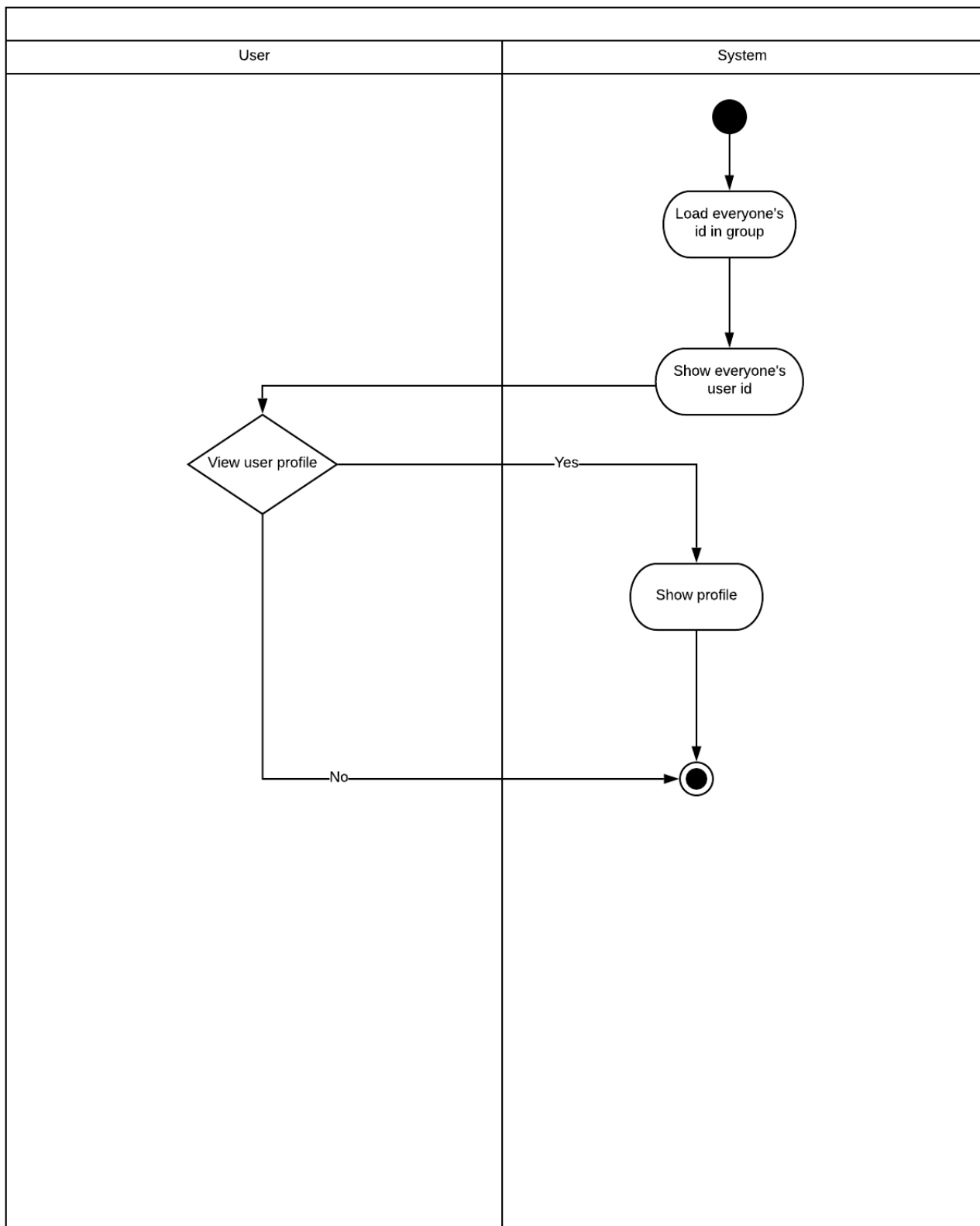
4.4.10 Swim-lane diagram of Profile View



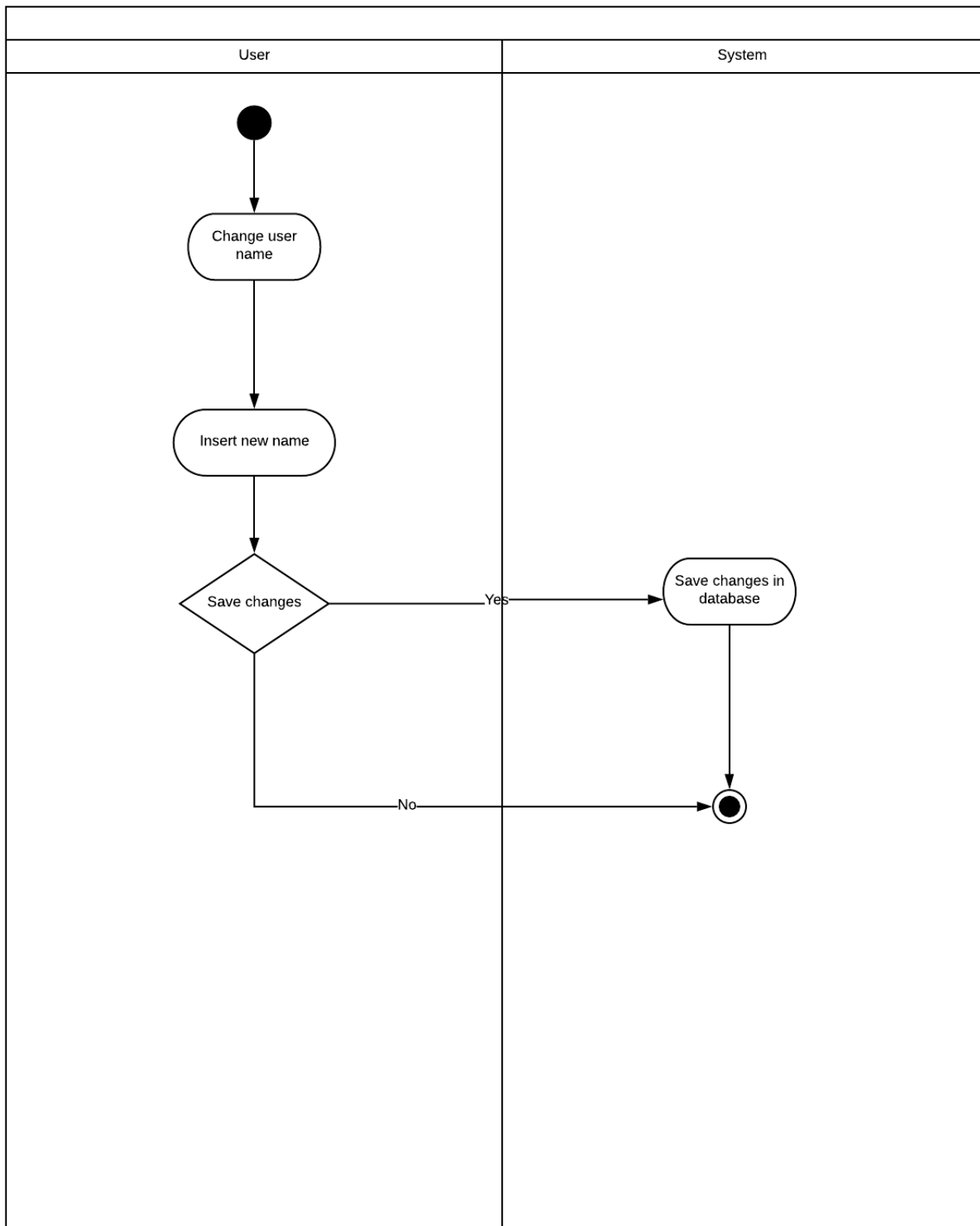
4.4.11 Swim-lane diagram of Camera View



4.4.12 Swim-lane diagram of Change profile photo



4.4.13 Swim-lane diagram of Change User name



Chapter Five

Data Model

5.1 Introduction

A data model is a conceptual representation of data structures (tables) required for a database and is very powerful in expressing and communicating the business requirements. In this chapter, we will discuss about the data model for Camera Synchronization.

5.2 Data object

A data object is a representation of information which has different properties or attributes that must be understood by software. We have found the following data objects in Camera Synchronization:

5.2.1 Noun Identify

Nouns are identified from the usage scenario and shown in the table below:

Table 1: Identifying Potential Data Objects

No.	Noun/Noun Phrases	Problem Domain/Solution Domain	Attributes
1.	Create Group	P	18,20,30,31,32,35,36
2.	Join Group	P	14,38,19,
3.	Group Member	S	6,5,14,26
4.	User	P	6,7
5.	Username	S	
6.	Id	S	
7.	Password	S	
8.	Account	S	

9.	System	P	11,12,35,36
10.	Information	S	
11.	Logged in	S	
12.	Logged out	S	
13.	Connection	P	
14.	Request	P	
15.	View	S	
16.	Online	S	
17.	Profile	P	40,41,26,5
18.	QR Code Sharing	S	
19.	Scan QR Code	S	
20.	Member	P	6,5,14,26,35,36
21.	Non Member	P	
22.	Photo	S	6, 44
23.	Capture	S	
24.	Synchronization	P	
25.	Offline	S	
26.	Profile photo	P	
27.	Settings	P	
28.	Database	P	
29.	Camera	P	

30.	Search ID	S	
31.	Search Name	S	
32.	IP Address	P	
33.	Photos	P	
34.	Switch	P	
35.	Accept	P	
36.	Decline	P	
37.	Remove	P	
38.	Sent	P	
39.	Connection Type	P	
40	Email	S	
41	Phone	S	
42	Profile Type	P	16,25
43	Authentication	P	
44	Date	S	

5.2.2 Prospective Data Objects:

User: User Id, User Password

General People Profile: Profile Photo, Name, WorkPlace, About, Email

Journalist Profile: Profile Photo, Name, WorkPlace, About, Email

Professional Photographer Profile: Profile Photo, Name, WorkPlace, About, Email

Group: Group Id, Grp Password, Group type

Gallery: Photo Id, Date, Photo

5.2.3 Analysis on Prospective Data Objects:

General People Profile, Journalist Profile, Professional Photographer Profile have same attributes. So we can merge them into Profile.

5.2.4 Actual Data Objects

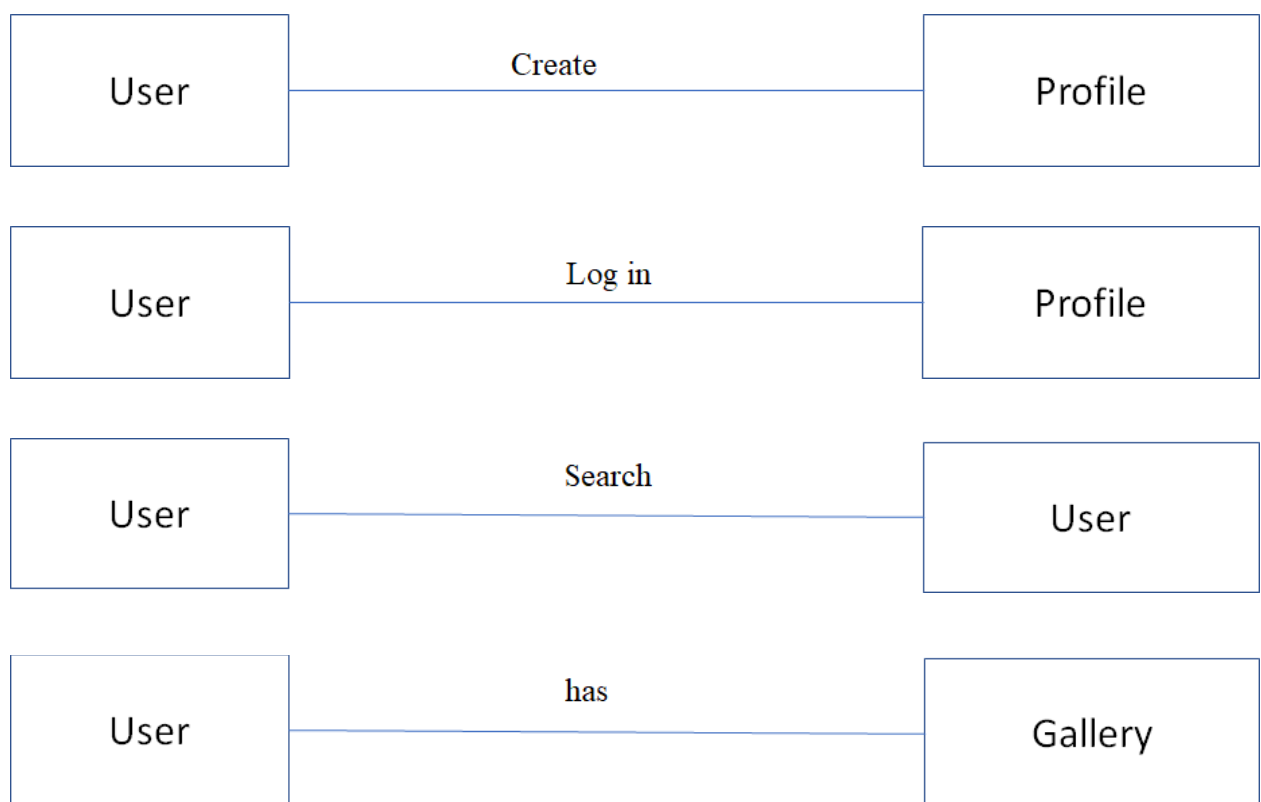
User: User Id, User Password

Profile: Profile Photo, Name, WorkPlace, About, Email, Phone, Profile Type

Group: Group Id, Grp Password, Group type

Gallery: Photo Id, Date, Photo

5.2.5 Relationship among Data Objects



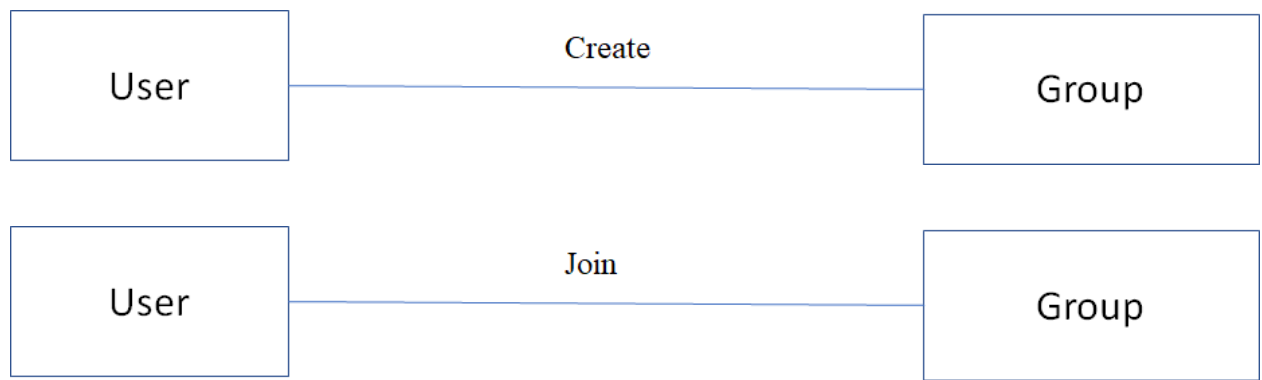


Figure : Relationship among Data Objects

5.3 ER diagram

Based on the data objects found, in order to show the relationship among the data objects, ER diagram that is Entity-Relationship Diagram is used widely. Here, the ER Diagram based on the data objects of Camera Synchronization is shown below.

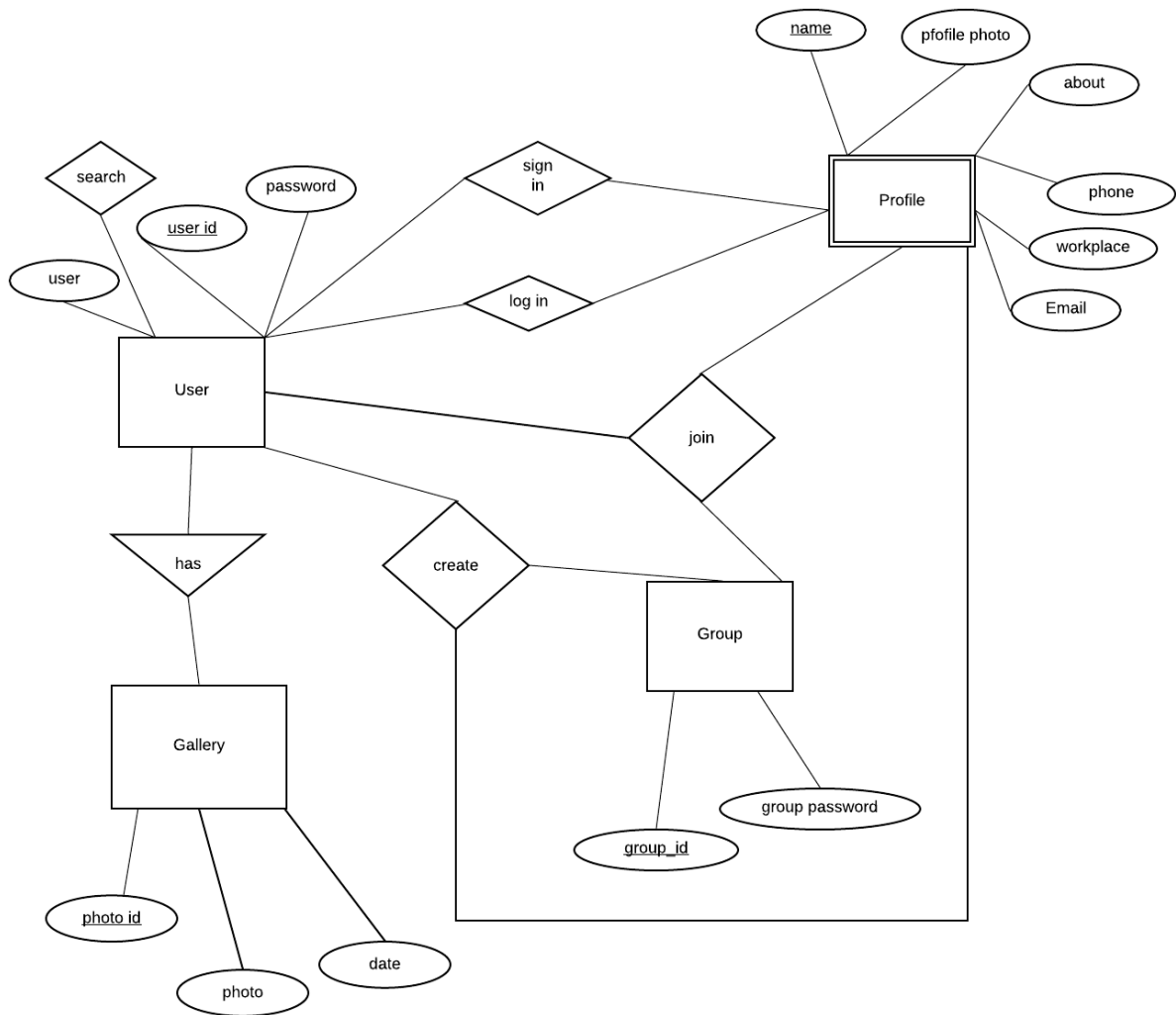


Figure : Entity Relationship Diagram

5.4 Data schema table

Based on the data objects, the following data schema tables can be created:

Table : User

Attribute	Type	Size
User Id	Varchar	20
User Password	Varchar	20

Table : Profile

Attribute	Type	Size
Profile photo	Blob	
Name	Varchar	20
Profile Type	Varchar	5
WorkPlace	Varchar	20
About	Varchar	20
Email	Varchar	10
Phone	Number	10

Table : Group

Attribute	Type	Size
Group Id	Number	10
Group type	Varchar	10
Grp Password	Varchar	20

Table :Gallery

Attribute	Type	Size
Date	datetime	
Photo ID	Varchar	10
Photo	LongBlob	10

Chapter Six

Class Based Model Of Camera Synchronization Android 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.

6.1 Grammatical Parsing and Analysis

To identify our analysis class, 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 scenery. 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 7, 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

These criteria are used to find Potential Classes in following table-

Table : Class Identification with General Classification

No.	Noun or Noun Phrases	P/S	General Classification (GC)
1.	Create Group	P	3
2.	Join Group	P	3
3.	Group Member	S	4
4.	User	P	4,7
5.	Username	S	2
6.	Id	S	2
7.	Password	S	2
8.	Account	S	4,7
9.	System	P	4
10.	Information	S	2
11.	Logged in	S	
12.	Logged out	S	
13.	Connection	P	
14.	Request	P	
15.	View	S	
16.	Online	S	
17.	Profile	P	2

18.	QR Code Sharing	S	
19.	Scan QR Code	S	2
20.	Member	P	4
21.	Non Member	P	4
22.	Photo	S	2
23.	Capture	S	
24.	Synchronization	P	
25.	Offline	S	2
26.	Profile photo	P	2
27.	Settings	P	
28.	Database	P	2
29.	Camera	P	2
30.	Search ID	S	
31.	Search Name	S	
32.	IP Address	P	
33.	Photos	P	2
34.	Switch	P	
35.	Accept	P	4
36.	Decline	P	4
37.	Remove	P	4
38.	Sent	P	

39.	Connection Type	P	3
40.	Email	S	2
41.	Phone	S	2
42	Profile Type	P	2
43	Authentication	P	3

6.1.2 CLASS IDENTIFIED 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

In table , with the help selection criteria we identified whether the noun is accepted for preliminary class or not.

Table : Class Identification with Selection Criteria

No.	Noun	P/S	Special Classification (SC) Accepted	Remarks
1.	Create Group	P		
2.	Join Group	P		
3.	Group Member	S		
4.	User	P	1,2,3,5	
5.	Username	S		
6.	Id	S		
7.	Password	S		
8.	Account	S	2,3,5	
9.	System	P	1,2,6	
10.	Information	S	1	
11.	Logged in	S		
12.	Logged out	S		
13.	Connection	P	2,1,3	
14.	Request	P	1	
15.	View	S		
16.	Online	S		
17.	Profile	P	4,3	
18.	QR Code Sharing	S		

19.	Scan QR Code	S		
20.	Member	P	3,4,2,5	
21.	Non Member	P		
22.	Photo	S		
23.	Capture	S		
24.	Synchronization	P		
25.	Offline	S		
26.	Profile photo	P		
27.	Settings	P	2	
28.	Database	P	1,2,6	
29.	Camera	P		
30.	Search ID	S		
31.	Search Name	S		
32.	IP Address	P		
33.	Photos	P		
34.	Switch	P	2,	
35.	Accept	P		
36.	Decline	P		
37.	Remove	P		
38.	Sent	P		
39.	Connection Type	P		

40.	Email	S		
41.	Phone	S		
42	Profile Type	P		
43	Authentication	P	2,6	

6.2.1 Attributes and Methods of Preliminary Classes

Analyzing the above table, we have categorized the verbs and convert them into method names. We put them to their respective classes and showed them in the table :

Table :Potential Classes after General and Selection Criteria

Potential Class	Nouns	Verb
User	User Id, Profile photo, Password,	send request, accept request, decline request, select settings option, change name, switch mode, update profile photo, view all, view profile, sign in, sign up
System		create account, load account, check id available, verification, show request, delete request, show searched user, save changed in settings
Authentication		provide new user Id, provide name, provide password, match password, provide id and password, provide new group Id,provide new group password
Non-member		Search, sign up, get confirmation, share songs, get guideline ,Search, create group, join group, sent Request, Cancel Request, Remove from group,

Member	member_ID, email_address, password, name, date_of_birth, gender, phone_number	Search,, sign in, sign out, edit account, create group, join group, sent Request, Cancel Request, Remove from group, delete account, get guideline
Database		Update, retrieve, store
Gallery	Photo id, Date, Photo	View, Search

6.2.2 Analysis on potential classes:

User, Member and Non-member have same attributes and verbs in common. So we can make User the superclass of these classes. Again, we can break the Account class into some classes consisted of same type of responsibility. These are Group, Settings, View..

6.2.3 Final Classes

The final classes with their methods are shown below-

Table : Authentication

Attributes	Methods
	provideUserId() provideName() providePassword() passwordMatched()

Table : System

Attributes	Methods
------------	---------

	searchId() searchGroupId() showSearchedId() showSearchedGroupId() createAccount() loadAccount() IdAvailable() Verification() showRequest() sendRequest() CreatGroup() deleteRequest() removeUserId() saveSettings()
--	--

Table : User

Attributes	Methods
User Id	Option() CreateGroup() AcceptRequest() seeRequest() CancelRequest() RemoveUser()

Table : Member

Attributes	Methods
name Profile photo Password Phone Email	Option() SearchMemberId() SearchGroupId() Log In() Log Out() settingsOption() changeName() switchMode() updateProfilePhoto() updateProfileInfo()

Table : Gallery

Attributes	Methods
Photo Date Photo Id	StorePhoto() DeletePhoto() SearchPhoto()

Table 15: View

Attributes	Methods
	viewOption() viewAll() viewProfile()

6.3 Class Responsibility Collaborator modeling

CRC modeling stands for Class Responsibility Collaboration modeling. CRC modeling includes class cards and CRC diagram.

6.3.1 Class cards

Class cards show attributes, methods and collaborating class names along with their responsibility. A responsibility comprises one or more methods together. The potential classes for the system are- Authentication, System, Account, Circle, Settings, Check In, Checked In Position, Home Position, Work Position, Temporary Position. The proposed class cards of these classes are shown below.

Table : Authentication Class Card

Responsibility	Collaborators
Receiving information for registration	System
Checking availability of user Id	System
Receiving id and password	
Verifying login	System

Table : System Class Card

Responsibility	Collaborators
Creating Account	User
Loading Account	User
Search User Id	User

Table 22: Gallery

Responsibility	Collaborators
Show	System

Table 23: User

Responsibility	Collaborators
Create Group	System
Join Group	System
RemovePhoto	Gallery
StorePhoto	Gallery

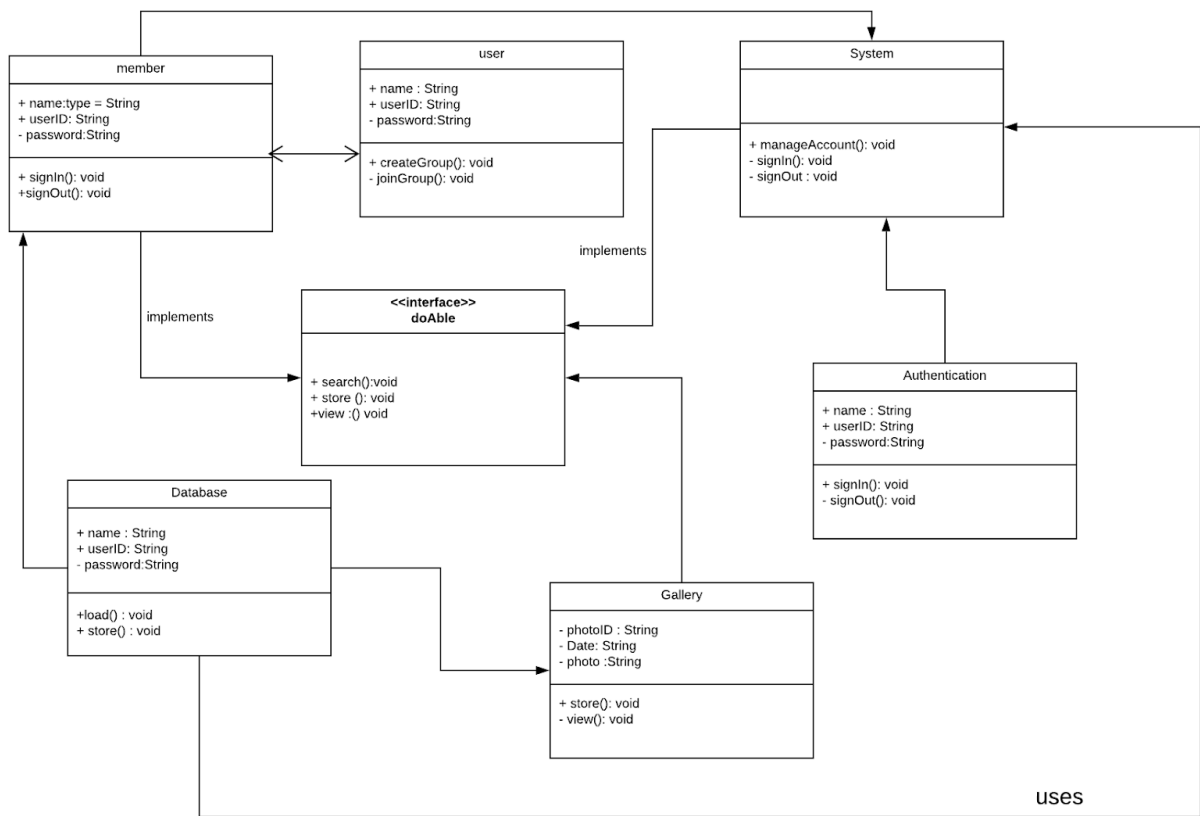
Table : Member

Responsibility	Collaborators
Update Profile Photo	
Update Profile Info	

Table : View Class Card

Responsibility	Collaborators
View Profile	Member
View Photo	Gallery

6.3.2 UML Diagram

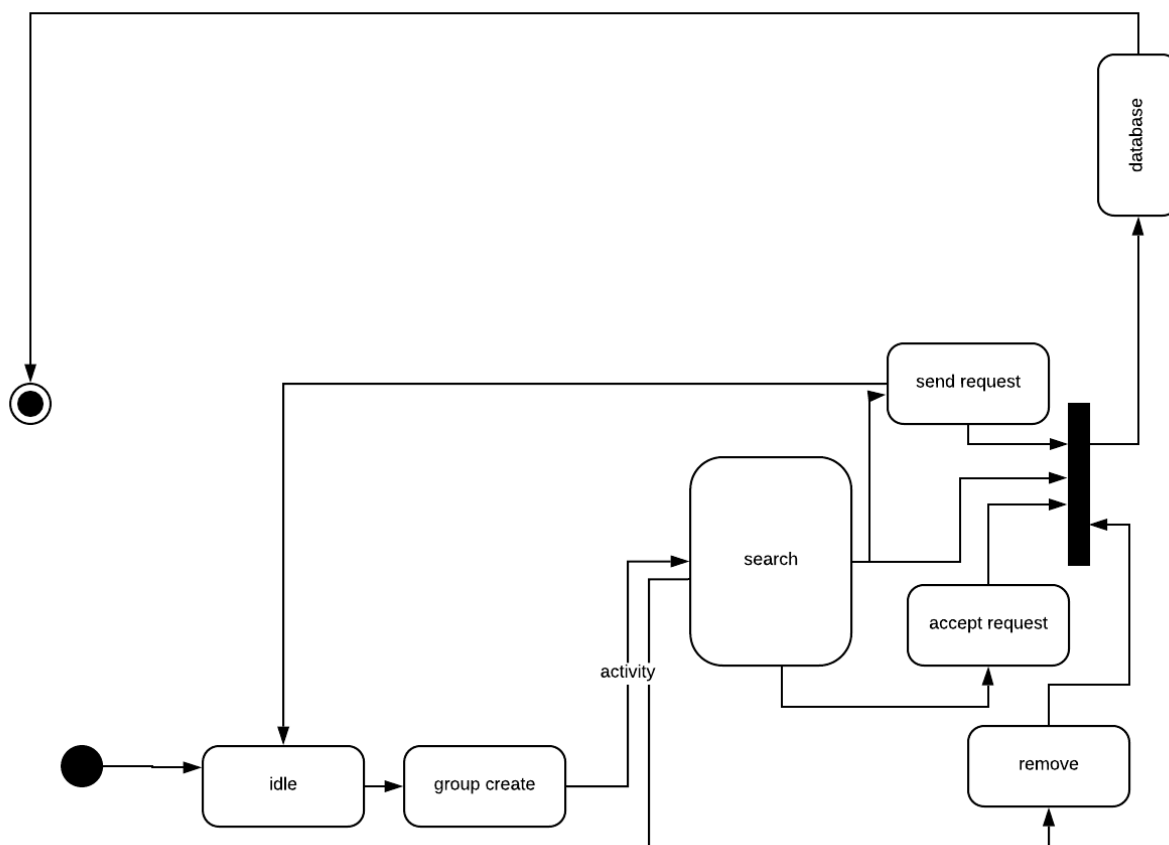


CHAPTER 7

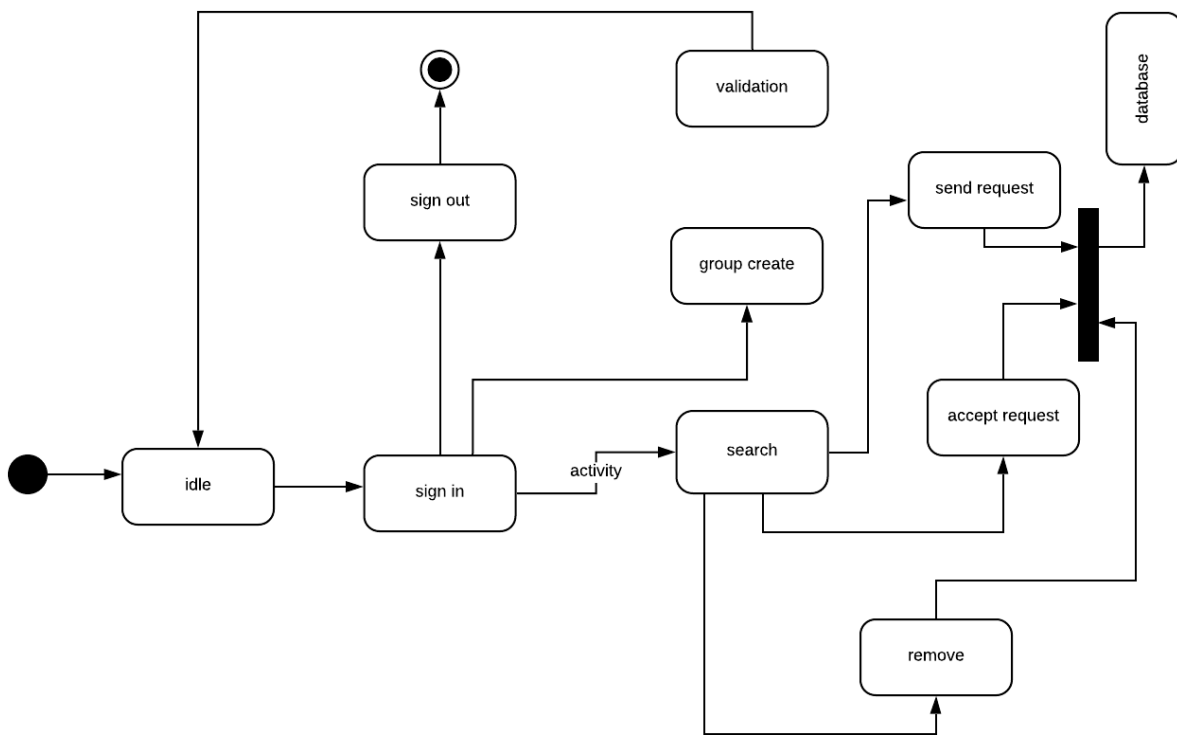
BEHAVIORAL MODEL OF 'CAMERA SYNCHRONIZATION' A MOBILE APPLICATION

The behavioral model indicates how software will respond to external events. There are two different behavioral representations. 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. State diagram shows the state in a module of an user. State diagram of this project are following:

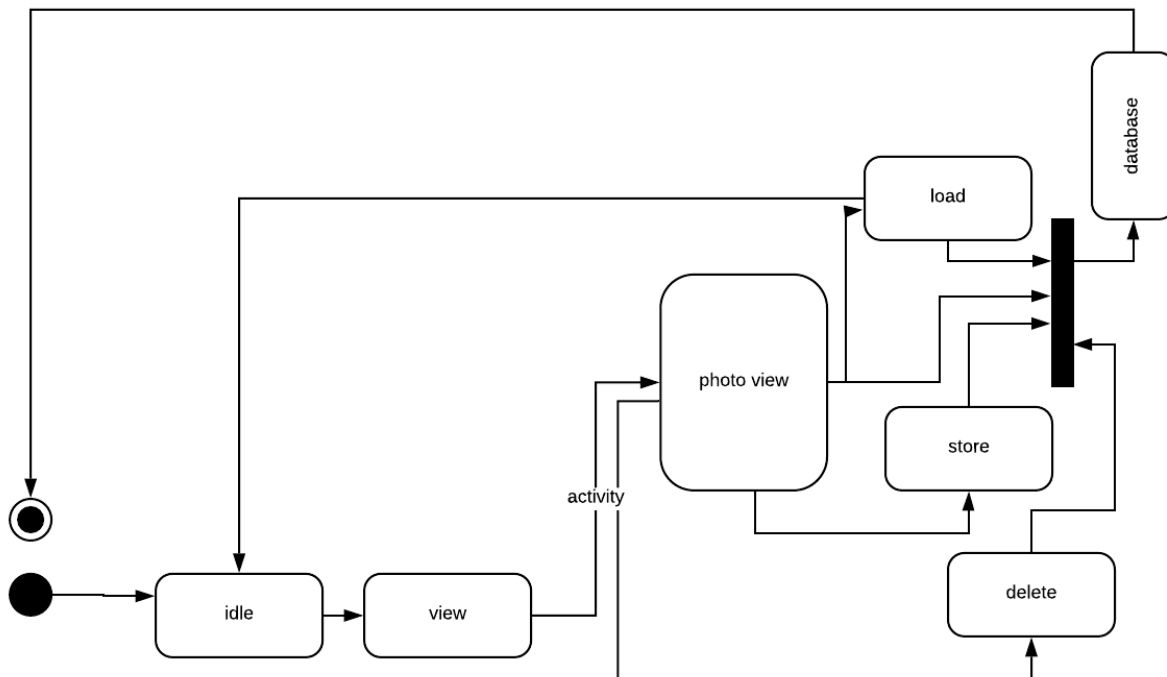
7.1 State diagram of User :



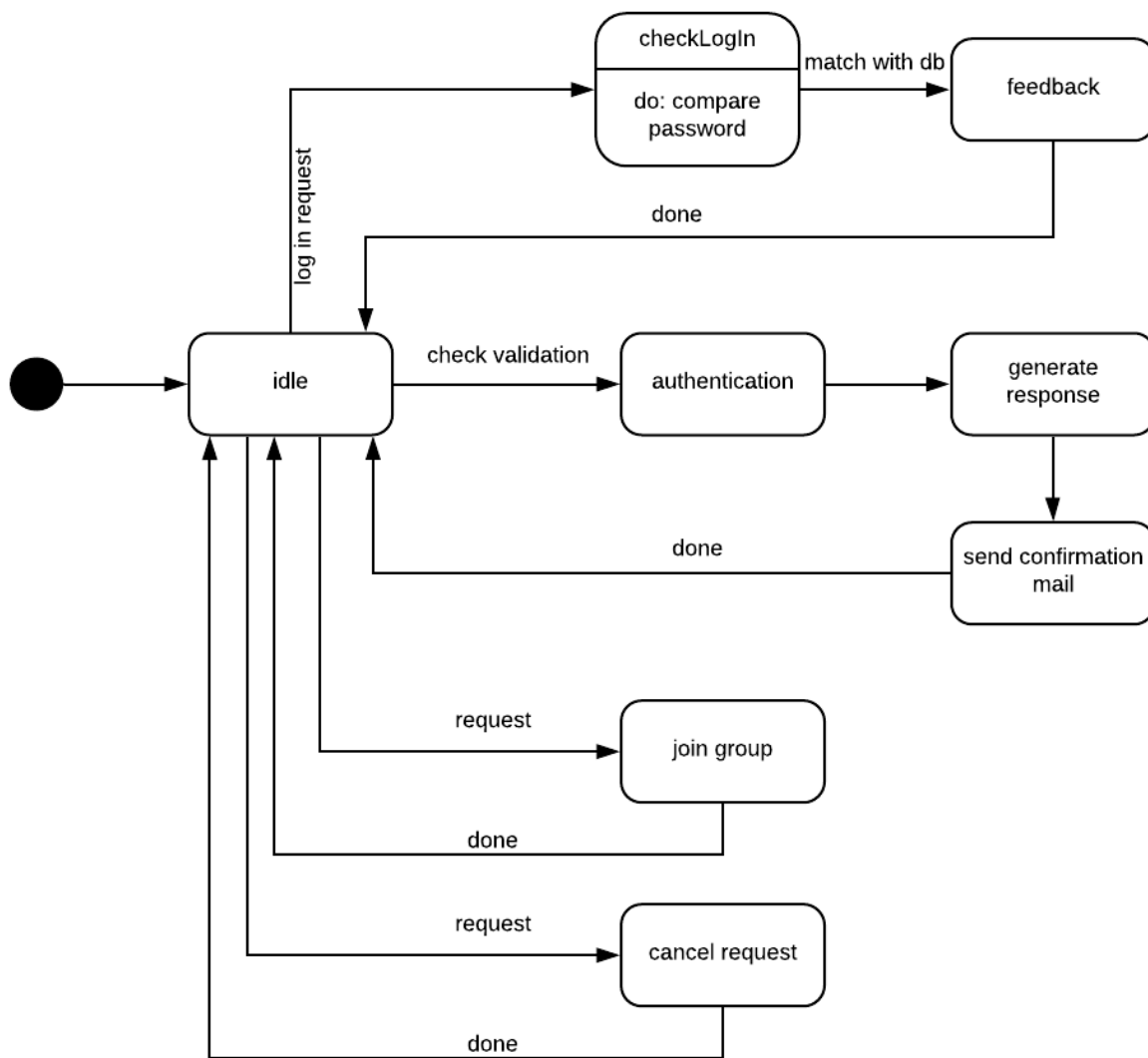
7.2 State diagram of Member:



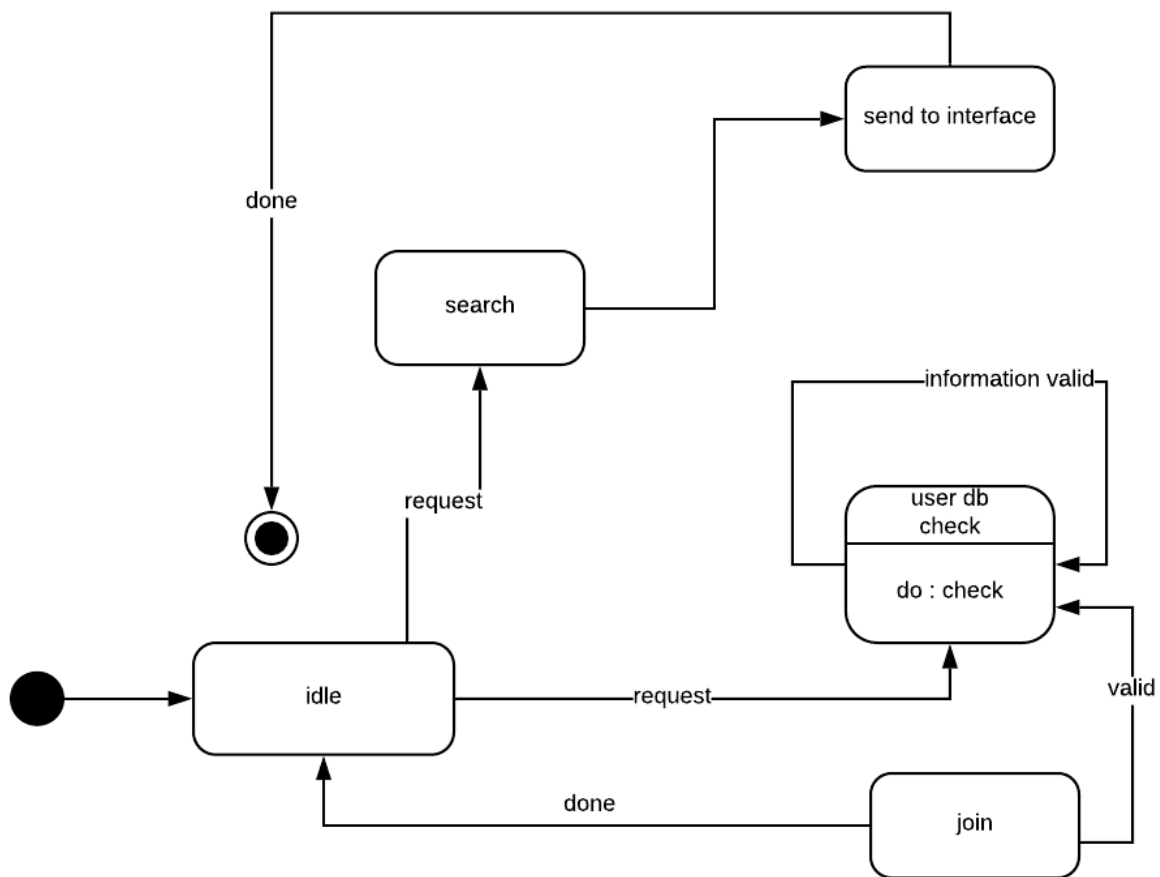
7.3 State diagram of Gallery:



7.4 State diagram of System:



7.5 State diagram of Database:



CHAPTER 8

DATA FLOW MODEL OF 'CAMERA SYNCHRONIZATION' A MOBILE APPLICATION

A data flow model is a 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 “Camera synchronization” is given below:

8.1.1 level 1:

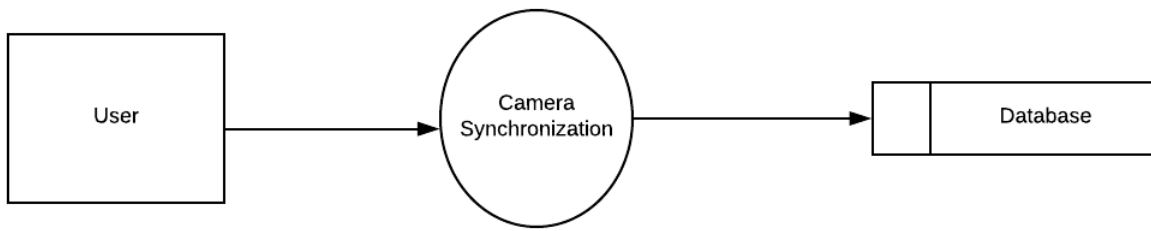
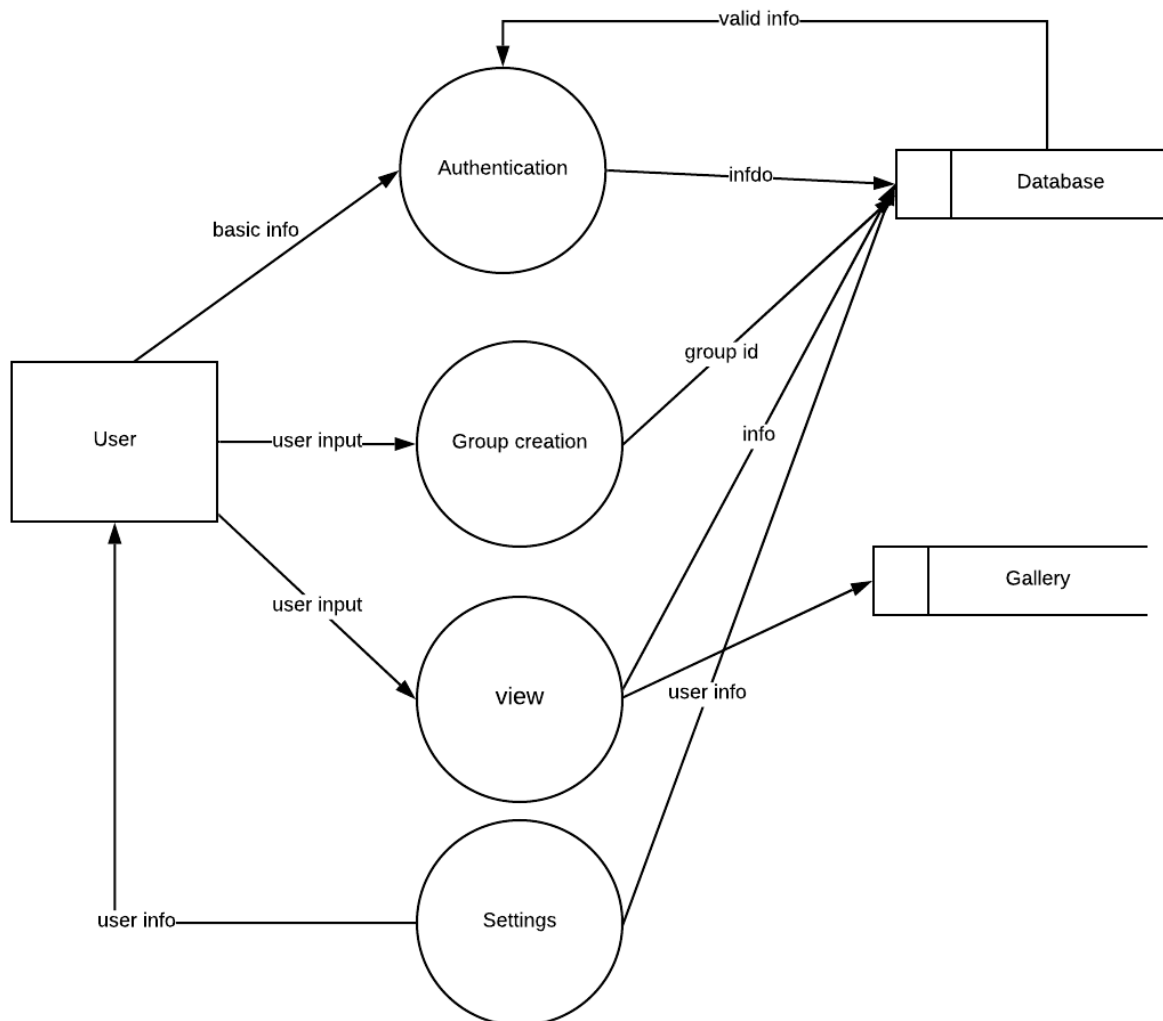


Figure 1: level 1 for Camera Synchronization

8.1.2: Level 2.1



level 2.1 for Camera Synchronization

Figure 2:

8.2.1 Level 2.2.1

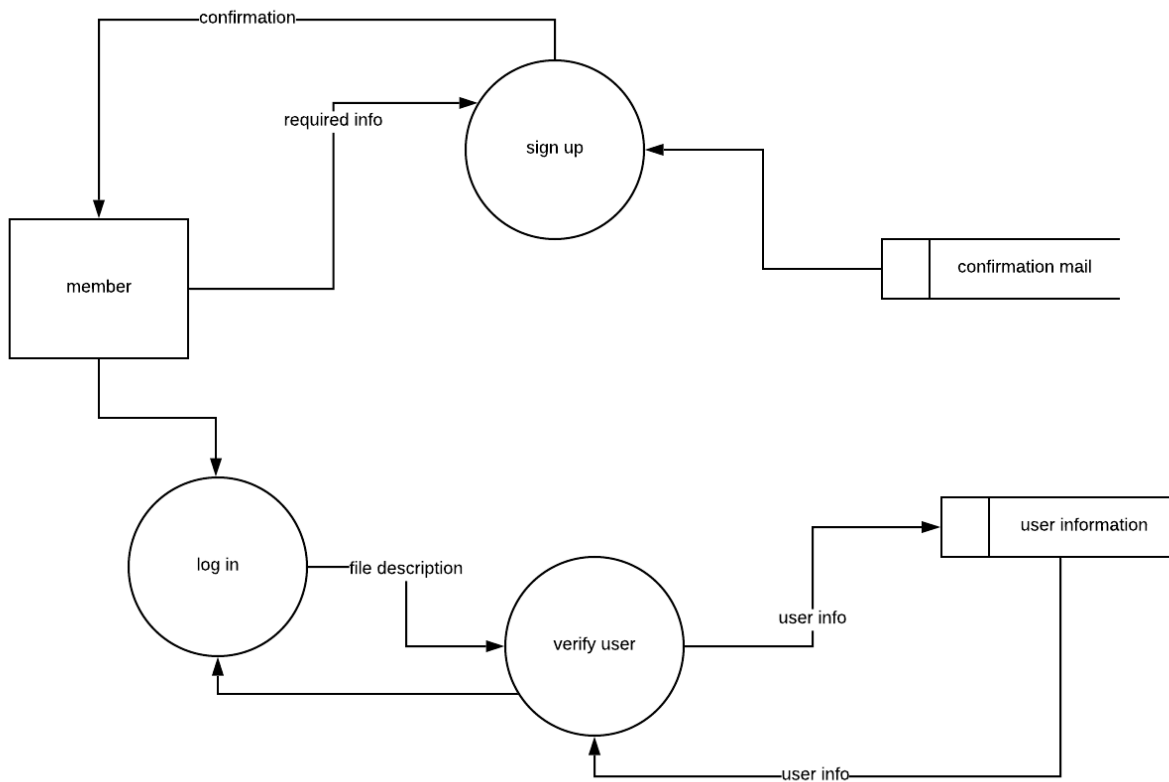


Figure 3: level 2.2.1 for Camera Synchronization

8.2.1 Level 2.2.2

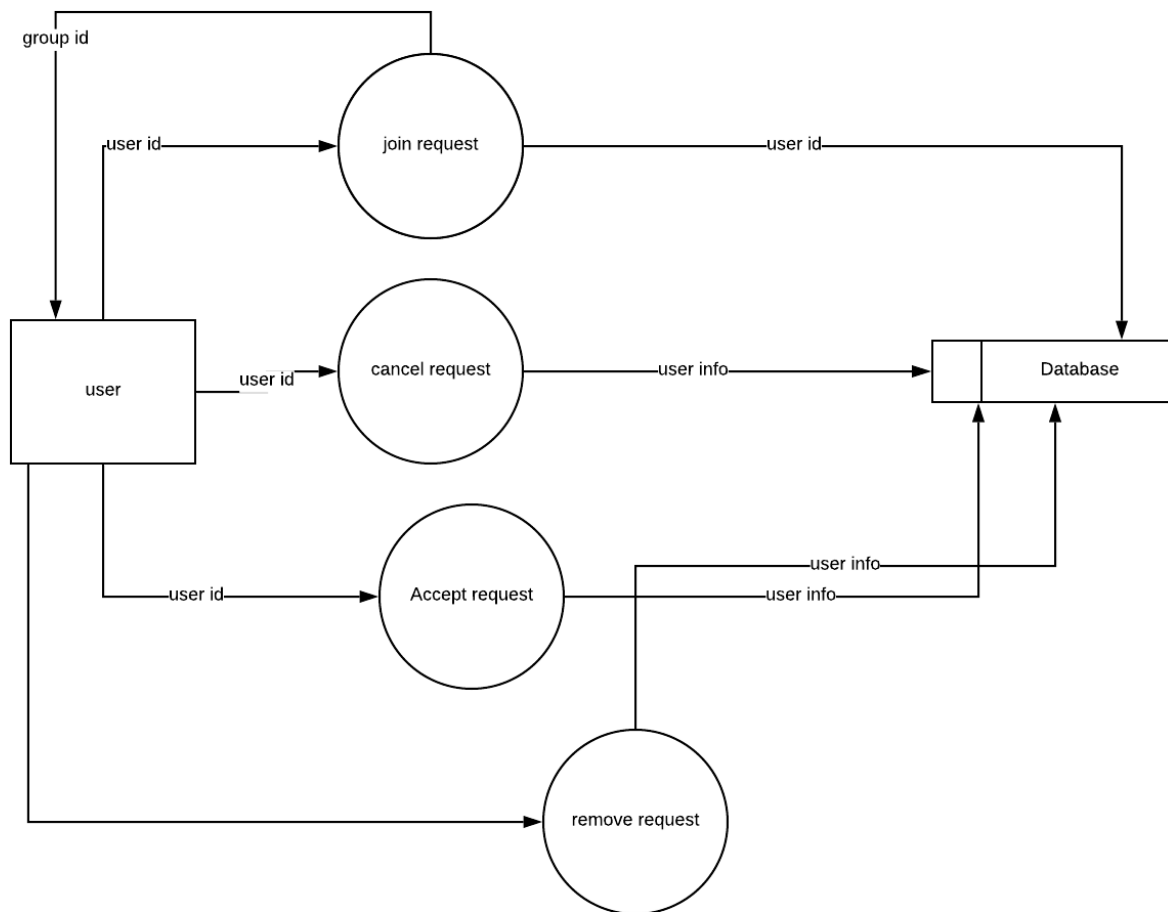


Figure 4: level 2.2.2. for Camera Synchronization

8.2.1 Level 2.3

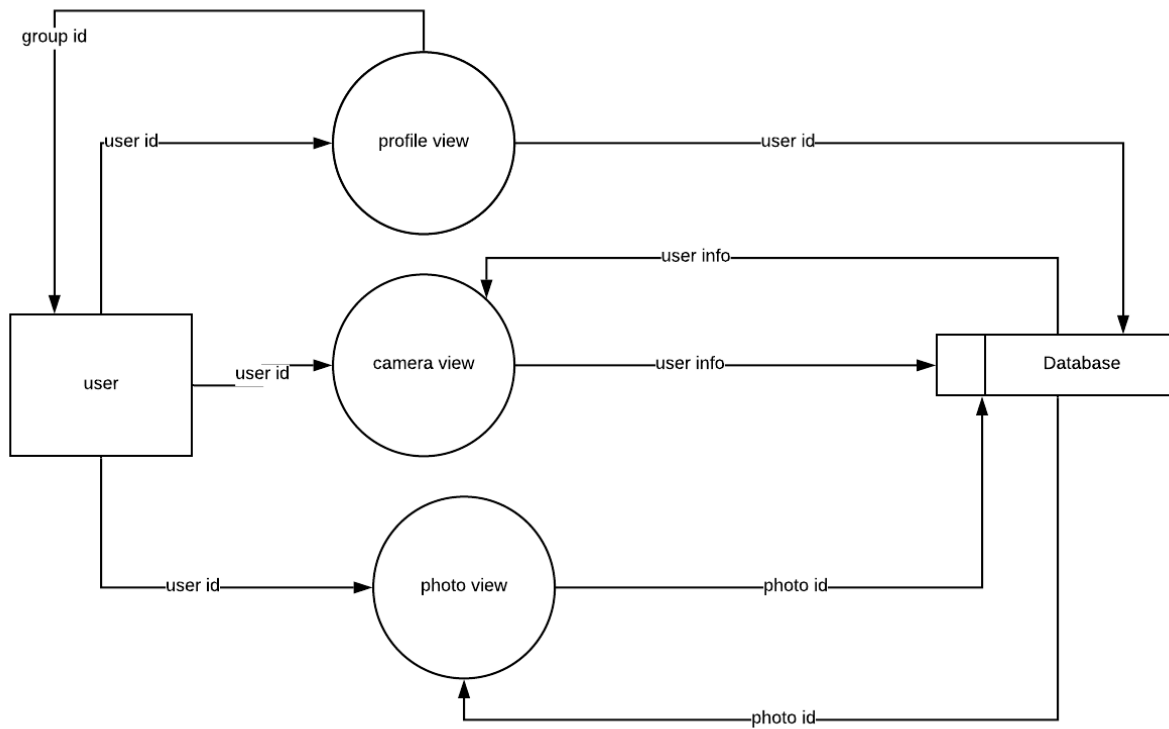


Figure 5: level 2.3 for Camera Synchronization

8.2.1 Level 2.4

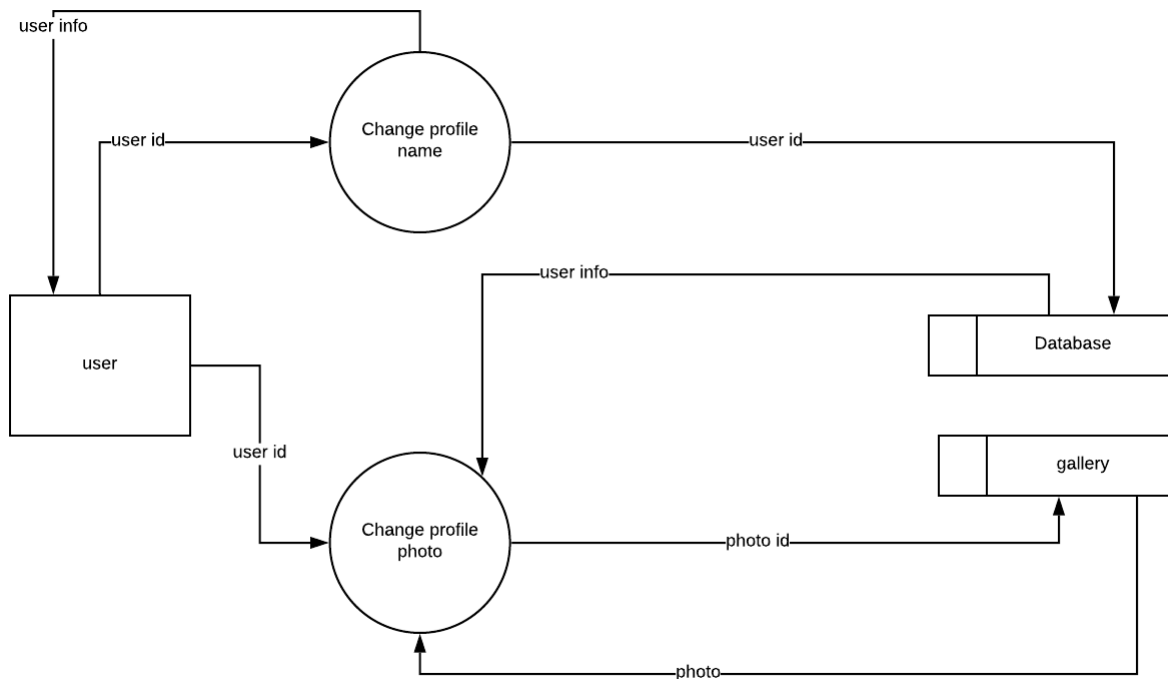
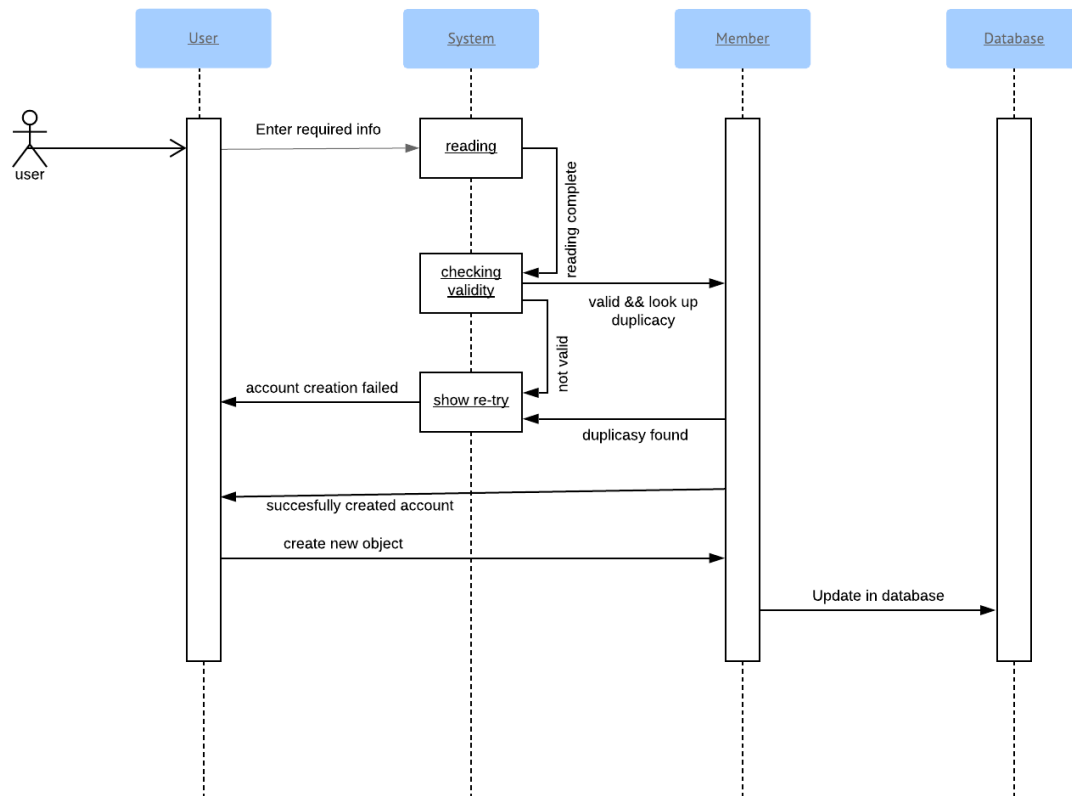


Figure 6: level 2.5 for Camera Synchronization

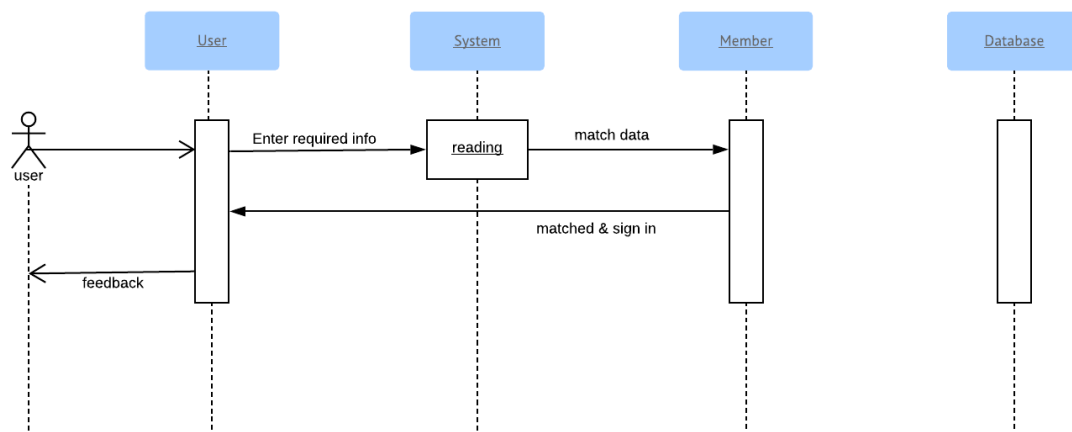
8.2 Sequence diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. We have shown the sequence diagram of three modules- Homepage, personal account and administration.

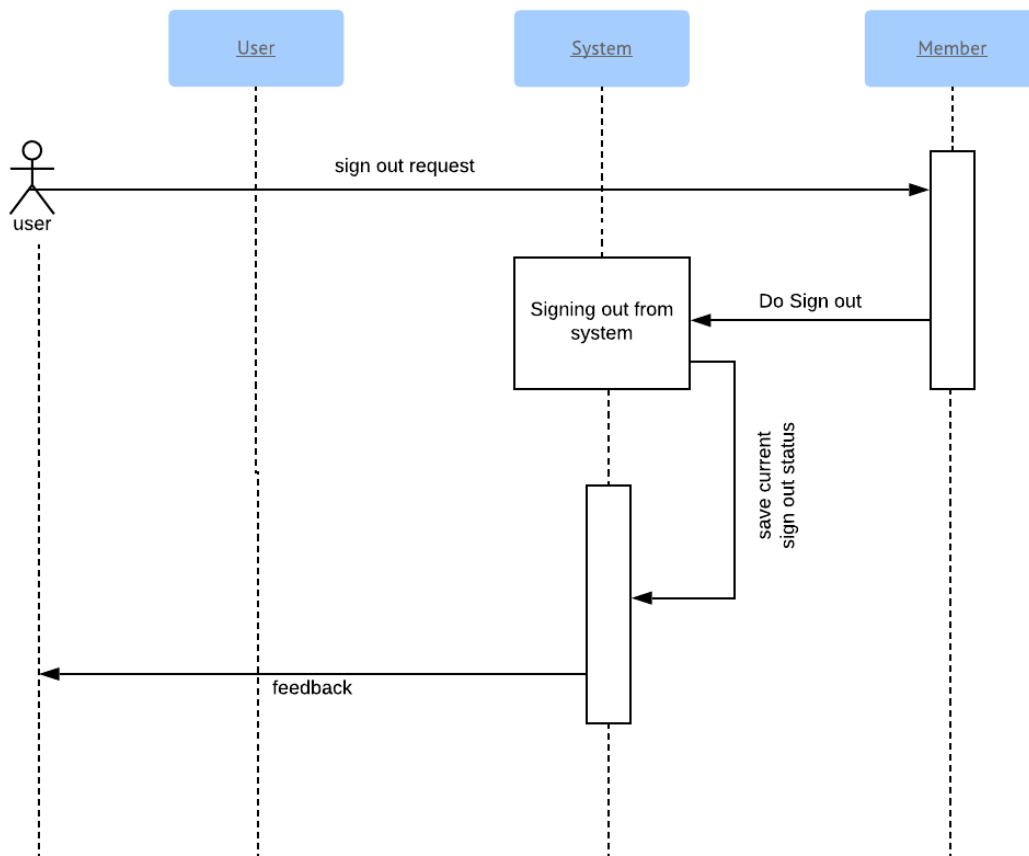
8.2.1 Sequence diagram of Sign Up



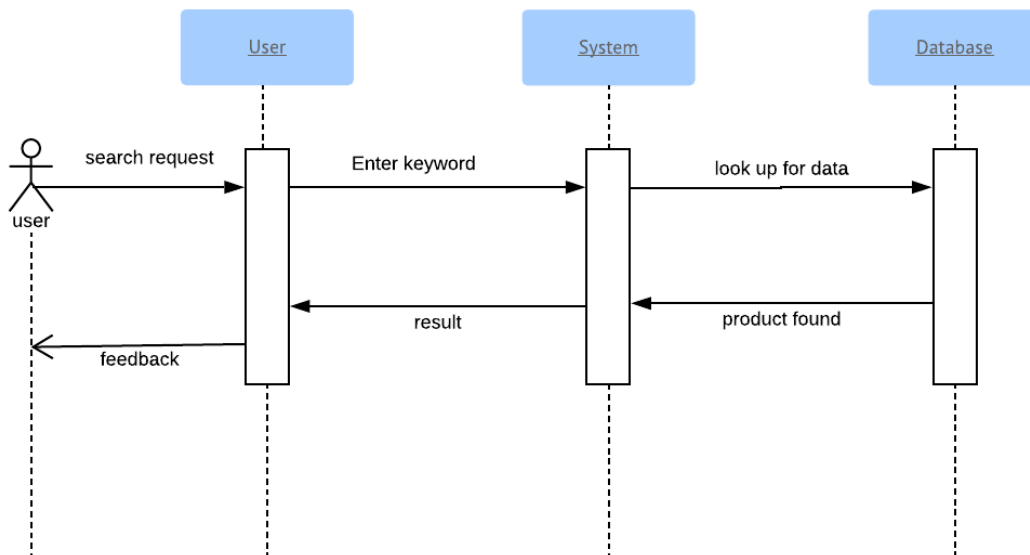
8.2.2 Sequence diagram of Sign In



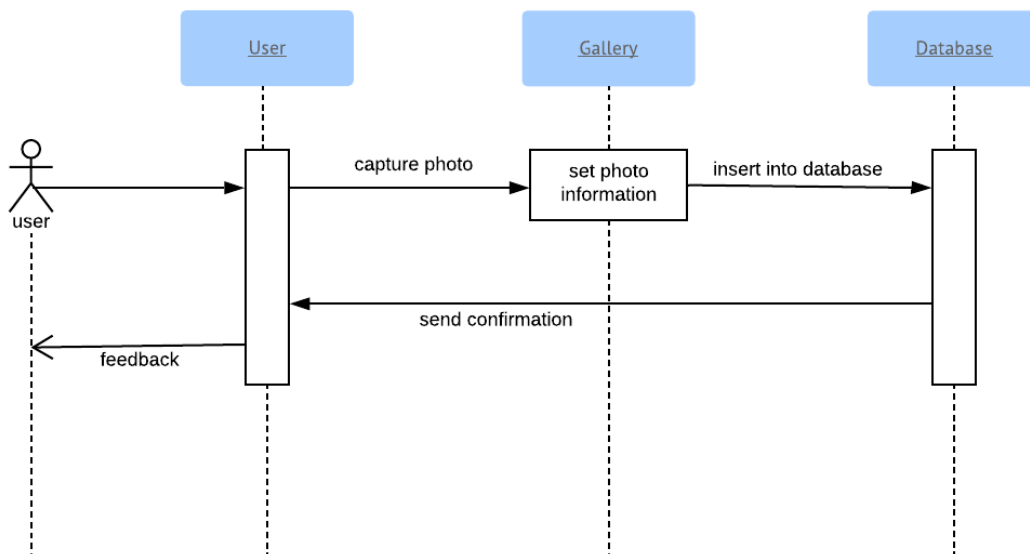
8.2.3 Sequence diagram of Sign Out



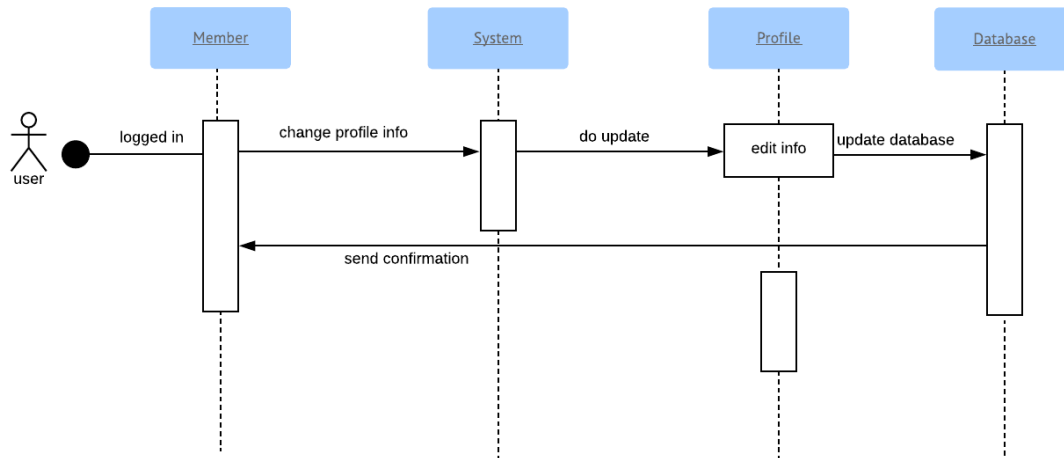
8.2.4 Sequence diagram of Search



8.2.5 Sequence diagram of Save photo



8.2.6 Sequence diagram of Update Profile



Chapter 9: Conclusion

We are very pleased to successfully draw the conclusion of software requirement specification and analysis of our software project - Camera Synchronization an Android App.

We hope our report will be able to convey a clear picture of our system to all stakeholders, and act as a basis throughout full development cycle. We have tried our best to make a complete, practical, consistent and unambiguous specifications; which helped us tremendously in our understanding of the scope and detailed process of software requirement engineering process. We think that this report has been written in an easy-to-read way as well as with full information required to have a good concept over the idea. We sincerely hope this document will be able to satisfy the goals all stakeholders expect from it. We hope that any reader going through this document can easily understand the whole idea behind the Camera Synchronization Android App. Hopefully, it will be an easy path-showing document for the implementation of the application!

