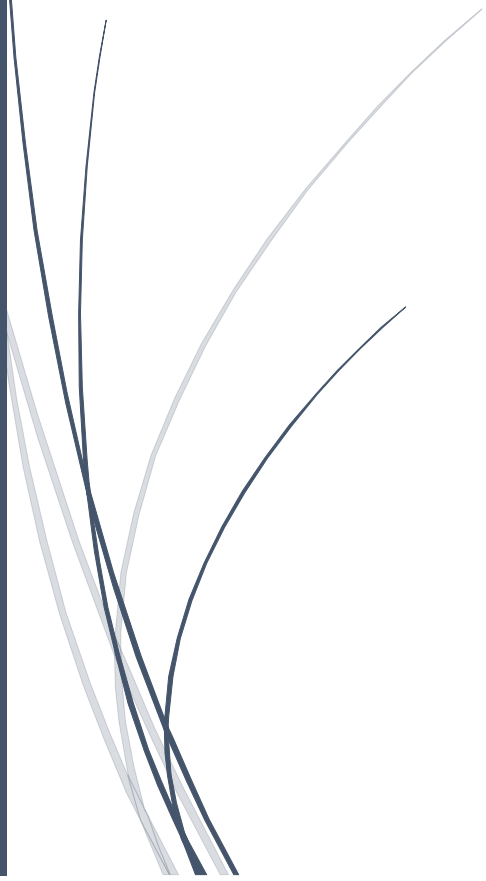
3/20/2019

Smart Tourist Guide



Software Project Lab -2
Smart Tourist Guide

Submitted to

SPL-Coordinators
Institute of Information Technology
University of Dhaka

Submitted By

Name	Roll
Muhabbat Sarker Eshan	BSSE0939
Md Jahirul Hoque Rifat	BSSE0940

Supervised by

Kishan Kumar Ganguly
Lecturer
Institute of Information Technology
University of Dhaka

LETTER OF TRANSMITTAL

20th March 2019
SPL-2 Coordinator,
Institute of Information Technology
University of Dhaka.
Subject: Submission of SPL-2 SRS term report on Smart Tourist Guide.
Sir,
With due respect, we are submitting the report on the above topic you assigned to us. In this report, we have given our best effort albeit some shortcomings. We earnestly hope that you would excuse our errors and oblige thereby.

Yours sincerely
Muhabbat Sarker Eshan –BSSE0939
Md Jahirul Hoque Rifat – BSSE0940

3nd Year, 5th Semester, 9th Batch
Institute of Information Technology
University of Dhaka
Session: 2015-16

Table of Contents

Table of Figures

List of Tables:

Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the SPL-2 Software Requirements Specification and Analysis (SRS) report on Smart Tourist Guide. We would like to thank our supervisor, Kishan Kumar Ganguly, Lecturer, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. In completing this paper, we have collected various important data and information from various students and tourists. We are thankful to all who helped us to prepare this report.

Purpose

This document initially describes the Software Requirement of Smart Tourist Guide. It contains functional, non-functional and supporting requirements and establishes a requirements baseline for the development of the system.

Chapter 1: Introduction

1.1 Purpose

This document is about the Software Requirements Specification (SRS) for Smart Tourist Guide. It contains detailed functional, non-functional and establishes a requirements baseline for development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as the official means of communicating user requirements to the developer and provides a common reference point for both the developer team and stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

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

- The project managers of the developer team will use this SRS to plan milestones and a delivery date and ensure that the developing team is on track during development of the system.

- The designers will use this SRS as a basis for creating the system's design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer's needs.

- The developers will use this SRS as a basis for developing the system's functionality.

1.3 Conclusion

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

Chapter 2: Inception of STG

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

2.1 Introduction

Inception is the initial phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved. 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 establish a basic understanding of the problem, the people who want a solution, the nature of the solution that is desired and the effectiveness of preliminary communication and collaborations between the other stakeholders and the software team.

To establish the groundwork, we have worked with the following factors related to the inception phases:

- List of stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Requirements questionnaire

2.1.1 List of Stakeholders

Stakeholder refers to any external entity or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that may be affected by its installation. At inception, a list of people who will contribute input as requirements are elicited. The initial list will grow as stakeholders are contacted because every stakeholder will be asked: “Whom else do you think I should talk to?”

To identify the stakeholders, we consulted with a member of Novo IT and asked him following questions:

● Who is paying for the project?
● Who will be using the project outcomes?
● Who gets to make the decisions about the project (if this is different from the money source)?
● Who has resources I need to get the project done?
● Whose work will my project affect? (During the project and also once the project is completed).

- What kind of places do you prefer to recommend?
- What kind of search do you prefer?
- Do you want to route and rate places ?

Stakeholders

Stakeholders are those people who get affected by the overall System directly or indirectly. In this project we initially find out the stakeholders of this project. Stakeholders are:
1. User(people and clients)
2. Admin

2.1.2 Multiple Viewpoints

Owner's Viewpoints:

• Should have a feature to change profile data.
• Capability of show route to destination from source.
• Should contain Facebook ID link for communication.
• Capability of category searching.
• Should contain features to recommend places.
• Account recovery
• Capability of storing trip history and details of previous visited places.

Client's Viewpoints:

• Capability of storing trip history and details of previous visited places.
• Must contain features to provide feedback and review.
• Must contain the feature of provide communication with other users.
• Should track down current location.

• Should have a feature of good recommendation of places.
• Notification after completion of any visiting places previously recommended.

2.1.3 Working Towards Collaboration

Common Viewpoints:

• Capability of storing history and details for every single visited places.
• Account recovery
• Notification after recommendation to user.

Conflicting Viewpoints:

• May not communicate with other users in the app directly.

Final Requirements:

• Capability of show route to destination from source
• Capability of tracking places.
• Should have a feature of good recommendation of places.
• Notification after completion of any place visited
• Capability of category basis searching.
• May have features to edit profile data.
• Account recovery.
• Capability of storing history and details for previous visited places.
• Must contain features to provide feedback.
•Should provide Facebook id link so that communicate with other user.

2.1.4 Requirements Questionnaire

We asked the stakeholders some context free questions to understand the project's overall performances and the goals of the project. Those context free questions have been added to section 2.1.1. These questions have helped us to identify the stakeholders. Then we asked our next question.

2.2 Conclusion

The Inception phase helped us to establish basic understanding about the Industrial Management System, identify the stakeholders who will be benefited if this system becomes automated, define the nature of the system and the tasks done by the system, and establish a preliminary communication with our stakeholders.

Chapter 3: Elicitation

This chapter specifies the Elicitation phase.

3.1 Introduction

Requirements Elicitation is a part of requirements engineering that is the practice of gathering
requirements from the users, customers and other stakeholders. Many difficulties were faced, like understanding the problems, making questions for the stakeholders, limited communication with the stakeholders due to a short amount of time and volatility. Though it is not easy to gather requirements within a very short time, these problems have been surpassed in an organized and systematic manner.

3.2 Eliciting Requirements

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

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

3.3 Collaborative Requirements Gathering

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.

- The meetings were conducted with a member of Novo IT, they were questioned about their requirements and expectations from the Smart Tourist Guide.
- The secretary was asked about the problems he is facing with the existing tourist applications.
- At last we selected our final requirements from the meetings.

3.4 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the customer into technical requirement for software. QFD's main aim is understanding that what is valuable to the customer and then deploys these values throughout the engineering process. There are mainly three types of requirements as QFD:

Normal Requirements:

Normal requirements include the objectives and
goals that are stated during meeting with customer for a product or system.We found some such objectives and goals during requirement analysis in inception step:

- Suggestion on route according to source place, destination place.
- Notification when system recommend user some places based on previous history.
- Account recovery of users.
- Capability of storing history of previous visited places and details for every previous visited places.

Expected Requirement:

These requirements may not have described by
the users but they are so fundamental that the absence of them will cause significant dissatisfaction.

- Capability of location track.
- Allow only valid users to get the app's services.
- Must have features to rate the places.
- Have features to review and rating the visited places.
- Allow only valid users to login

Exciting Requirement:

These requirements are beyond customer's expectation but proved to be very exciting.

- Contains features to provide feedback

Usage scenario

Authentication:

Sign Up:

In the “Smart Tourist Guide”, there is an authentication part, where it allows the user to access the system. Admin and people are the users of this system. User can create accounts in two different ways i.e. Facebook and Email. If user want to create an account via Facebook, he/she must provide Facebook id’s email/phone and Facebook password. If user want to sign up using email, he must provide valid email address and email password. Firebase backend services will verify these credentials in background and return a response.

In sign up process user also have to give first name, last name, email id, address, phone number, and account type (types control the access to the system). No name can contain any number, punctuation mark or special character and the length of the name should be between 2 characters and 20 characters. Password should contain at least a letter, a number and length need to be minimum 8 characters. These later part of sign up process, will be user profile data. There will be duplicity and validity (syntax) checking for email and password. If all the information is correct, user will sign up into account.

Sign in:

If any user has an account, he/she can sign in to the system. User will login in by two different ways i.e. using Facebook and Email address. If user have valid Facebook account, he must provide Facebook id, email/phone and Facebook password for login. If user want to login using Email, he must provide valid email address and password. If the password is wrong, there is a retry option. If the retry count is 3, the user will be blocked for 2 minutes. After block time has finished user can attempt to sign in to the system again.

Forget Password:

If user forget his/her password, a auto generated code will be sent to their phone number. By entering the valid code user will reset their password. If the code is wrong, there is a retry option. If the retry count is 3, the user will be blocked for 2 minutes. After block time has finished user can attempt to change password in to the system again.

Profile

Edit profile:

Any user can change their information. To change information, he/she has to sign in then change information. He/she has to confirm the changes and the changes will be confirmed.

Sign Out:

User has an option to log out from the system. User will automatically be logged out of his/her account if the account remains idle for more than 7 days.

Searching

Category Search:

User can select the category of browsing type wise. As the system has different type of places (Park, Zoo, Museum, Restaurants, Waterfall, Sea-beach) user can select place type and view place type.

Manual Searching:

If user want to search according to his/her place of interest, he/she can search manually. Actually category basis searching do search in the given area. So, if user want to search outside of the area this part will work fine.

Route:

When user select the places, system will provide the best route to destination from the source. System will track their location to ensure that routing will start. For this turning on GPS and an active internet connection is mandatory.

Recommendation

Category:

User can select the category of browsing type wise. As the system has different type of places (Park, Zoo, Museum, Restaurants, Waterfall, Sea-beach) user can select place type and view place type wise. System will provide some recommendation for user when they select particular category.

Route:

When user select the places, system will provide the best route to destination from

the source. System will track their location to ensure that routing will start. For this turning on GPS and an active internet connection is mandatory.

Feedback

Rating:

User can rate a particular place after visiting one place according to his own satisfaction level. Only registered user can rate the places.

Comment:

User can write a comment after visiting this particular places based on surrounding (beauty, environment etc). To give comment of a place, he has to be a registered user. So sign in is required.

Review:

User can share of his/her experience with other users by writing review of previous visited places. For this, he/she need to be a authenticate user.

Chapter 4: Scenario Based Modeling

4.1 Introduction

In this model the system is described from the user's point of view. As this is the first model, it serves as input for creation of other modeling elements.

4.2 Use Case Scenario

Level 0	Level 1	Level 2
Smart Tourist Guide	Authentication	Sign up
		Log in
		Forget password
	Profile	Edit profile
		Logout
	Searching	Category
		Manual Search
		Route
	Recommendation	Category
		Route
	Feedback	Rating
		Comment
	Review	Post
		Edit review
	Notification	Recommendation
		Confirmation
	Settings	User manual
		Helpline

Table-1 Use case scenario

4.3 Use Case Descriptions

4.3.1 Authentication

4.3.1.1 Sign up

Use Case: Sign up
Primary Actor: Admin, User

Goal in context: to create an account

Preconditions:
1. System has been designed for Sign up
2. System has interface for Sign up

Triggers: Admin and user has to sign up
Scenario:
1. Visit page and sign up
2. Provided the required information
3. Check availability of email and password and Facebook
4. Verify information
Exception:
· Same Facebook Id
· Same email
· Verification failed
Priority: Essential, must be implemented.
When Available: First increment

4.3.1.2 Sign In

Use Case: Sign In
Primary Actor: User, Admin

Goal in context: To log in an account

Preconditions:

1**.** System has been designed for sign in

2. System has interface for sign in
Triggers: Admin and user has to Sign in
Scenario:

1. Visit page and sign in

2. Provide require Information

3. Verify information

Exception:
· Same facebook Id
· Verification Failed
· Same Email
Priority: Essential, must be implemented.
When Available: First increment

4.3.1.3 Forget Password

Use Case: Forget password
Primary Actor: User, admin

Goal in context: To recover account password

Preconditions:

1. System has been designed for Forget password

2. System has interface for Forget Password

Triggers: Admin and user have to recover password
Scenario:

1. Visit app and forget password

2. Verify Information

Exception:
· Password not in range
· Username ambiguous

Priority: Essential, must be implemented.
When Available: First increment

4.3.2 Profile

4.3.2.1 Edit Pofile

Use Case: Edit profile
Primary Actor: User, Admin

Goal in context: to manage an account

Preconditions:

1. System has been designed to change profile

2. System have to interface to change profile

Triggers: User has to change profile
Scenario:

1. Visit the application and login
2. Provide required information for login

3. Provide information that need to be change
4. Send notification for successful change

Exception:

Information is same as the previous one
Priority: Essential, must be implemented.
When Available: First increment

4.3.2.2 Log Out

Use Case: Sign Out
Primary Actor: User, Admin

Goal in context: sign out from site

Preconditions:

1. System has been designed for sign out
2. System has interface for sign out
3. User must be logged in

Triggers: Admin and user has to sign out
Scenario:

1. Visit page
2. Click Sign out button

Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.3.1 Category

Use Case: Category
Primary Actor: User

Goal in context: Show the category of places

Preconditions:

1. System has been designed for category
2. System has interface for category
3. User must be logged in

4. check availability of places
Triggers: User can select a category
Scenario:
1. Visit page
2. Click category button
Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.3.2 Manual Search

Use Case: Manual Search
Primary Actor: User

Goal in context: Show the result of manual search

Preconditions:

1. System has been designed for manually search
2. System has interface for manual search
3. User must be logged in
4. Check availability of places
Triggers: User can search according to his point of interest
Scenario:
1. Visit page
2. Should be logged in
3. Can search manually
Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.3.3 Route

Use Case: Route
Primary Actor: User, Admin

Goal in context: Show the route of places

Preconditions:

1. System has been designed for route
2. System has interface for route
3. User must be logged in
4. Check availability of places

Triggers: User can select a category
Scenario:
1. Visit page
2. Show route

Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.4.1 Category

Use Case: Category
Primary Actor: User

Goal in context: Show recommendation on nearby category places

Preconditions:

1. System has been designed for recommendation
2. System has interface for recommendation
3. User must be logged in
4. Check availability of places

Triggers: User can select a category
Scenario:
1. Visit page
2. Provide user information for recommendation

Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.4.2 Route

Use Case: Route
Primary Actor: User

Goal in context: Show recommend places route to destination

Preconditions:

1. System has been designed for recommendation
2. System has interface for recommendation
3. User must be logged in
4. check availability of places

Triggers: User can select route button
Scenario:
1. Visit page
2. Provide user information for recommendation

Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.5.1 Feedback

Rating

Use Case: Rating
Primary Actor: User

Goal in context: Can rate the places

Preconditions:

1. System has been designed for rating
2. System has interface for give rating
3. User must be logged in
4. check availability of places

Triggers: User can select rating
Scenario:
1. Visit page
2. Provide user information for rating

Exception:
· User not logged in

Priority: Essential, must be implemented.
When Available: First increment

4.3.5.2 Feedback

Comment

Use Case: Comment
Primary Actor: User

Goal in context: Can write comments

Preconditions:

1. System has been designed for feedback
2. System has interface for write comment
3. User must be logged in
4. Check availability of places

Triggers: User can have comment option
Scenario:
1. Visit page
2. Provide user information for rating

Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.5.1 Review

Post

Use Case: Post
Primary Actor: User

Goal in context: Can post about the places

Preconditions:

1. System has been designed for Review
2. System has interface for give review
3. User must be logged in

4. check availability of places
Triggers: User can write review
Scenario:
1. Visit page
2. Provide user information for Review
Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.3.6.2 Notification

Confirmation

Use Case: Confirmation
Primary Actor: User, Admin

Goal in context: System can notify users

Preconditions:

1. System has been designed for notifications
2. System has interface for confirmation
3. User must be logged in
4. Check availability of places
Triggers: User can receive notification
Scenario:
1. Visit page
2. Provide user information for Notification
Exception:
· User not logged in
Priority: Essential, must be implemented.
When Available: First increment

4.4 Use Case Diagram:

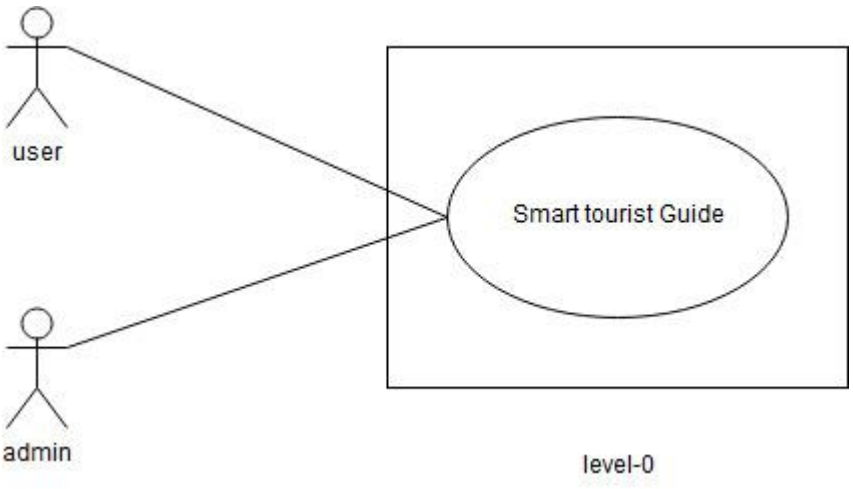


Figure 1: Level 0 Smart Tourist Guide

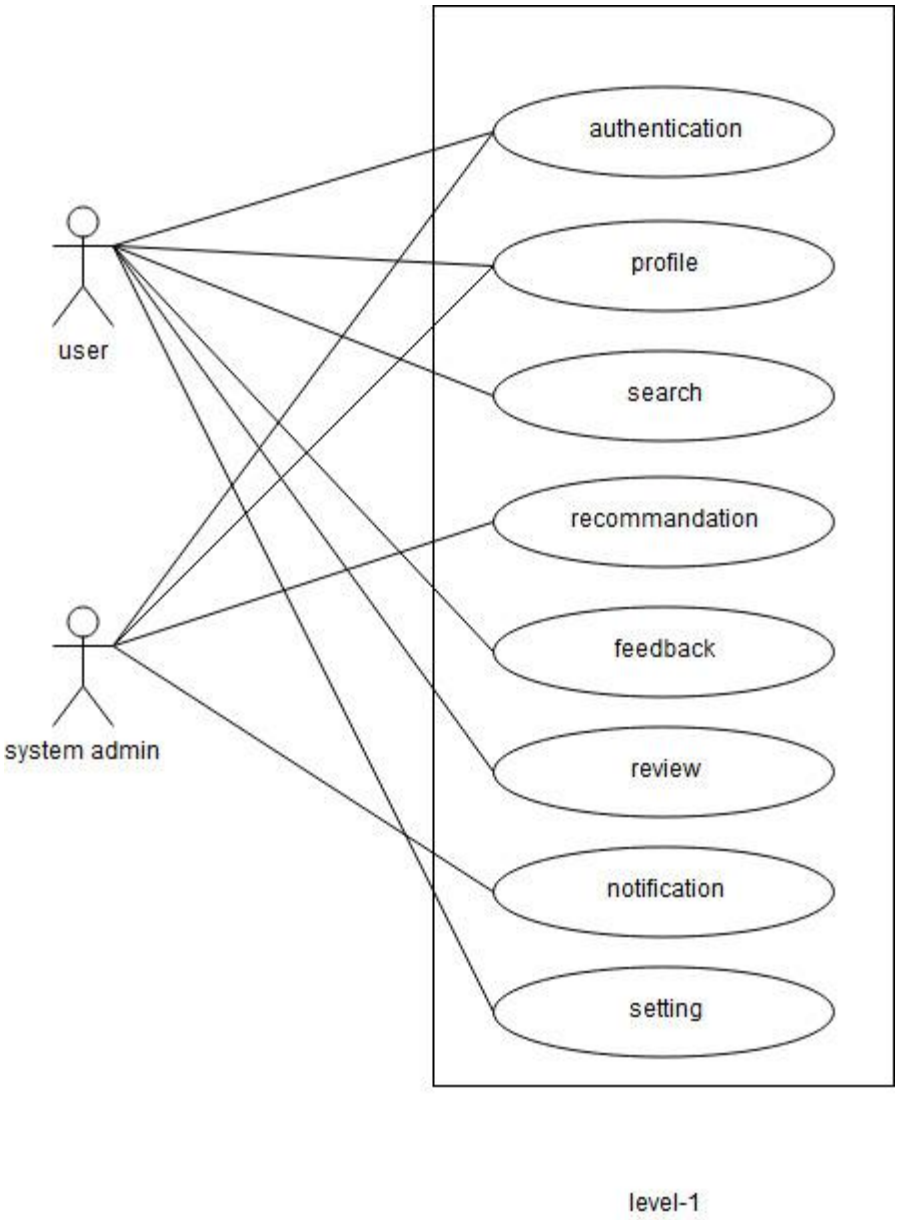


Figure 2: Level 1 Smart Tourist Guide

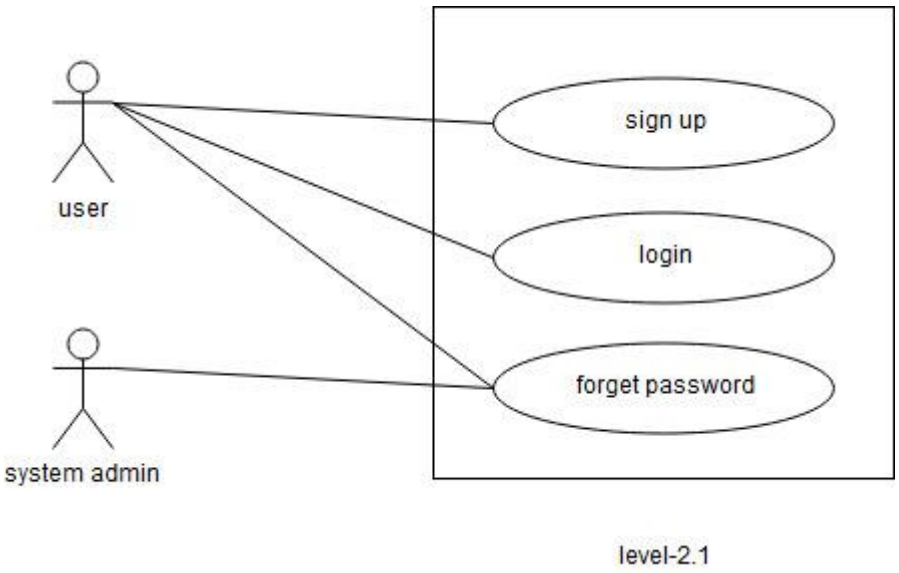


Figure 3: Level 2.1 Smart Tourist Guide

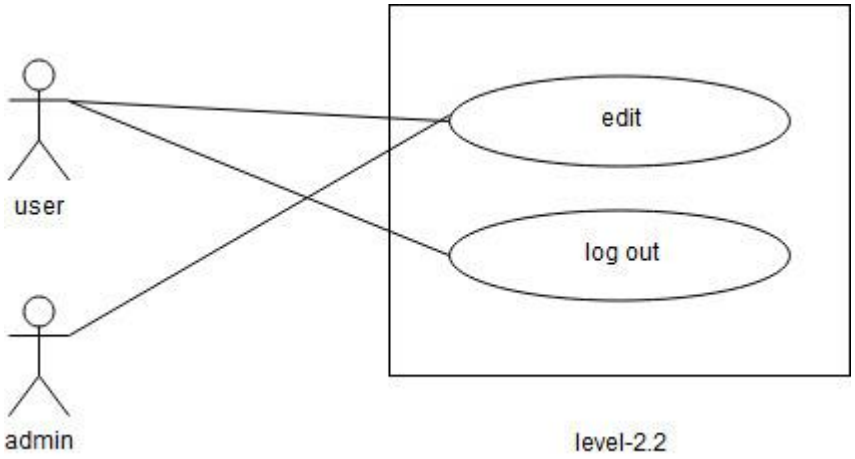


Figure 4: Level 2.2 Smart Tourist Guide

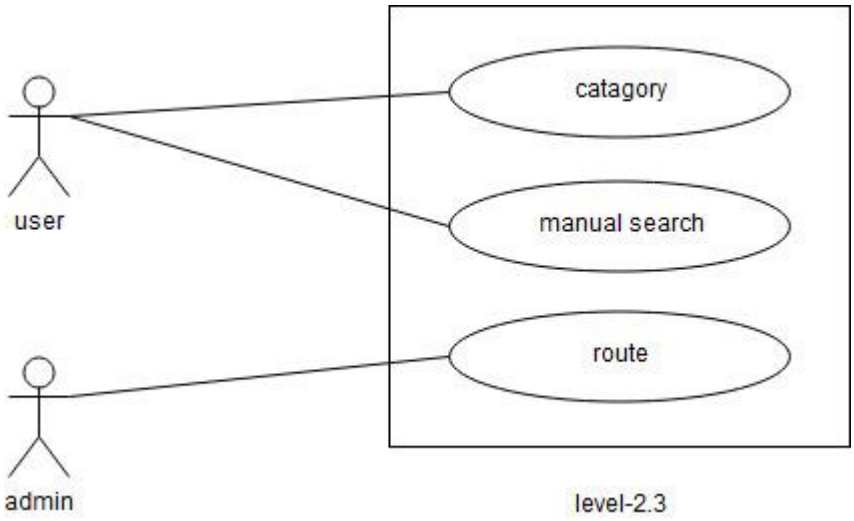


Figure 5: Level 2.3 Smart Tourist Guide

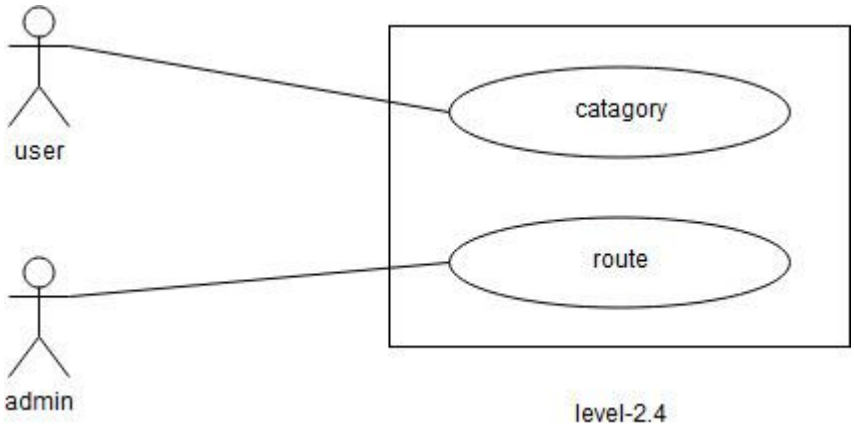


Figure 6: Level 2.4 Smart Tourist Guide

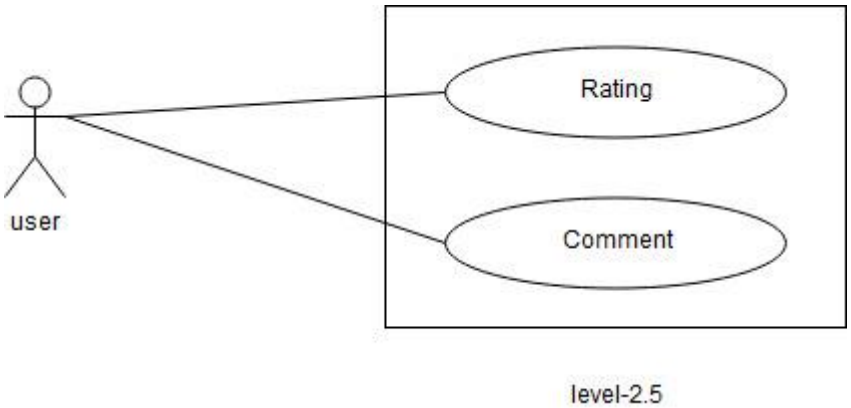


Figure 7: Level 2.5 Smart Tourist Guide

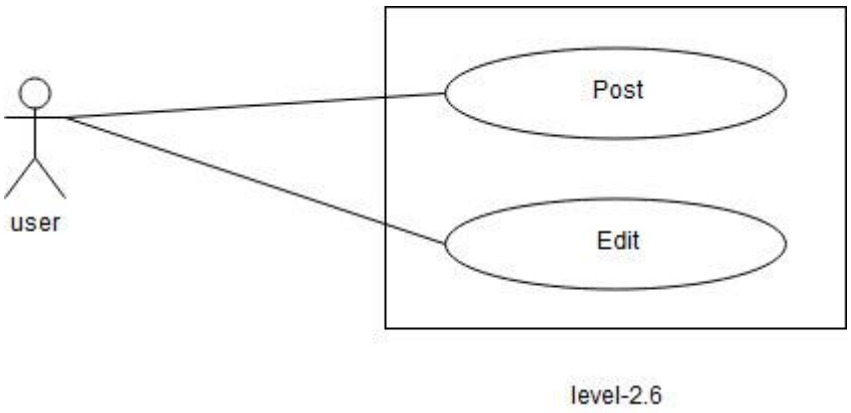


Figure 8: Level 2.6 Smart Tourist Guide

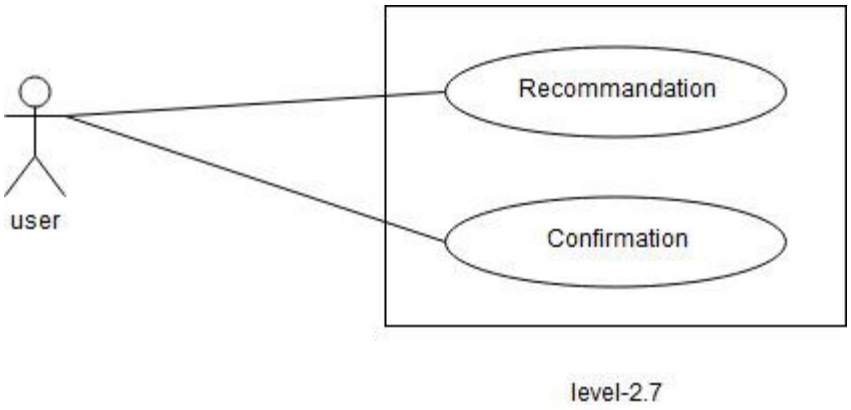


Figure 9: Level 2.7 Smart Tourist Guide

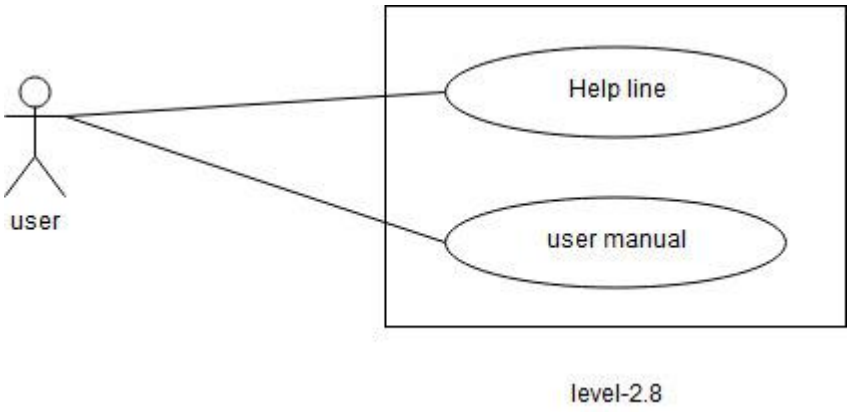


Figure 10: Level 2.8 Smart Tourist Guide

4.5 Activity Diagram and Swimlane Diagram of generated Use Cases:

Use case 1: Sign Up
Activity Diagram:

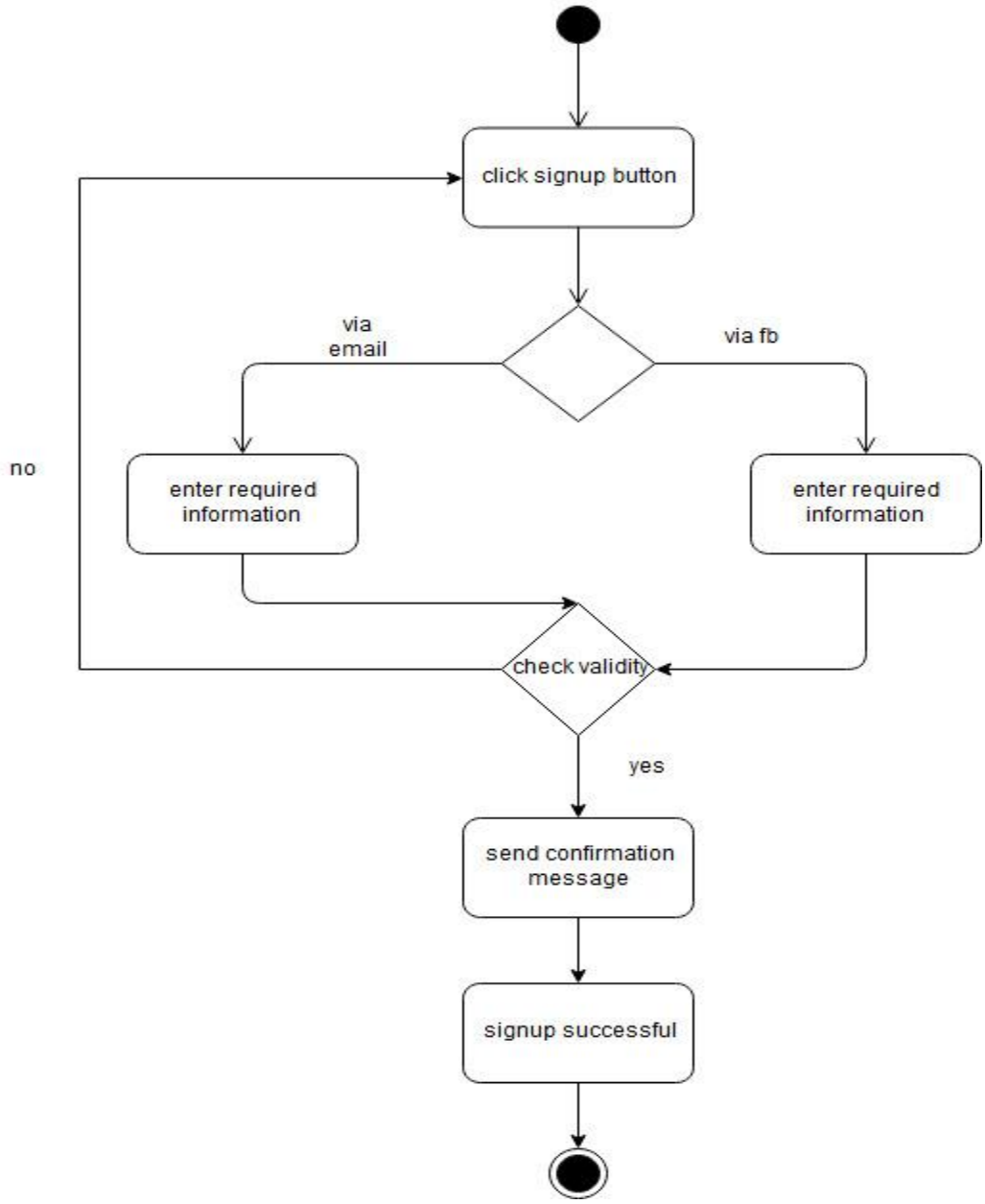


Figure 11: Activity Diagram for Sign-Up

Swimlane Diagram:

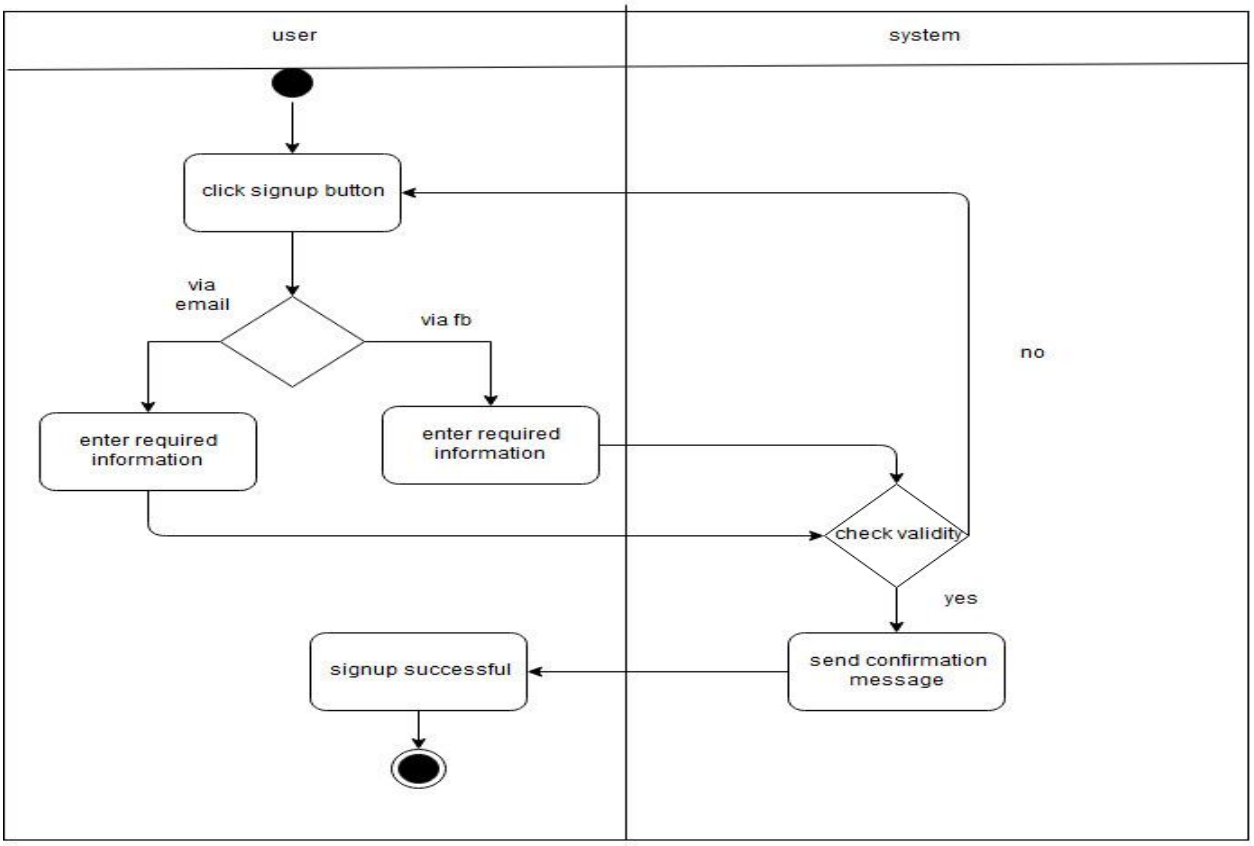


Figure 12: Swimlane Diagram for Sign up

Use case 2: Login

Activity Diagram:

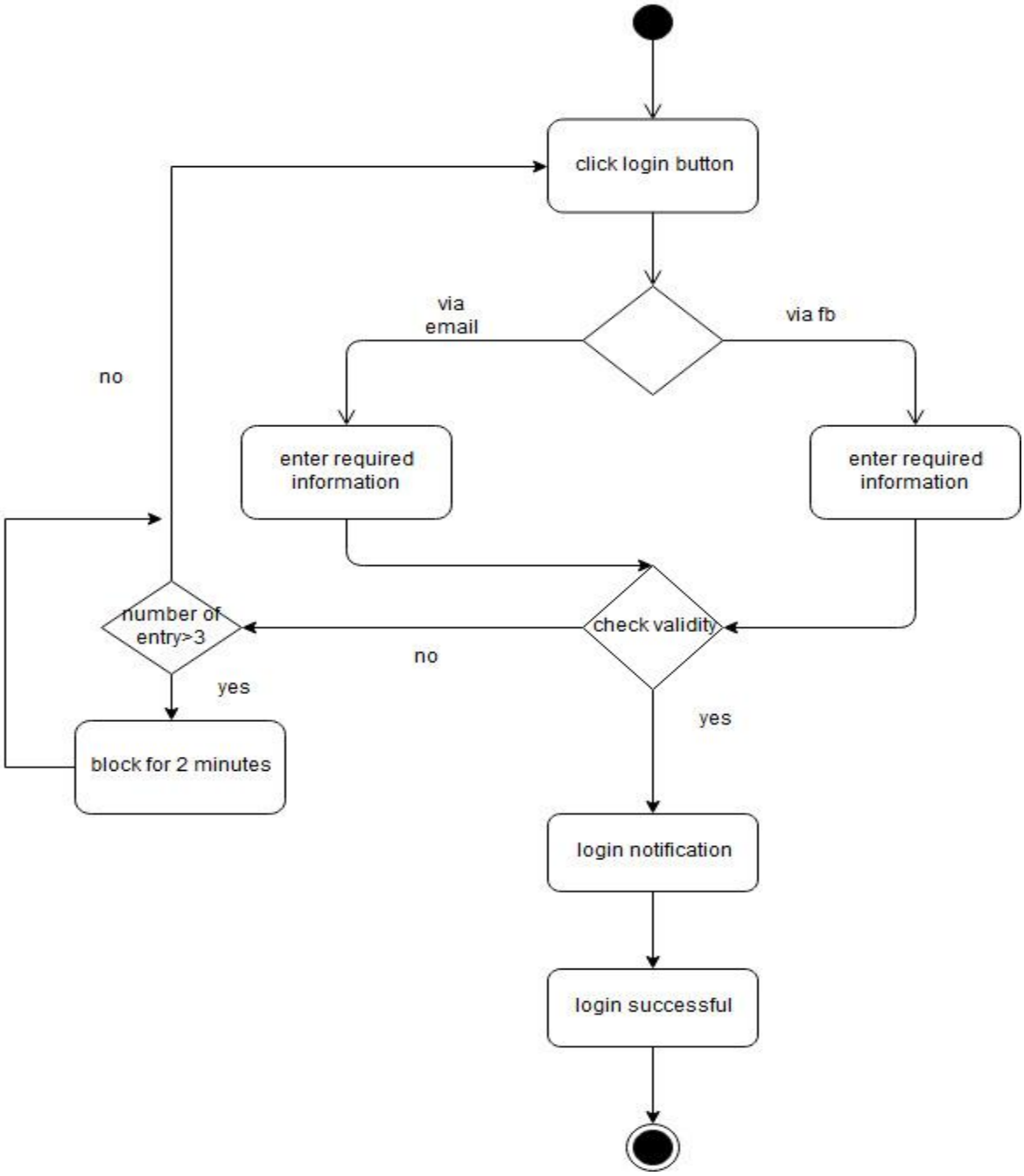


Figure 13: Activity Diagram for Login

Swimlane Diagram:

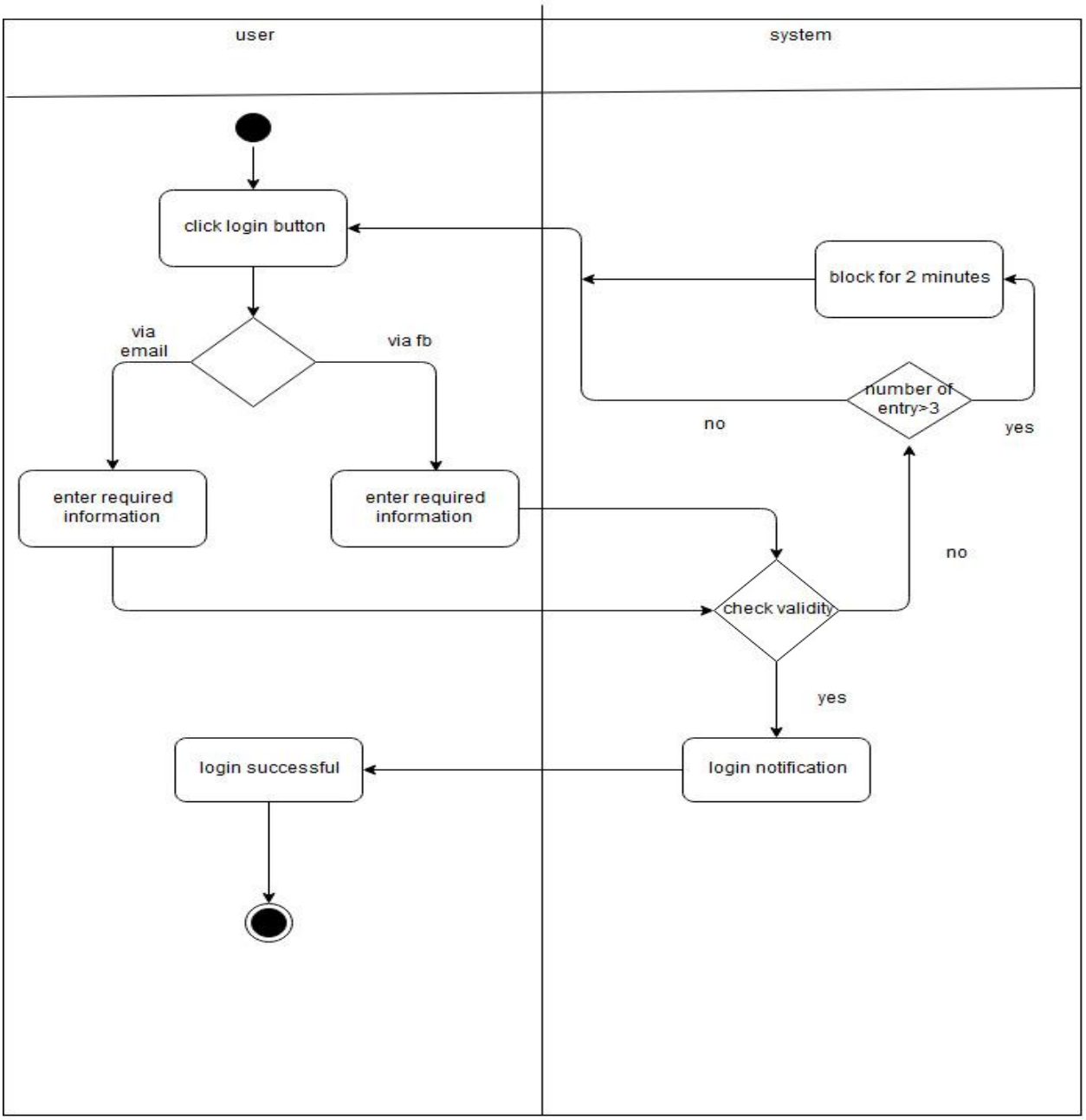


Figure 14: Swimlane Diagram for Login

Use case 3: Forget Password

Activity Diagram:

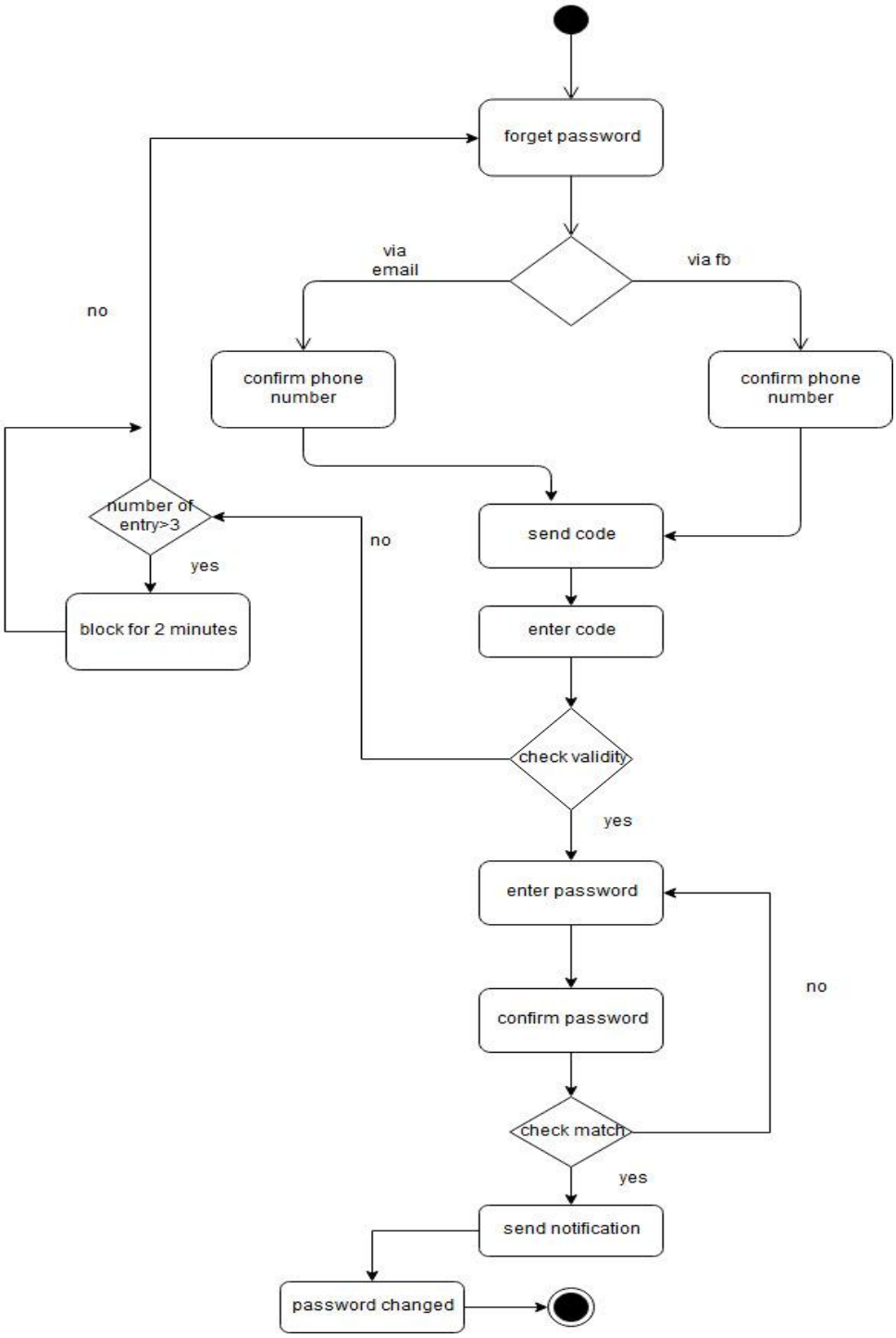


Figure 15: Activity Diagram for Forget Password

Swimlane Diagram:

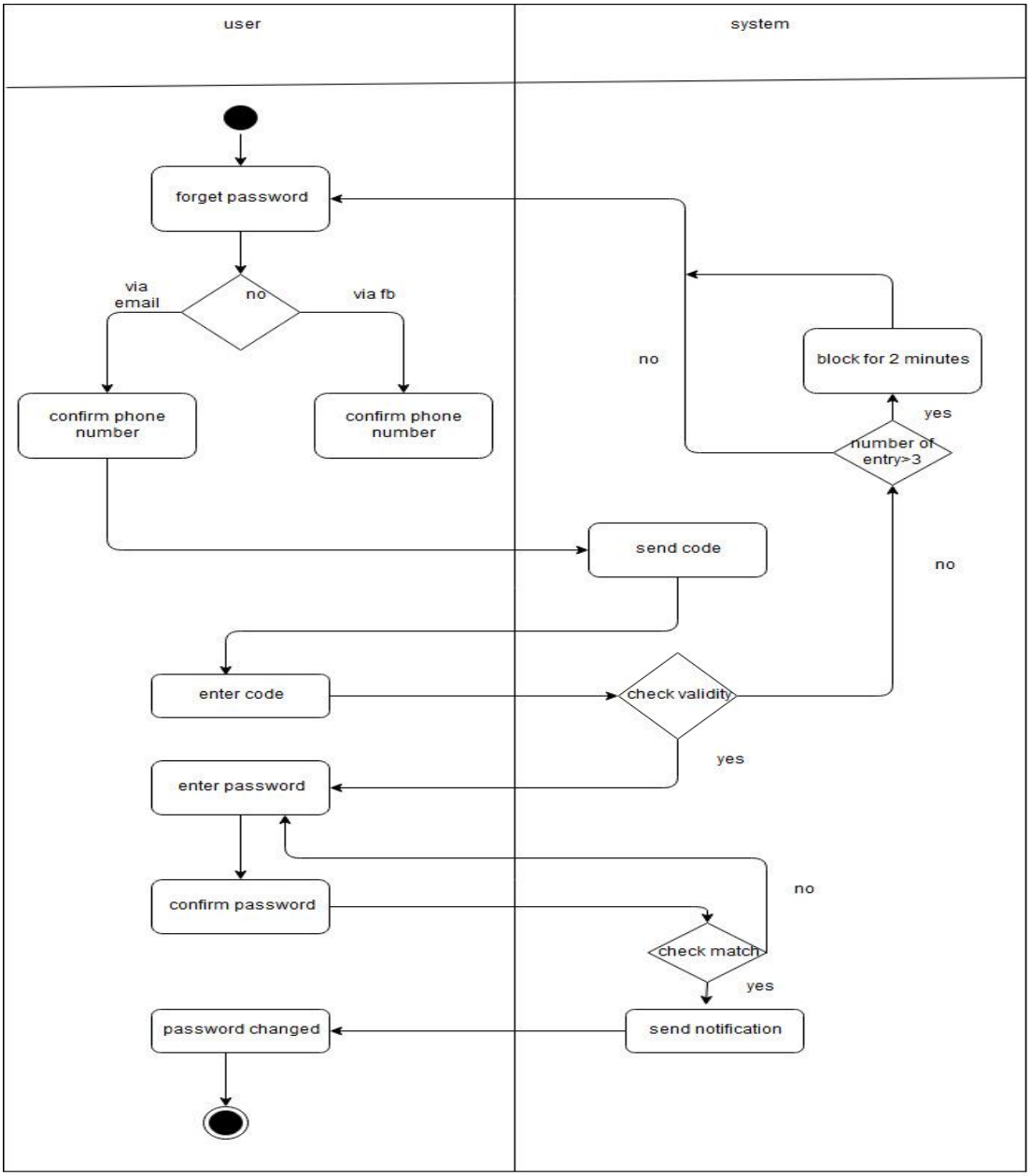


Figure 16: Swimlane Diagram for Foget Password

Use case 4: Sign Out

Activity Diagram:

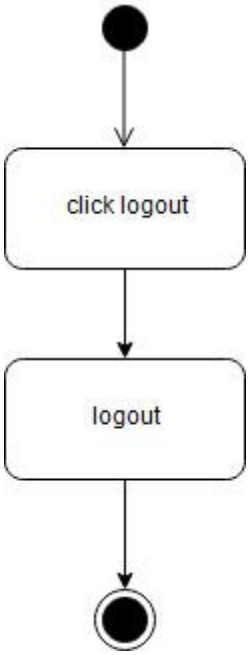


Figure 17: Activity Diagram for Log out

Swimlane Diagram:

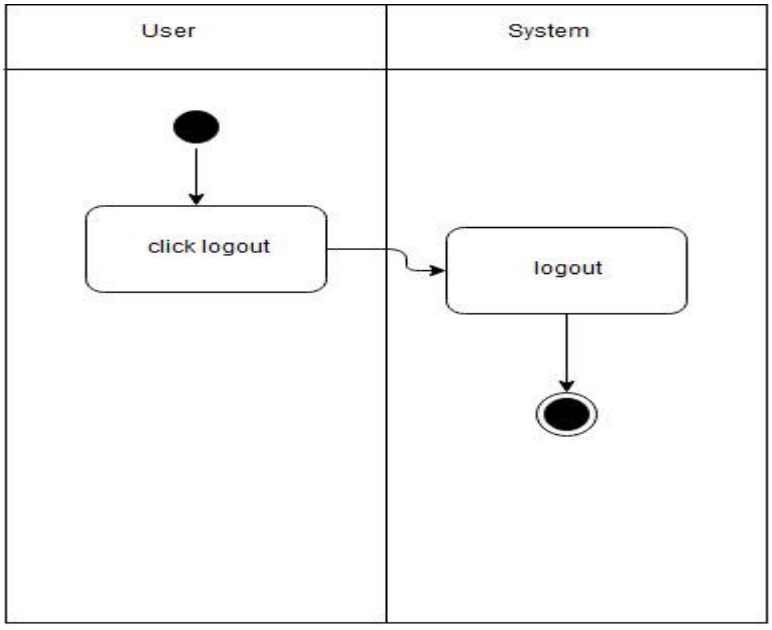


Figure 18: Swimlane Diagram for Logout

Use case 5: Edit Profile
Activity Diagram:

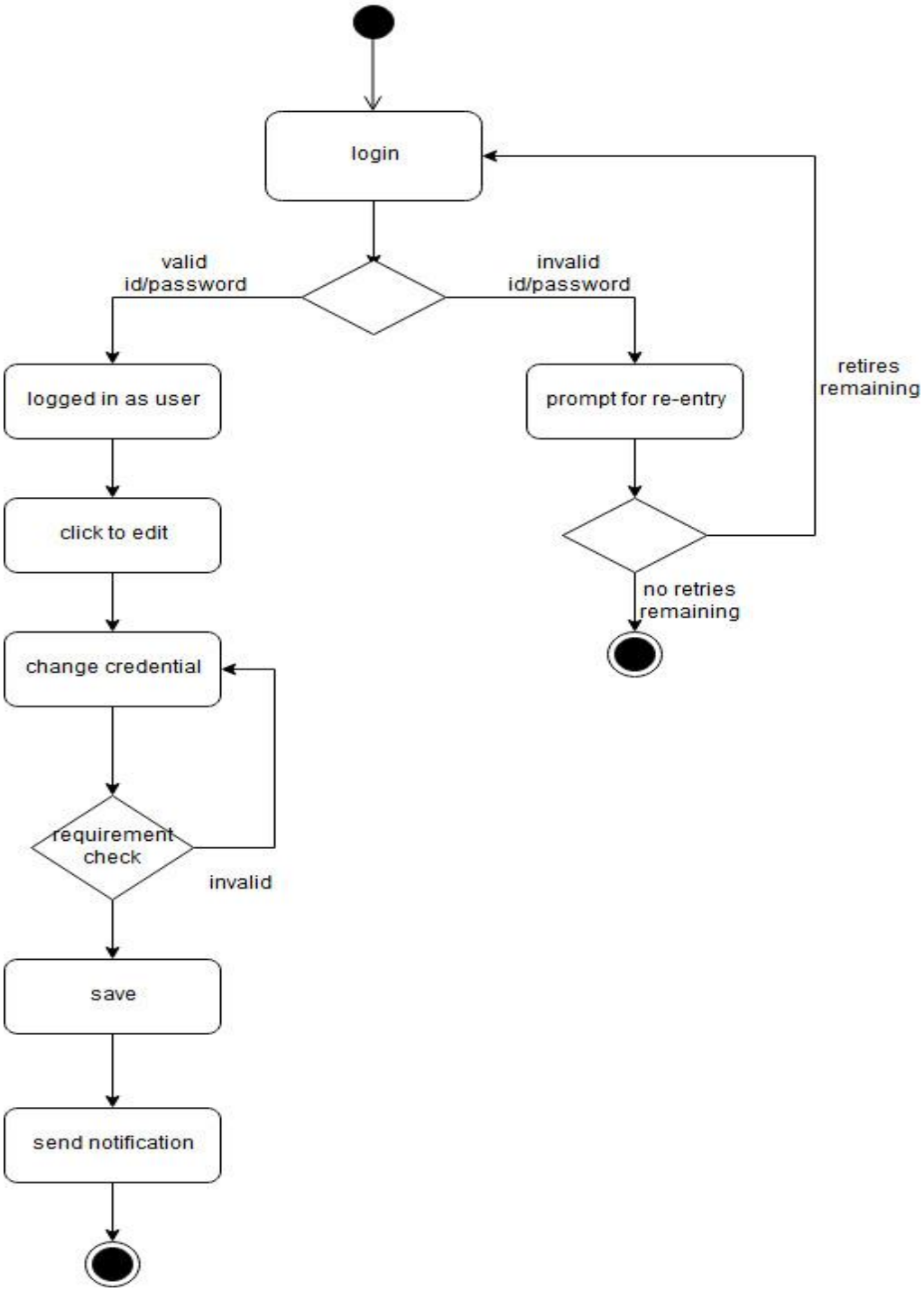


Figure 19: Activity Diagram for Edit Profile

Swimlane Dagram:

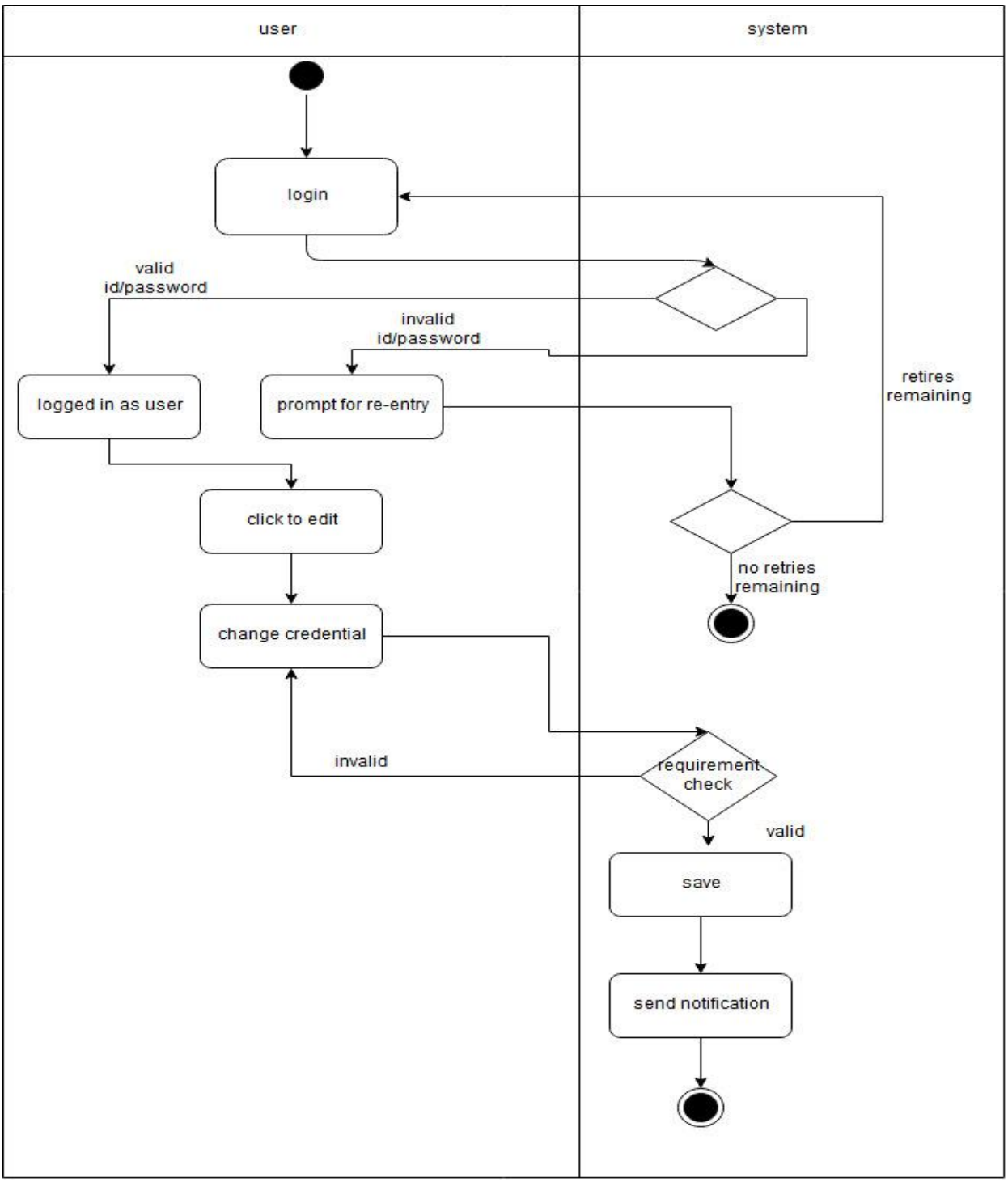


Figure 20: Swimlane Diagram for Edit Profile

Use case 6: Category search
Activity Diagram:

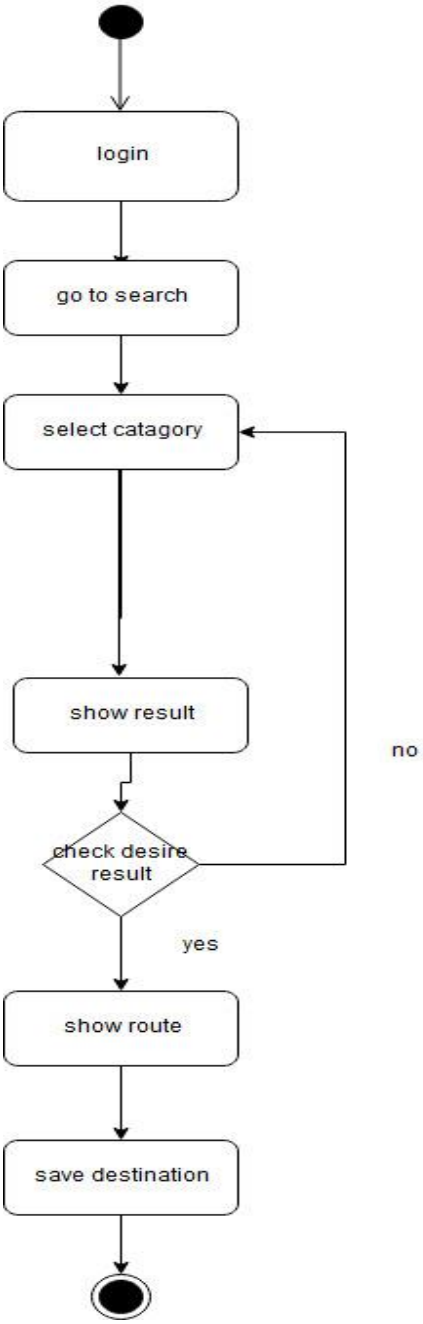


Figure 21: Activity Diagram for Category basis Search

Swimlane Diagram:

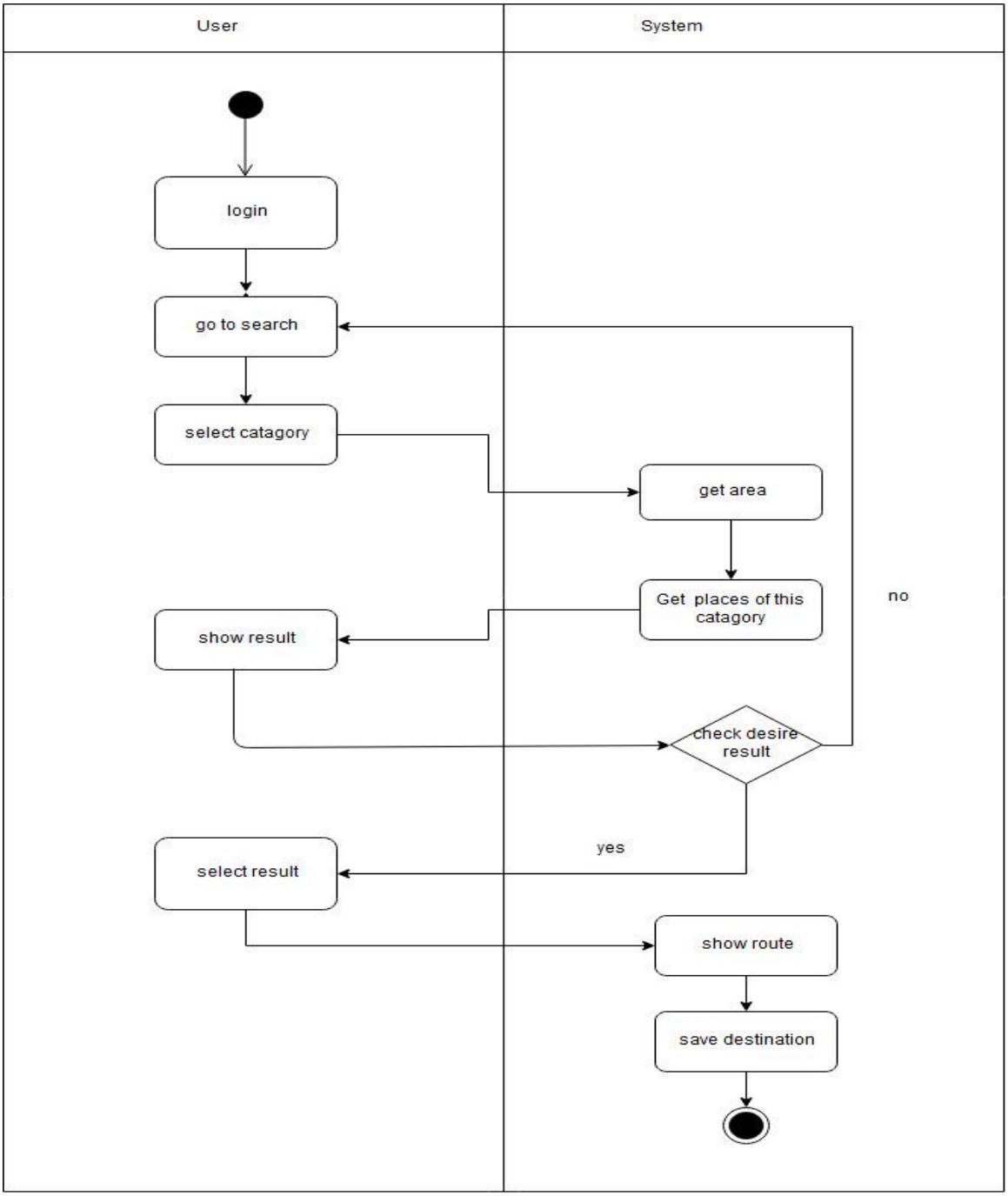


Figure 22: Swimlane Diagram for Category basis Search

Use case 7: Manual Search
Activity Diagram:

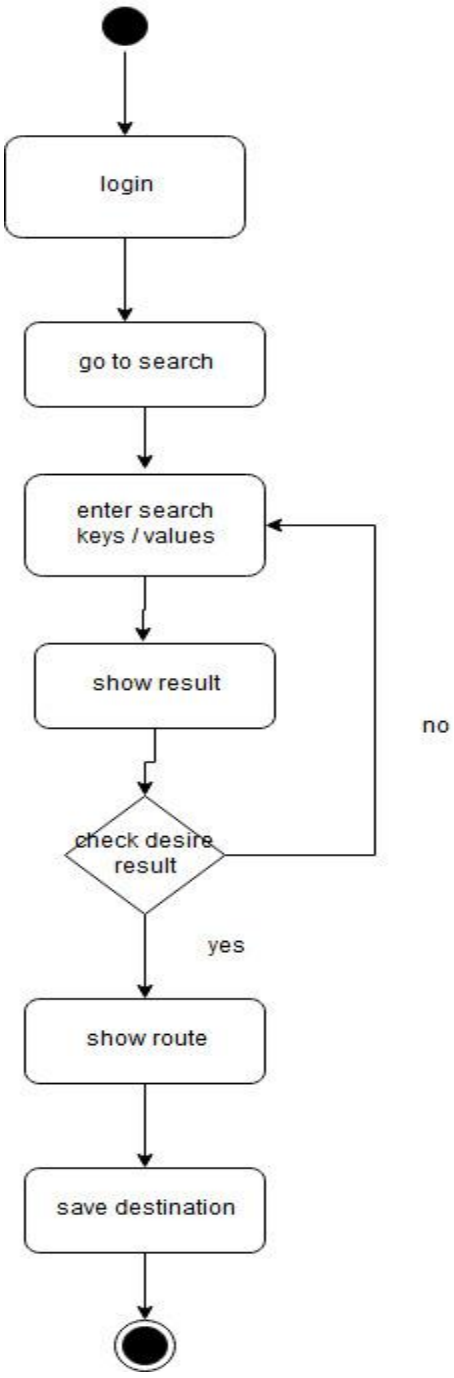


Figure 23: Activity Diagram for manual search

Swimlane Diagram:

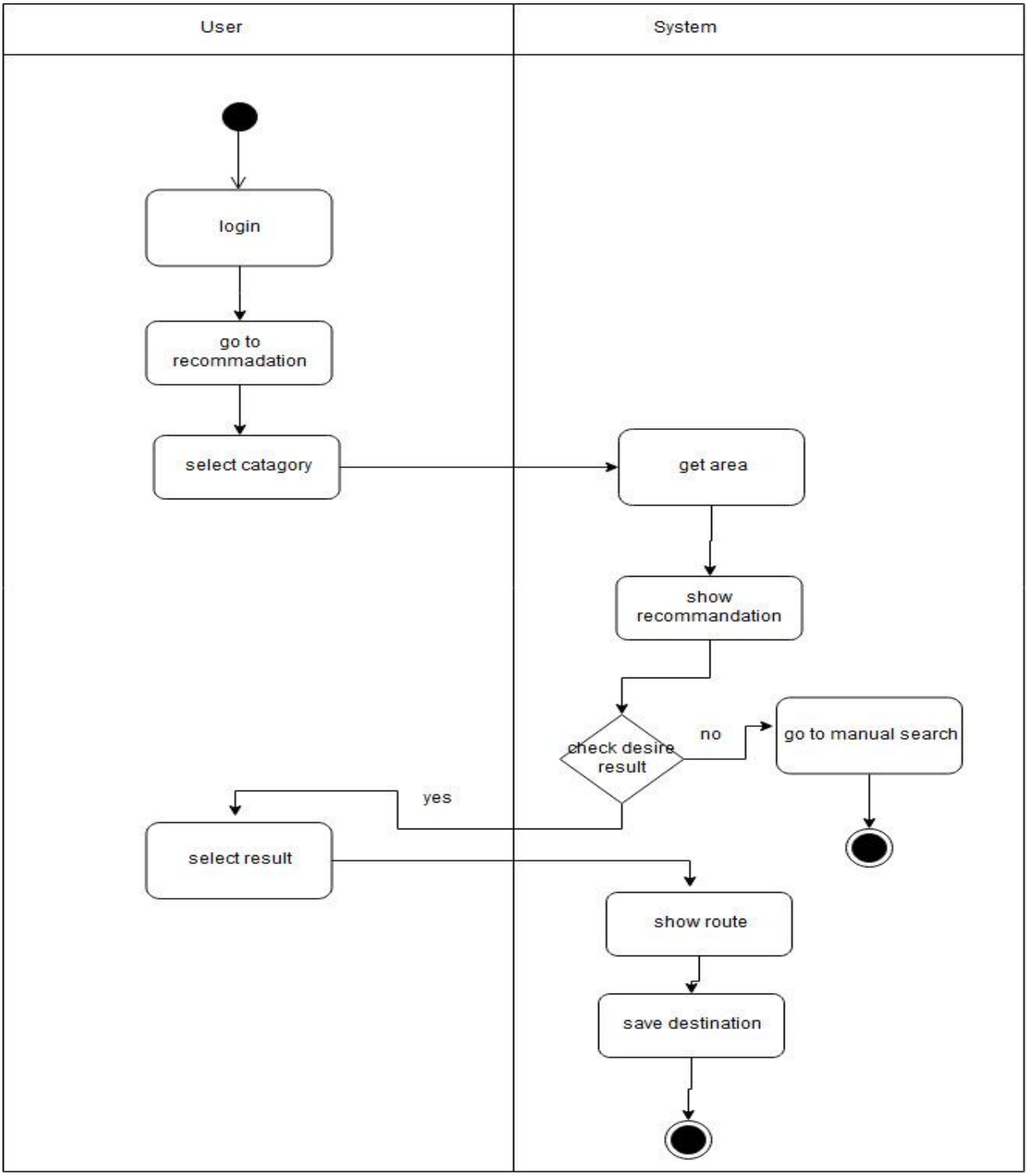


Figure 24: Swimlane Diagram for Manual Search

Use case 8: Recommendation

Activity Diagram:

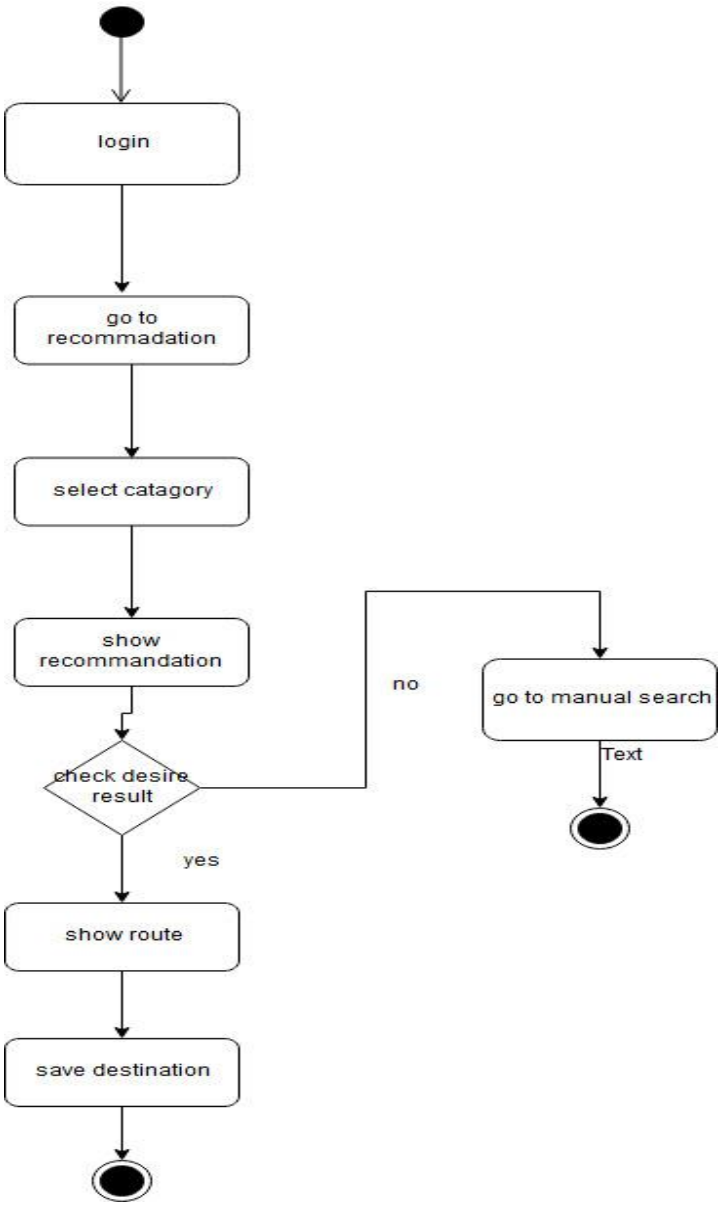


Figure 25: Activity Diagram for Recommendation

Swimlane Diagram:

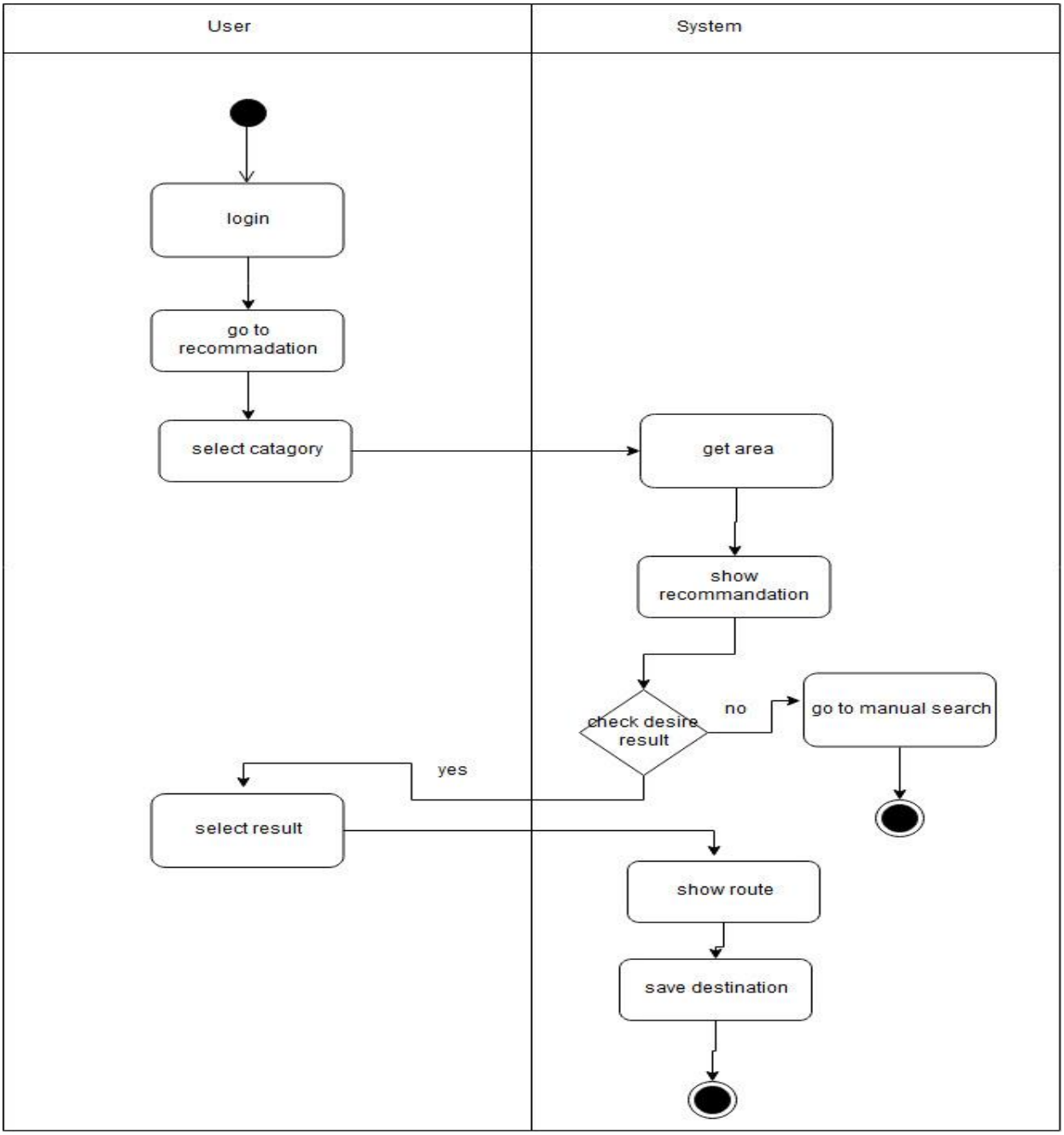


Figure 26: Swimlane Diagram for Recommendation

Use case 9: Route
Activity Diagram:

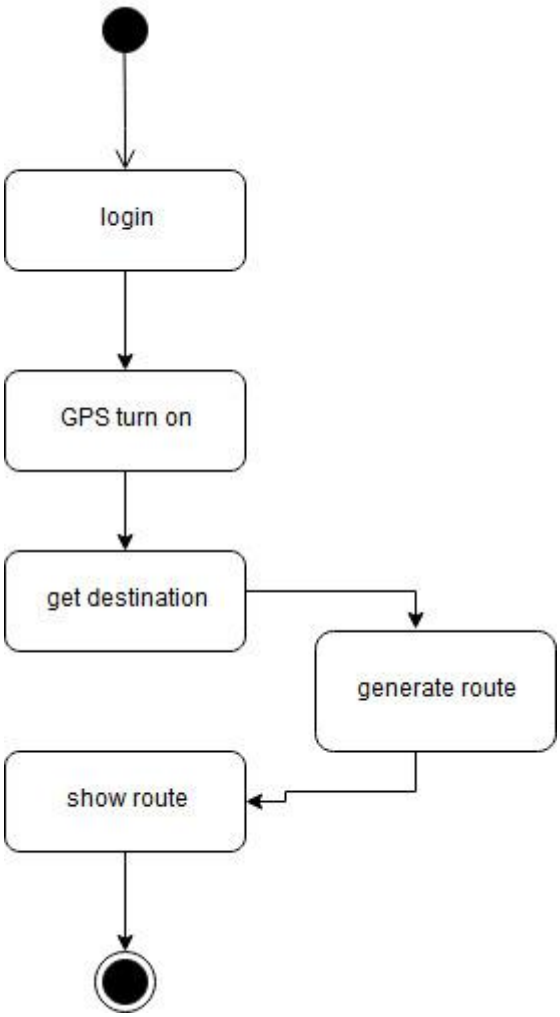


Figure 27: Activity Diagram for Route

Swimlane Diagram:

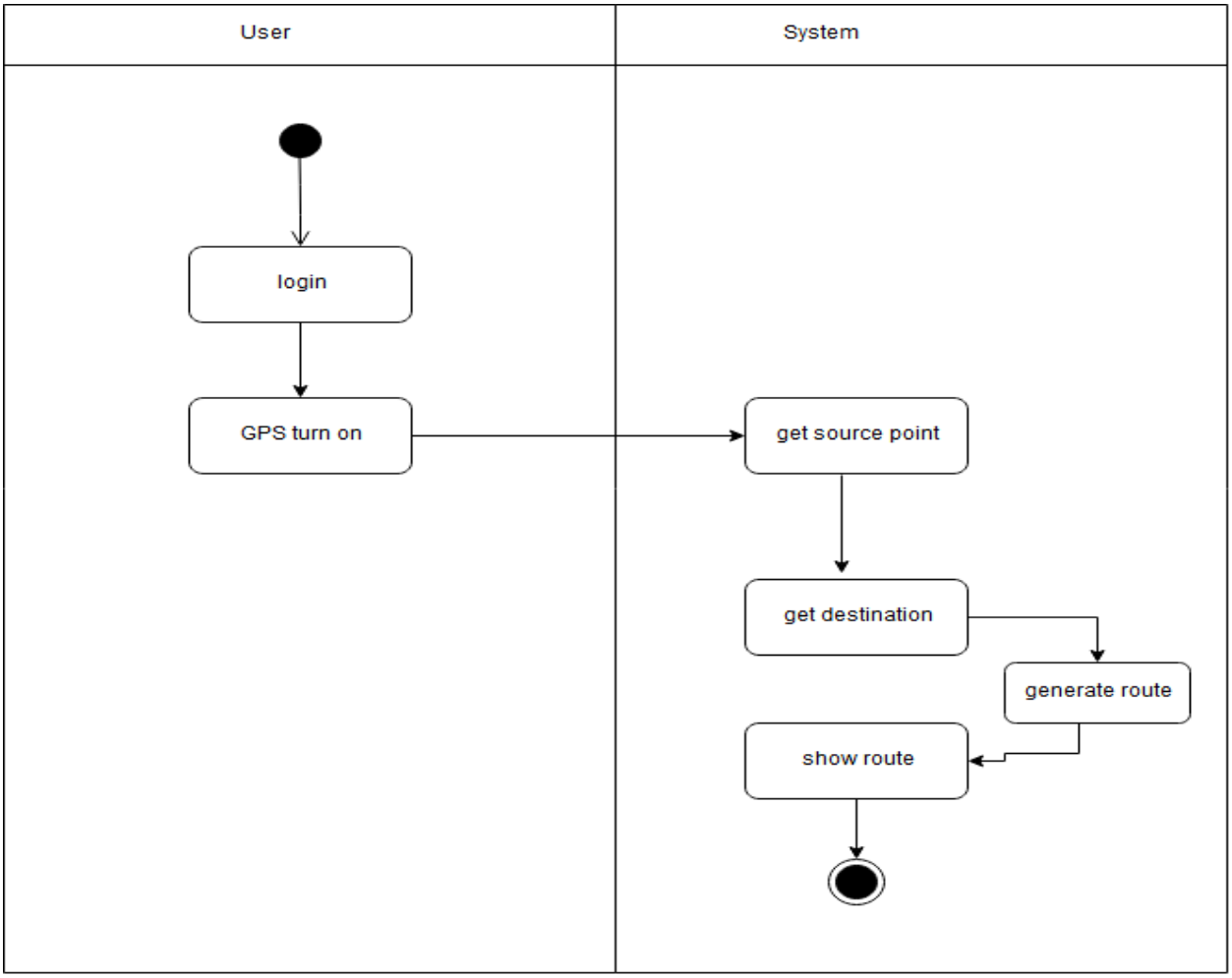


Figure 28: Swimlane Diagram for Route

Use case 10: Feedback
Activity Diagram:

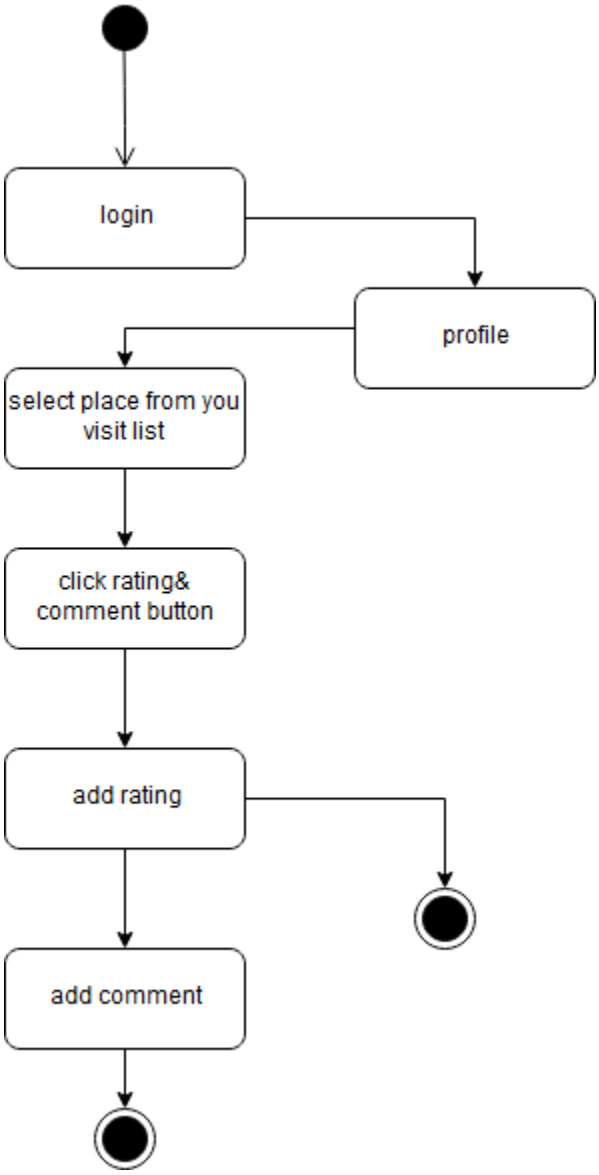


Figure 29: Activity Diagram for Feedback

Swimlane Diagram:

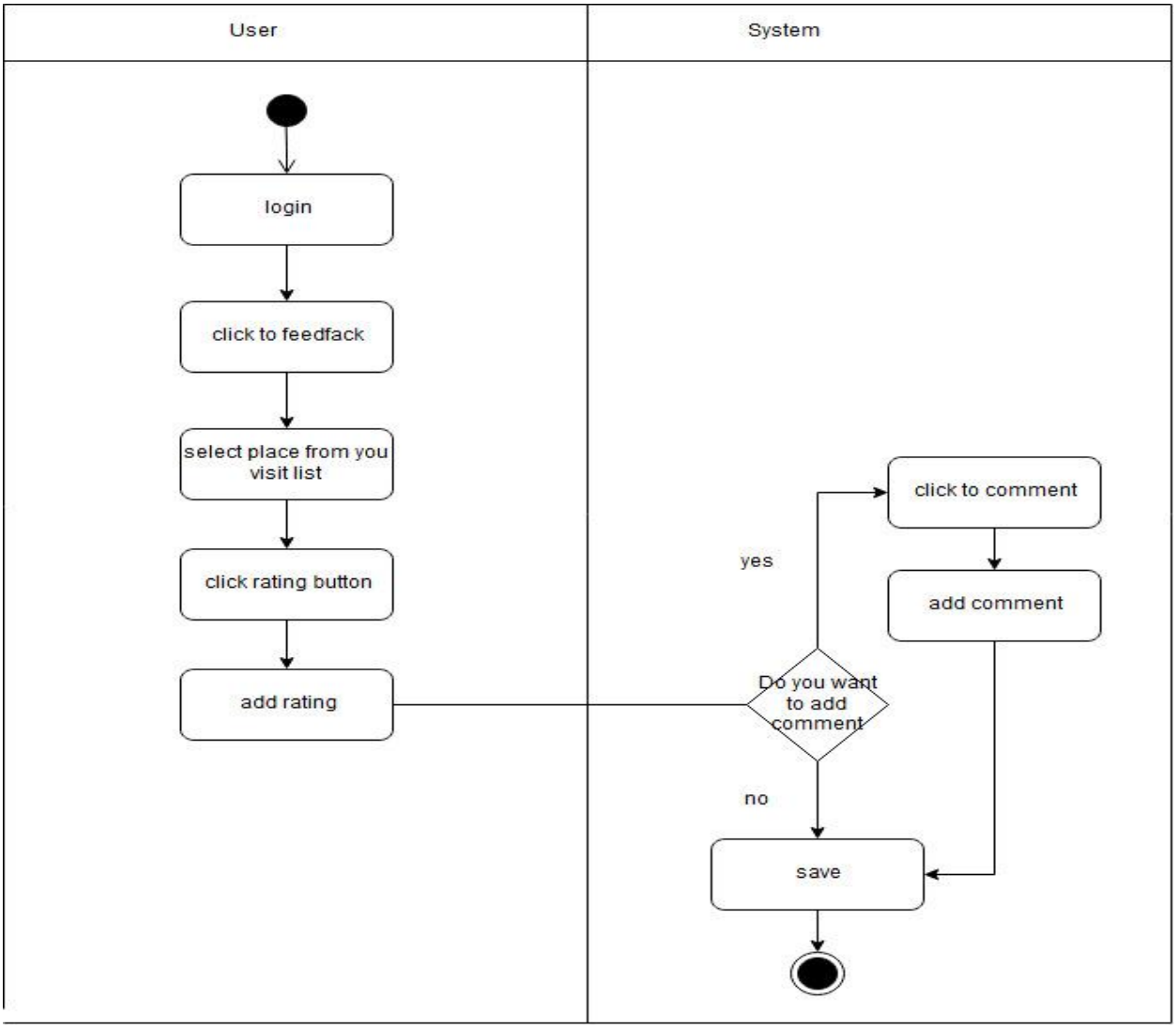


Figure 30: Swimlane Diagram for Feedback

Use case 11: Post
Activity Diagram:

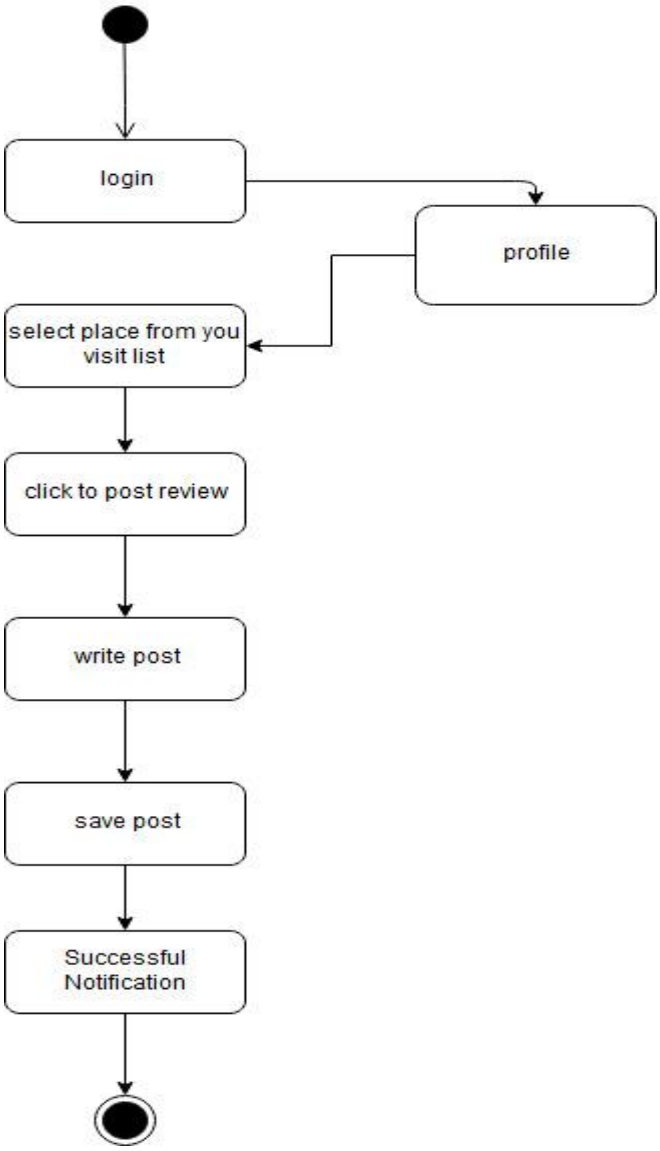


Figure 31: Activity Diagram for Post

Swimlane Diagram:

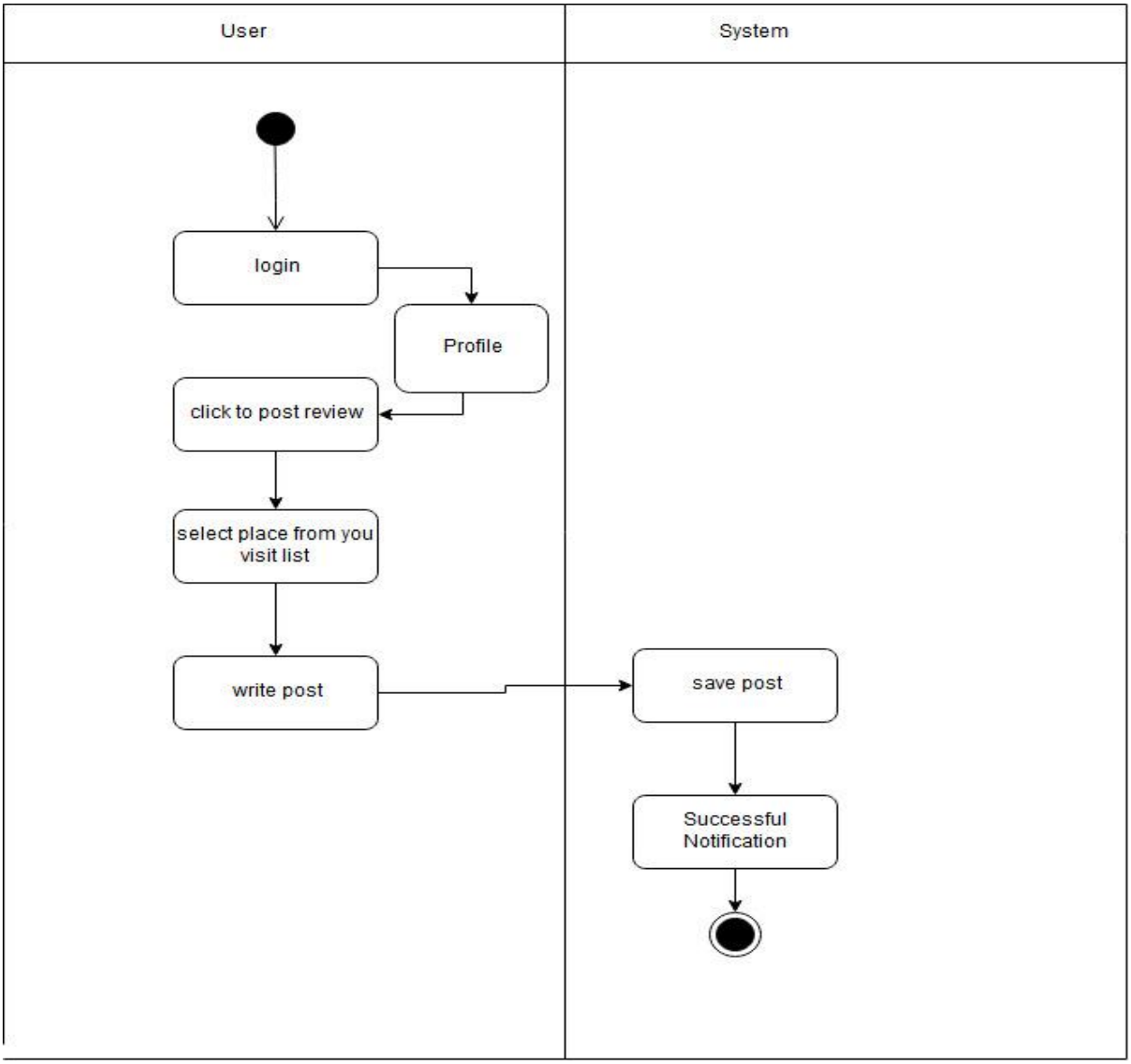


Figure 32: Swimlane Diagram for Post

Use case 12: Edit Post

Activity Diagram:

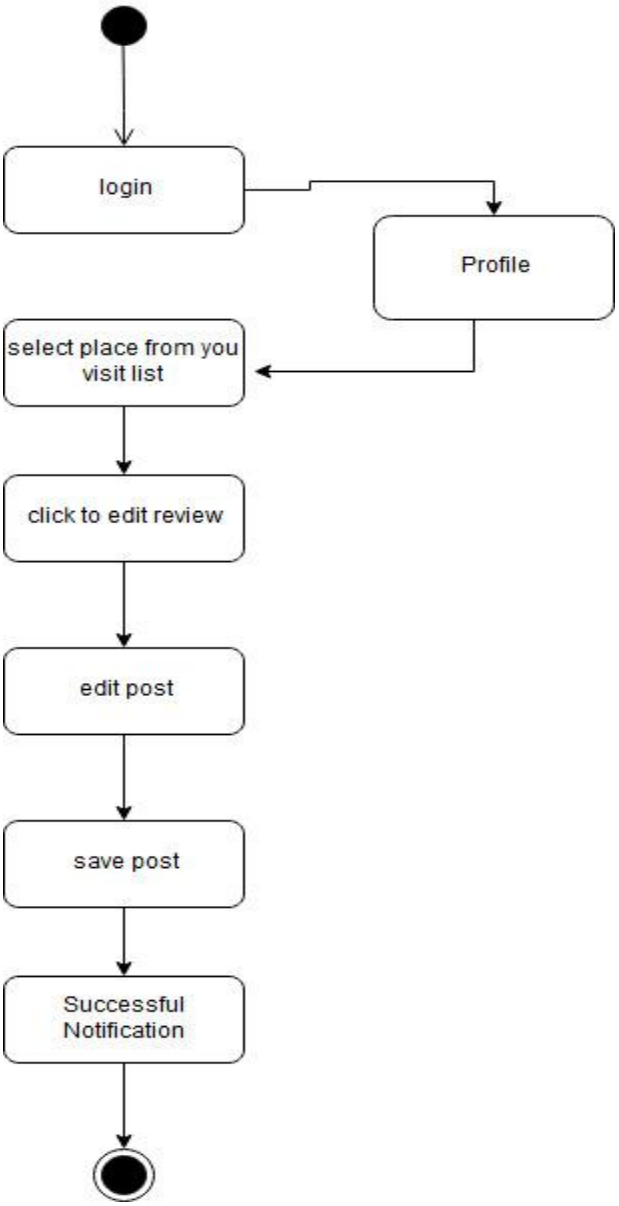


Figure 33: Activity Diagram for Edit Post

Smimlane Diagram:

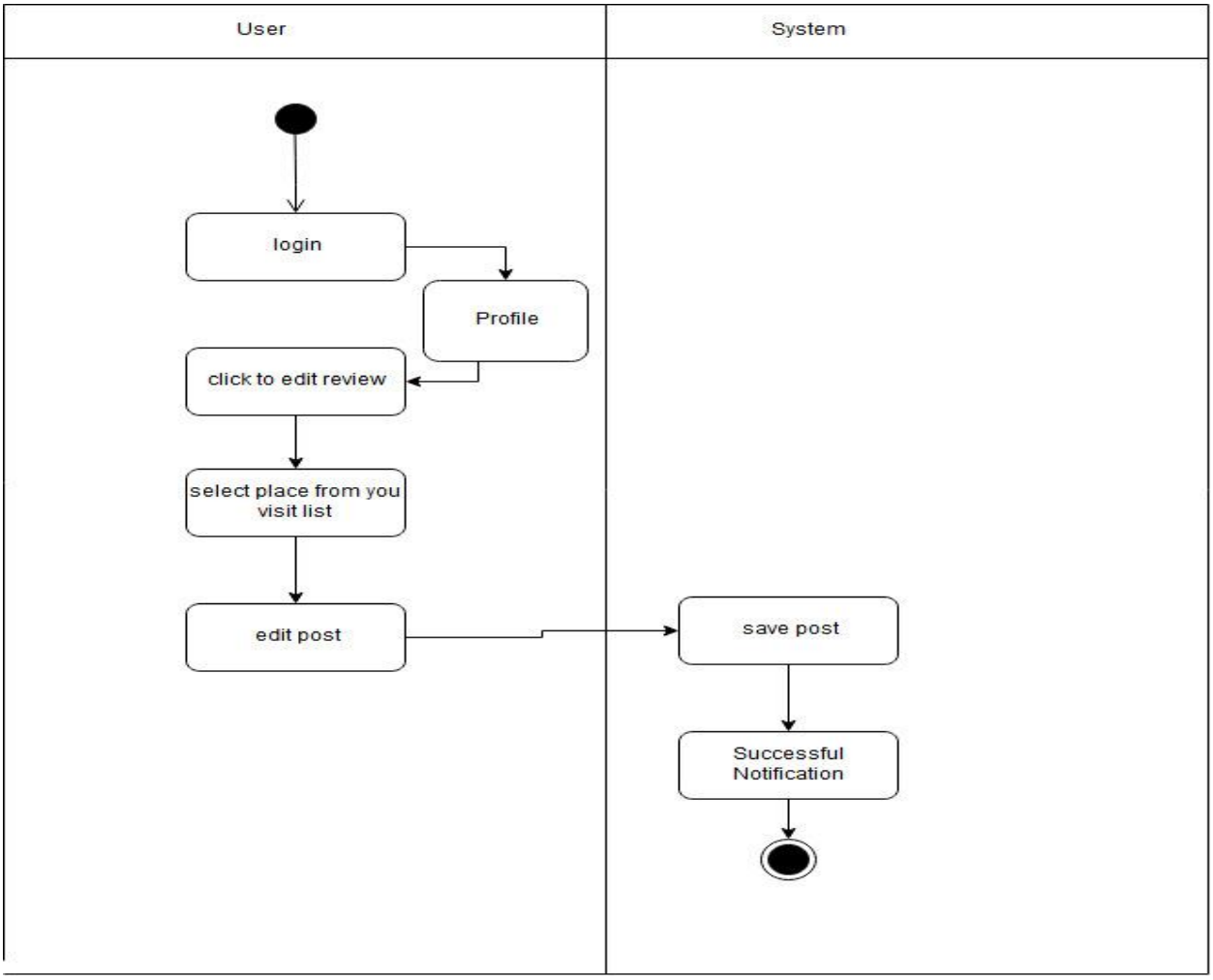


Figure 34: Swimlane Diagram for Edit Post

Chapter 5: Data Model

Introduction

Software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, the software team may choose to create a data model as part of overall requirements modeling.

Data Object Selection

A data object is a representation of information which has different properties or attributes that must be understood by software. Here is the table of potential data objects.

5.2.1 Noun Identification

Numbers	Nouns	Problem space/ Solution space	Attributes
1	System	S	2,3,11,24
2	User	P	10,4,5,7,19,21,20,25,27 30
3	Account	S	6,7,12,13,14,15
4	Facebook	P	
5	Email	S	
6	Phone Number	P	
7	Password	P	
8	Firebase	S	
9	Credentials	P	
10	Login	P	
11	Code	S	
12	Profile	S	
13	Age	P	
14	Gender	P	
15	Address	S	
16	Place	P	
17	Location		
18	Button		
19	Rating		16,32,34

20	Comment		
21	Feedback		19,20,2,3
22	Current Account		
23	Customer		
24	Recommendation		1,2,30,48
25	Searching		1,2,16,17,27,30-35
26	Verification		
27	GPS		
28	Google		
29	Internet		
30	Category		
31	Park		
32	Restaurant		
33	Hotel		
34	Cafe		
35	Zoo		
36	Museum		
37	Hospital		
38	Sea-beach		
39	Mountain		
40	River		
41	Water-fall		
42	Data-field		
43	Selection		
44	Source		
45	Destination		
46	Route		
47	Track		
48	Notification		1,2,24
49	Date		
50	History		
51	Database		2,71
52	Environment		
53	Confirmation		
54	Authentication		
55	Document		

56	Basis		
57	Window		
58	Option		
59	Term		
60	Journey		
61	Travel		
62	Procedure		
63	Message		
64	SMS		
65	ID		
66	Review		72,73,2,3
67	Image		
68	City		
69	Communication		
70	Position		
71	Admin		1,2,3
72	Post		
73	Edit		
74	Birthday		
75	Name		

Table 2: Noun Identification

5.2.2 Potential data objects

System: 2, 3, 11, 24

User: 10,4,5,7,19,21,20,25,27,30

Notification: 1,2,24

Database: 2, 71

Account: 6,7,12,13,14,15

Route: 30-35

Rating: 16, 32, 34

Review: 2, 3, 72, 73

Feedback: 19, 20, 2, 3

Recommend: 1, 2, 30, 48

Searching: 1, 2, 16, 17, 27, 30-35

Admin: 1, 2, 3

Category: 30-35

5.2.3 Analysis for finalizing Data Objects

1) Category and Searching are an extension of Searching data objects so category and searching data objects can be extended to Searching object.
2) Route and Recommendation are an extension of recommendation data objects so route and recommendation data objects can be extended to recommendation object.

3) Rating and Feedback are an extension of Feedback data objects so rating and feedback data objects can be extended to feedback object.

5.2.4 Finalized data objects

No	Entity	Attribute
1	User	User_id, user_name, email, Address,Phone
2	Account	Account-id, email, password
3	Admin	Admin-id, Email, phone, Password
4	Route	Route_id, source, destination.
5	System	System_name,Recommendation_no
6	Notification	Notification_id, date
7	Place	Place_id, longitude, latitude
8	Recommendation	Recommendation_no,place_id,route_id

Table 3: Finalizing data objects

5.3 Data Objects Relation:

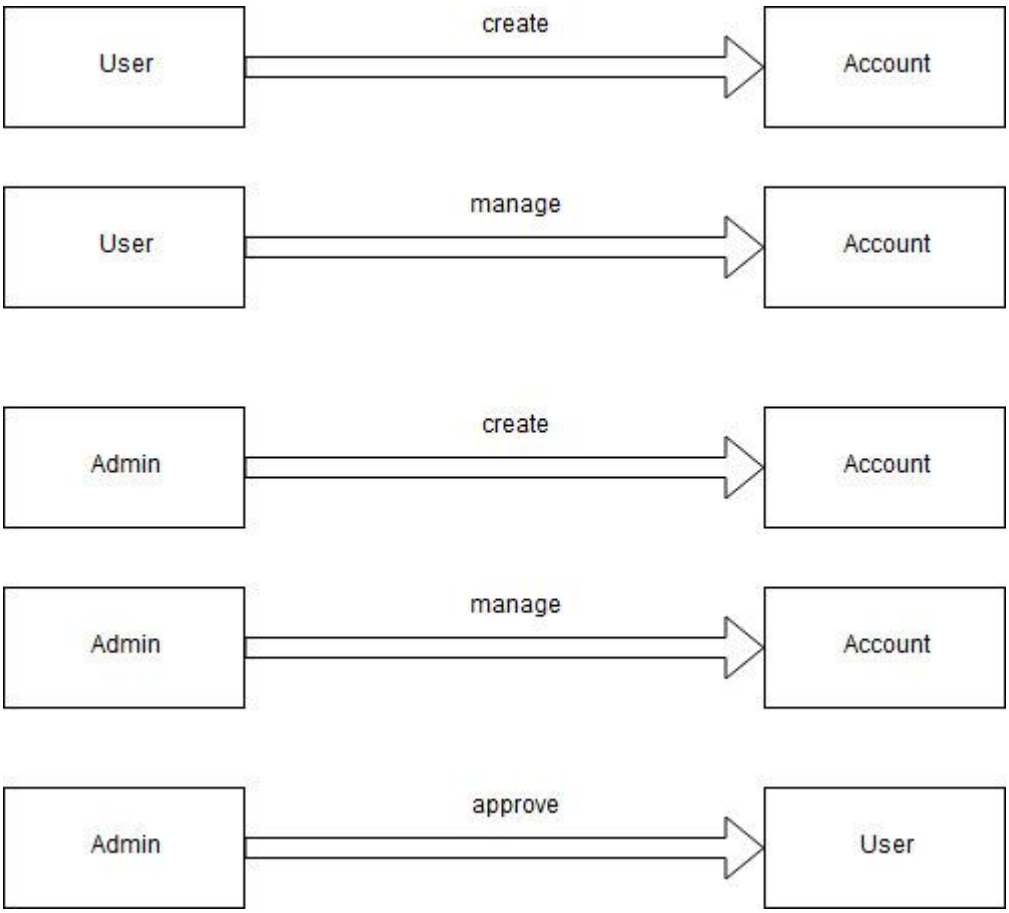


Fig 35:Data object relation

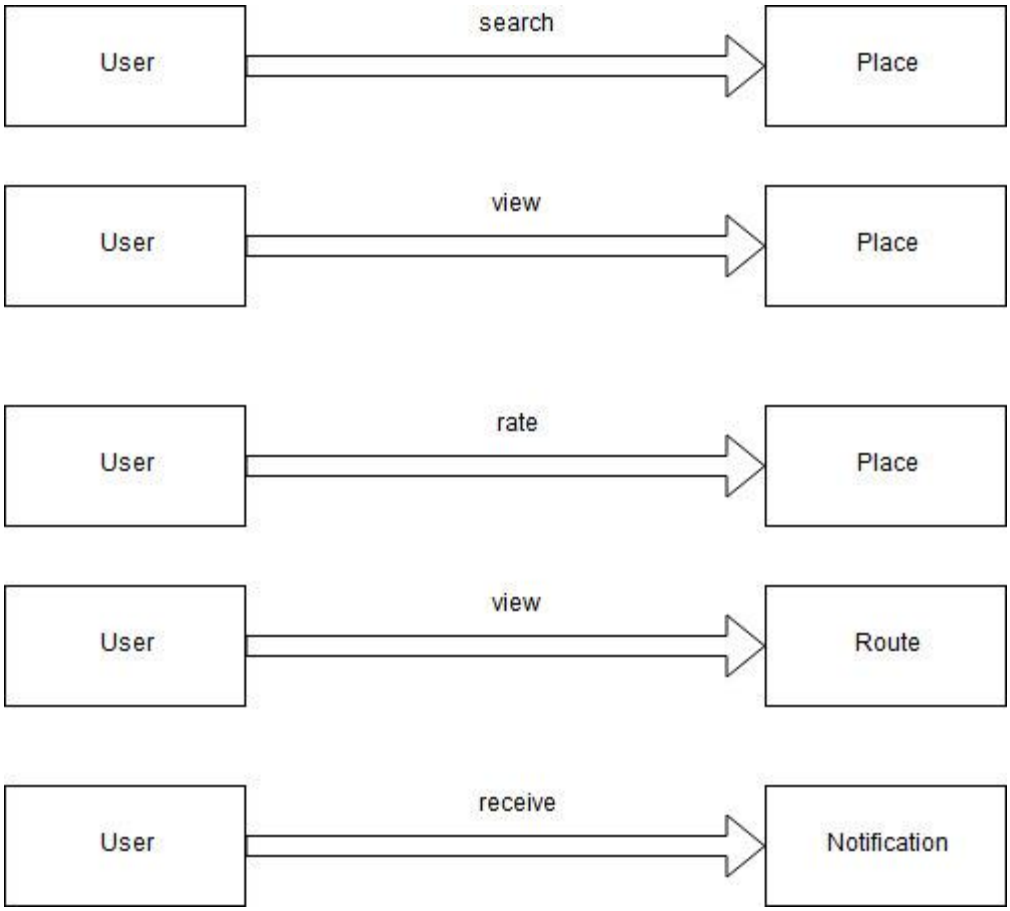


Fig 36: Data object relation

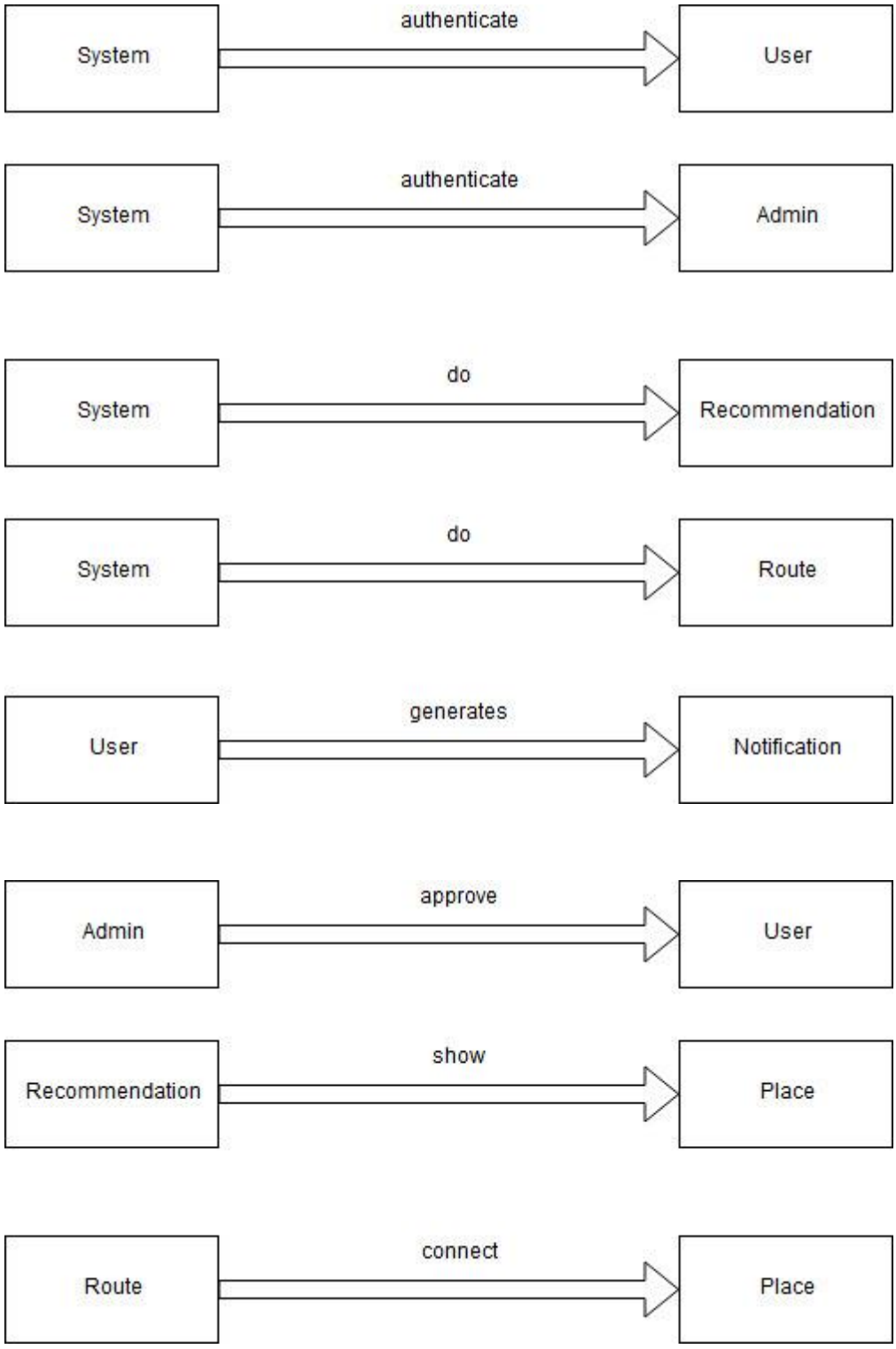


Fig 36: Data object relation

5.4 ER Diagram

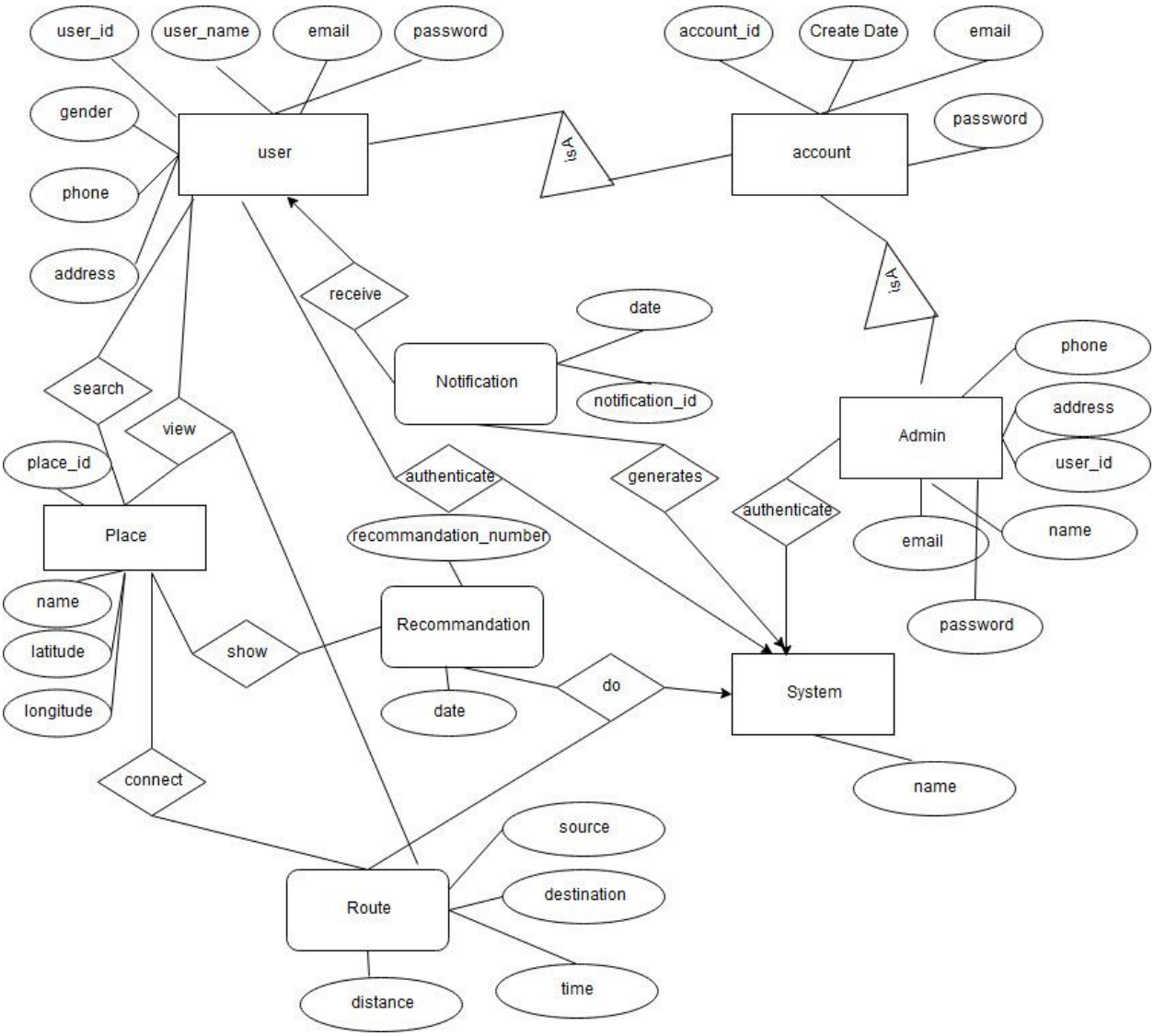


Fig 37: ER diagram

5.5 Schema Diagram

User		
Attribute	Type	Size
User_id	Number	20
User_name	Varchar2	30
Email	Varchar2	30
Password	Varchar2	30
Gender	Varchar2	20
Address	Varchar2	50
Phone	Number	20

Table 4: Schema Diagram (User)

Account		
Attribute	Type	Size
Account_id	Number	20
Create_date	Varchar2	30
Email	Varchar2	30
Password	Varchar2	30
Phone	Number	20

Table 5: Schema Diagram(Account)

Notification		
Attribute	Type	Size
Notification_id	Number	20
Date	Varchar2	30
User_id	Number	30
System.name	Varchar2	30

Table 6: Schema Diagram(Notification)

Place		
Attribute	Type	Size
Place_id	Number	20
Name	Varchar2	30
Longitude	Number	30
Latitude	Number	30
User_id	Number	30
Recommendation_no	Number	20

Table 7: Schema Diagram(Place)

Route		
Attribute	Type	Size
Route_id	Number	20
Distance	Varchar2	30
Estimated_Time	Number	30
Source	Varchar2	30
Destination	Varchar2	30

User_id	Number	30
Place_id	Number	30
System_name	Varchar2	30
Recommendation_no	Number	30

Table 8: Schema Diagram(Route)

System		
Attribute	Type	Size
Name	Varchar2	30
User_id	Number	30
Route_id	Number	30
Recommendation_no	Number	20

Table 9: Schema Diagram(System)

Recommendation		
Attribute	Type	Size
Date	Number	20
Name	Varchar2	30
Longitude	Number	30
Route_id	Number	30
Place_id	Number	30
Recommendation_no	Number	20

Table 10: Schema Diagram(Recommendation)

Chapter 6: Class Based Model

This Chapter is intended to describe class-based modeling of Pharmacy Management System.

6.1 Class Based Modeling Concept

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

6.2 General Classifications

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

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

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

Numbers	Nouns	General Classifications
1	System	4,7
2	User	3,4,5
3	Account	4
4	Facebook	2,7
5	Email	
6	Phone Number	
7	Password	
8	Firebase	
9	Credentials	

10	Login	3
11	Code	
12	Profile	3
13	Age	
14	Gender	
15	Address	
16	Place	
17	Location	1
18	Button	
19	Rating	
20	Comment	
21	Feedback	
22	Current Account	
23	Customer	
24	Recommendation	5,7
25	Searching	5,7
26	Verify	
27	GPS	
28	Google	
29	Internet	
30	Category	
31	Park	
32	Restaurant	
33	Hotel	
34	Café	
35	Zoo	
36	Museum	
37	Hospital	
38	Sea-beach	
39	Mountain	
40	River	
41	Water-fall	
42	Data-field	
43	Selection	
44	Source	
45	Destination	

46	Route	
47	Track	
48	Notification	
49	Date	
50	History	
51	Database	
52	Environment	
53	Confirmation	
54	Authentication	
55	Document	
56	Basis	
57	Window	
58	Option	
59	Term	
60	Journey	
61	Travel	
62	Procedure	
63	Message	
64	SMS	
65	ID	
66	Review	
67	Image	
68	City	
69	Communication	
70	Position	
71	Admin	4,5,7
72	Way	
73	Button	
74	Birthday	
75	Name	
76	Verification	

Table 11: General Classifications

6.3 Selection Criteria

The potential classes were then selected as classes by six Selection Criteria.
A potential class becomes a class when it fulfills all six characteristics.

1. Retained Information
2. Needed Services
3. Multiple Attributes
4. Common attributes
5. Common operations
6. Essential requirements

Numbers	Nouns	Selection
1	Database	6
2	Searching	1,2,3
3	Feedback	2,3
4	Account	1,2,3,5
5	Admin	1,2,3,4,5
6	Recommendation	2,6
7	User	1,2,3,4,5
8	Review	3
9	System	1,2,3
10	Notifications	2,3,4,5

Table 12: Selection Criteria

6.4 Associate Noun and Verb Identification

The nouns and the verbs associated with the potential classes are identified to find out the attributes and methods of each class.

No	Potential Class	Noun	Verbs
1	Account	User, Admin, Verification code	Sign in, Sign Up, send verification code
2	User	First name Last name User id Email Facebook Id Password	Edit profile search, update user information, recover password
3	Admin	Email Facebook Id Password Address	Approves request block user, change type
4	System	Place User	Show routes, generate notification Give recommendation
5	Notification	Id Receiver	Will be generated and sent
6	Feedback	Rating Comment place	Give rating
7	Review	Post Edit Post	Write post Share experience
8	Database	User Place	Delete information, save information, Update information, retrieve information

9	Search	Category Route	Search by place of interest, search by category
10	Recommendation	Notification place	Recommends places, Sent notification of recommended places

Table 13: Noun and verb identification

6.5 Attribute Selection

No	Potential Class	Noun
1	Account	User, Admin, Verification code
2	User	First name Last name User id Email Facebook Id Password
3	Admin	Email Facebook Id Password Address
4	System	Place User
5	Notification	Id Receiver
6	Feedback	Rating Comment place
7	Review	Post Edit Post

8	Database	User Place
9	Search	Category Route
10	Recommendation	Notification place

Table 14: Attribute Selection

6.6 Methods Identification:

No	Class	Methods
1	Account	signUp() logIn() signOut() sendVerificationCode() recoverPassword() verifyPassword() verifyEmail() verifyUser() manageForgetPassword()
2	User	setUserID() getUserID() setFirstName() getFirstName() setLastName() getLastName() setEmail() getEmail() setPhoneNo() getPhoneNo() setAddress() getAddress() comment() ratePlace() searchPlace() editProfile()

3	Admin	getName() setName() approvesRequest() setEmail() getEmail() setPhoneNo() getPhoneNo() setAddress() getAddress() SetAdmminId() getAdminId() setPhoneNo() getPhoneNo()
4	System	generateNotification() showSearchedItem() sendRecommendation() generateRecommendation()
5	Notification	sentNotification () getSender() setSender() setReceiver() getReceiver()
6	Feedback	setRating() getRating() setComment() writeComment()
7	Review	writePost() editPost() getDescription() setDescription()
8	Database	insertInfo() updateInfo() deleteInfo() retrieveInfo()
9	Search	searchTopRatingPlace()

		searchManually() searchByCategory()
10	Recommendation	showRoute() giveRecommendation() sentNotification()

Table 15: Method Identification

6.7 Finalizing Classes

To identify the final classes, it was required to check if there can be any hierarchies, merges, additional attributes, methods or classes. These identifications are given below:

1. There are two types of users in the system. So the user class could be the parent class of the admin class and customer class. But, as the user class and the admin class has different attributes and methods there is no need of different class.

6.8 Class Cards

Account	
Attributes	Method
User Admin Verification code Email Password	signUp() logIn() signOut() sendVerificationCode() recoverPassword() verifyPassword() verifyEmail()

	verifyFB() verifyUser() manageForgetPassword()
Responsibility	Collaborator
Need user credentials to create account Store data in Database	Admin User Database

Table 16: CRC Diagram for Account Class

User	
Attributes	Method
First name Last name User id Email Facebook Id	setUserID() getUserID() setFirstName() getFirstName() setLastName() getLastName() setEmail()

Password	getEmail() setPhoneNo() getPhoneNo() setAddress() getAddress() comment() ratePlace() searchPlace() editProfile()
Responsibility	Collaborator
User can search User can review User get recommendation User data strore into database Type of account	Search Review Recommendation Feedback Database Account

Table 17: CRC Diagram for User Class

Admin	
Attributes	Method
Email Facebook Id Password Address	getName() setName() approvesRequest() setEmail() getEmail() setPhoneNo() getPhoneNo() setAddress() getAddress() SetAdmminId() getAdminId() setPhoneNo() getPhoneNo()
Responsibility	Collaborator
Type of account Admin data store into Database	Account Database

Table 18: CRC Diagram for Admin Class

System	
Attributes	Method
Place User Account	generateNotification() showSearchedItem() sendRecommendation() generateRecommendation()
Responsibility	Collaborator
Generate recommendation Generate recommendation	Notification Recommendation

Table 19: CRC Diagram for System Class

Notification	
Attributes	Method
	sentNotification ()

Id Receiver Date	getSender() setSender() setReceiver() getReceiver() setDate() getDate()
Responsibility	Collaborator
System generate notification	System

Table 20: CRC Diagram for Notification Class

Feedback	
Attributes	Method
  Rating Comment place	setRating() getRating() setComment() writeComment()
Responsibility	Collaborator

User get feedback and write post	User

Table 21: CRC Diagram for FeedBack Class

Review	
Attributes	Method
Edit Post	writePost() editPost() getDescription() setDescription()
Responsibility	Collaborator
User give review and rating	User

Table 22: CRC Diagram for Review Class

Database	
Attributes	Method
User Id DB name	insertInfo() updateInfo() deleteInfo() retrieveInfo()
Responsibility	Collaborator
Database store information	 System User Admin

Table 23: CRC Diagram for Database Class

Search	
Attributes	Method
Category Place	searchTopRatingPlace() searchManually()

	searchByCategory()
Responsibility	Collaborator
User can search System maintain Search	User System

Table 24: CRC Diagram for Search Class

Recommendation	
Attributes	Method
Route Notification Recommendation	showRoute() giveRecommendation() sentNotification()
Responsibility	Collaborator
Sent Notification System manages Recommendation	System Notification

Table 25: CRC Diagram for Recommendation Class

6.9 UML Diagram

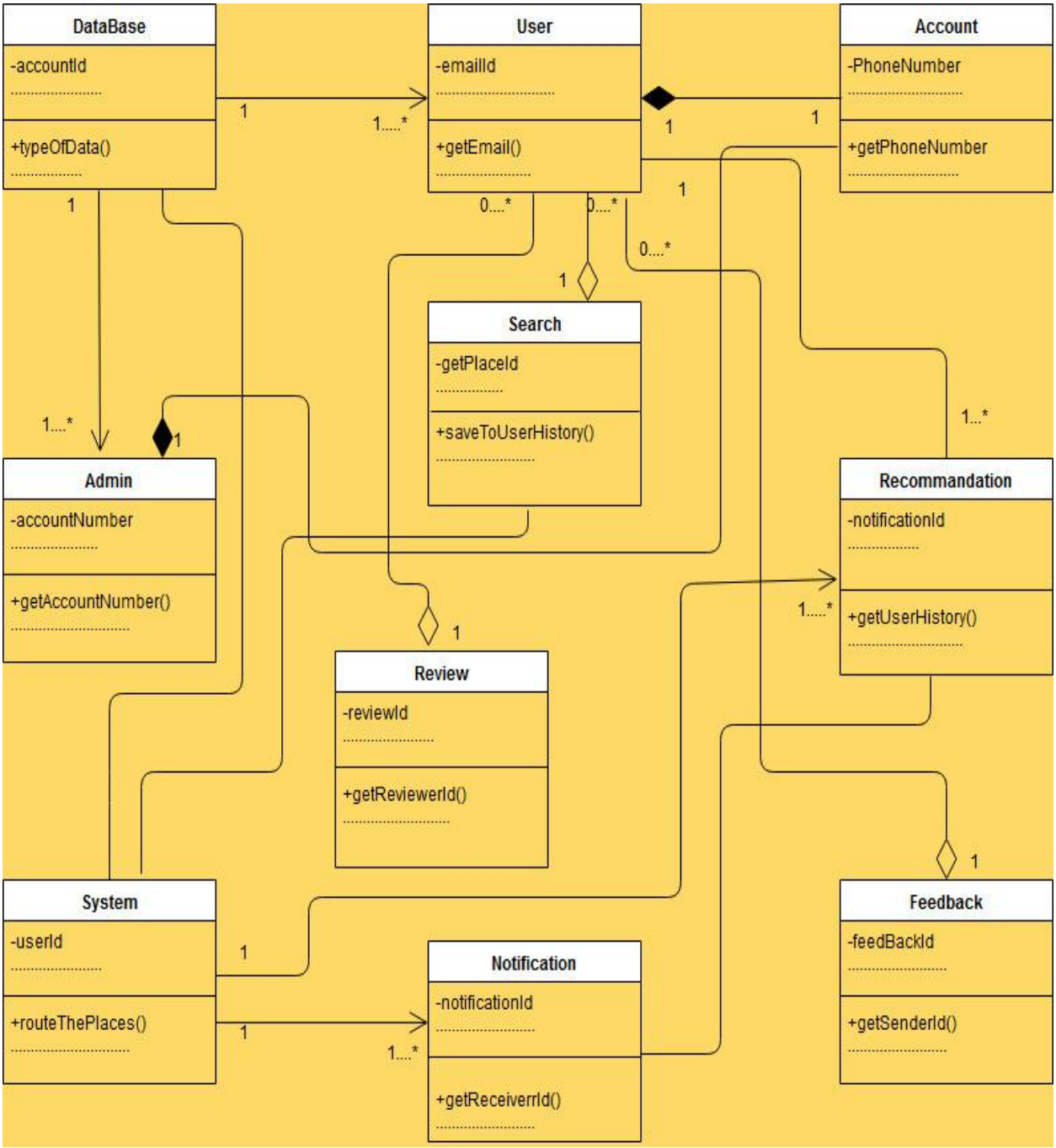


Fig 38: Uml Diagram

Chapter 7: Flow-Oriented Model

This chapter focuses on the flow oriented modeling.

7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today. It provides additional insight into system requirements and flow.

7.2 Data Flow Diagram (DFD)

The DFD takes an input-process-output view of a system. In the figures, data objects are represented by labeled arrows and transformations are represented by circles.

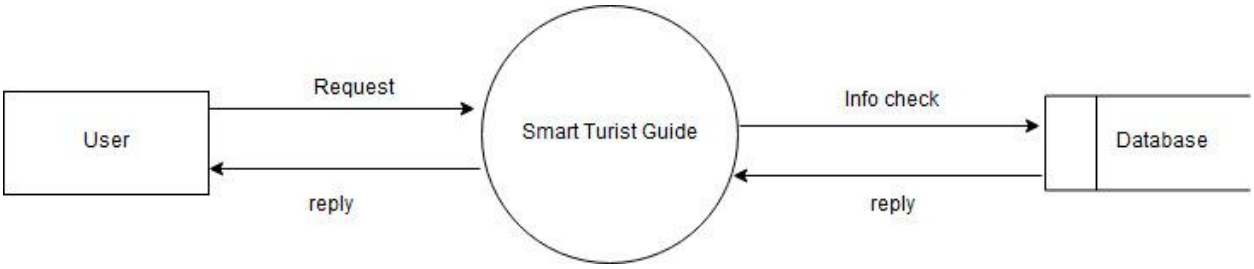


Fig 39: level -0 DFD

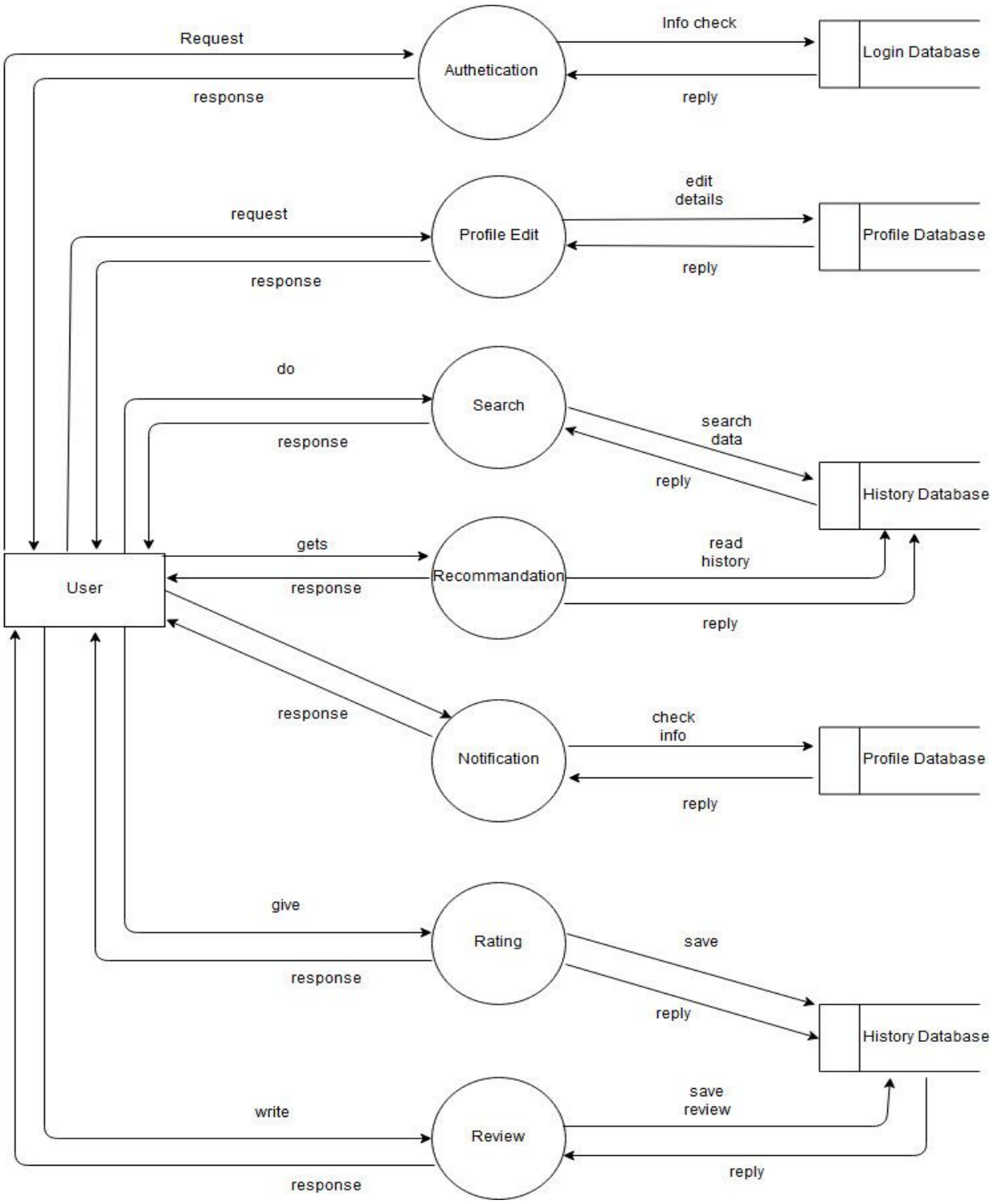
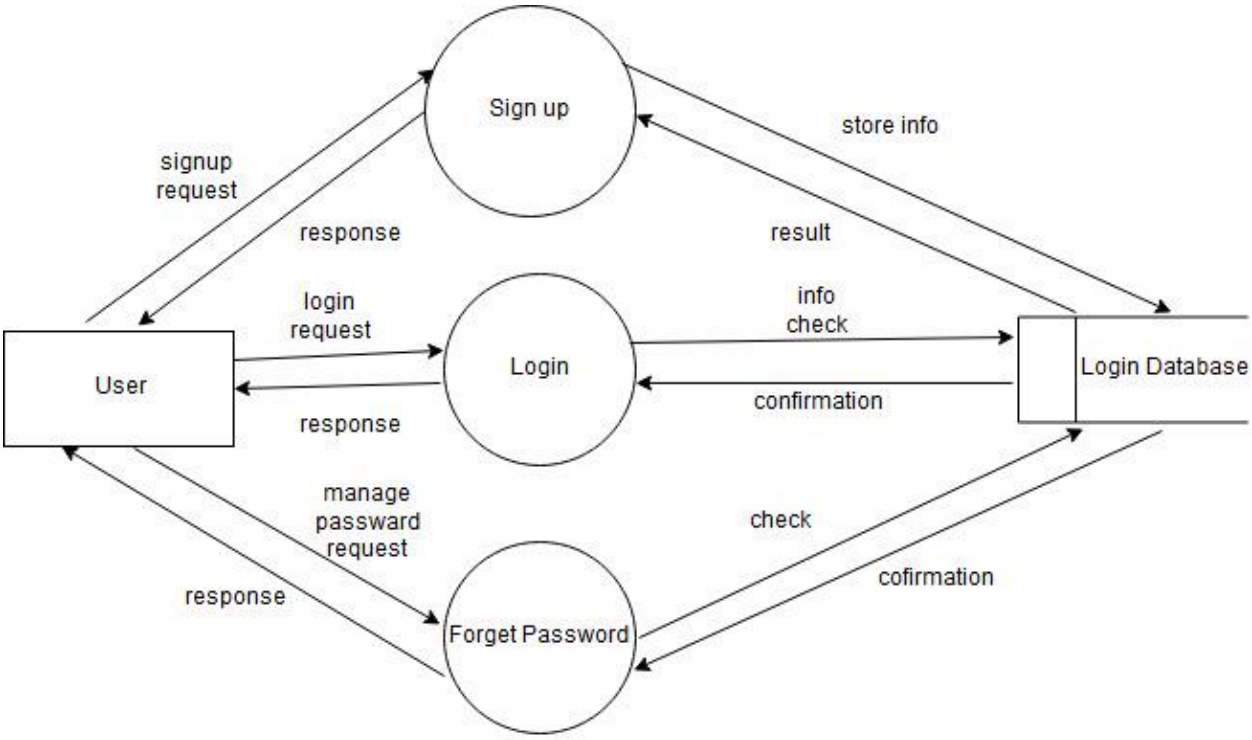


Fig 40: level -1 DFD



Text

Fig 41: level 2.1-1 DFD

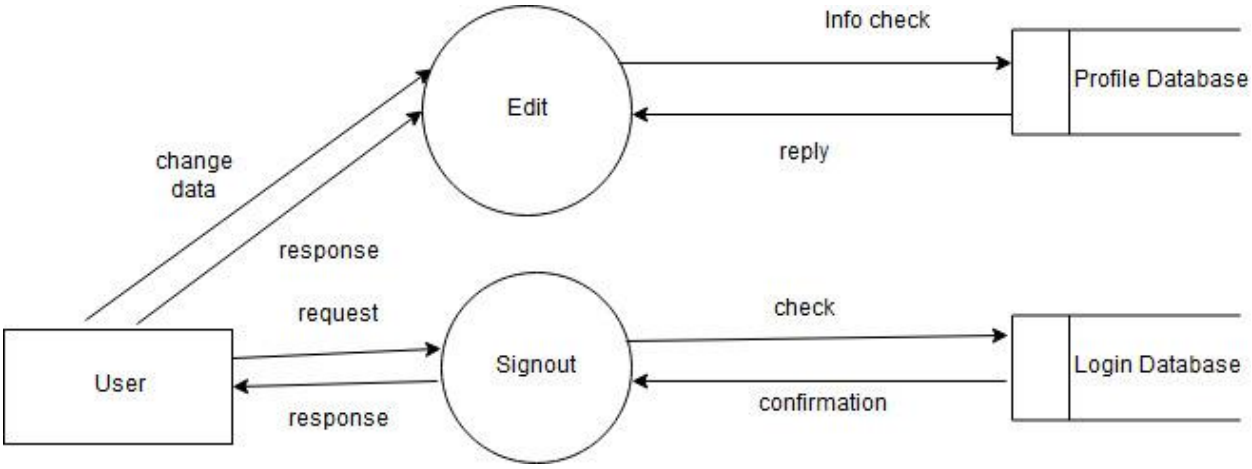


Fig 42: level 2.2-1 DFD

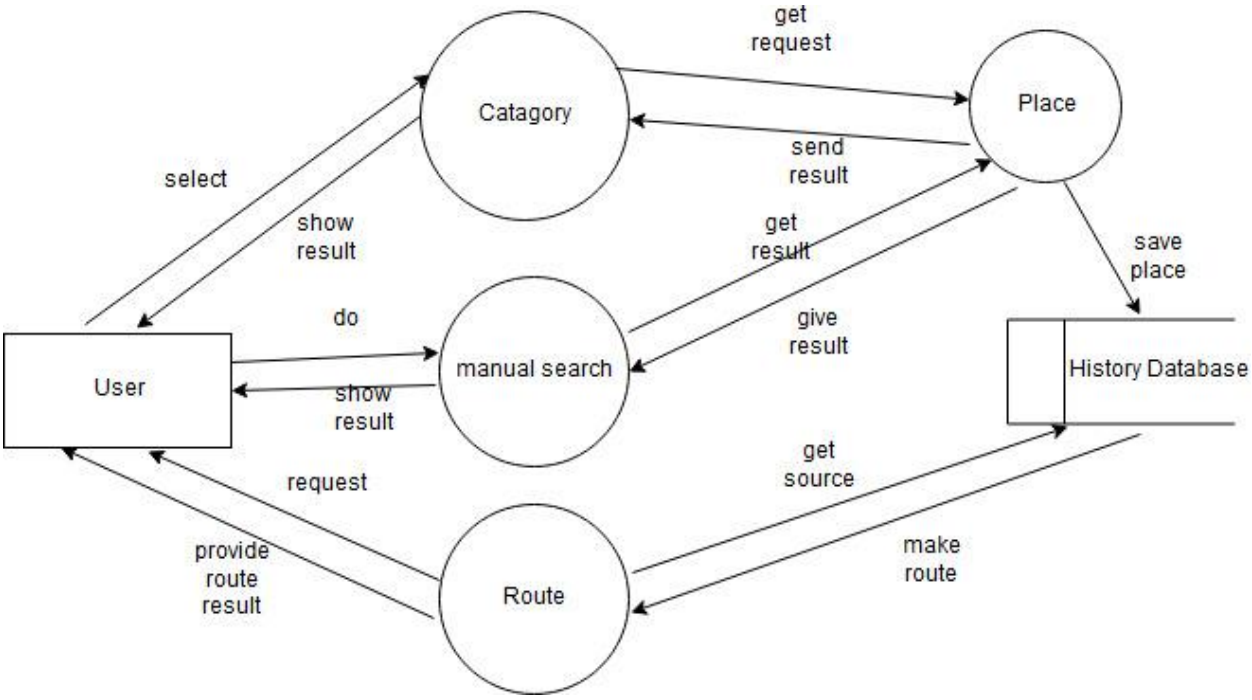


Fig 43: level 2.3-1 DFD

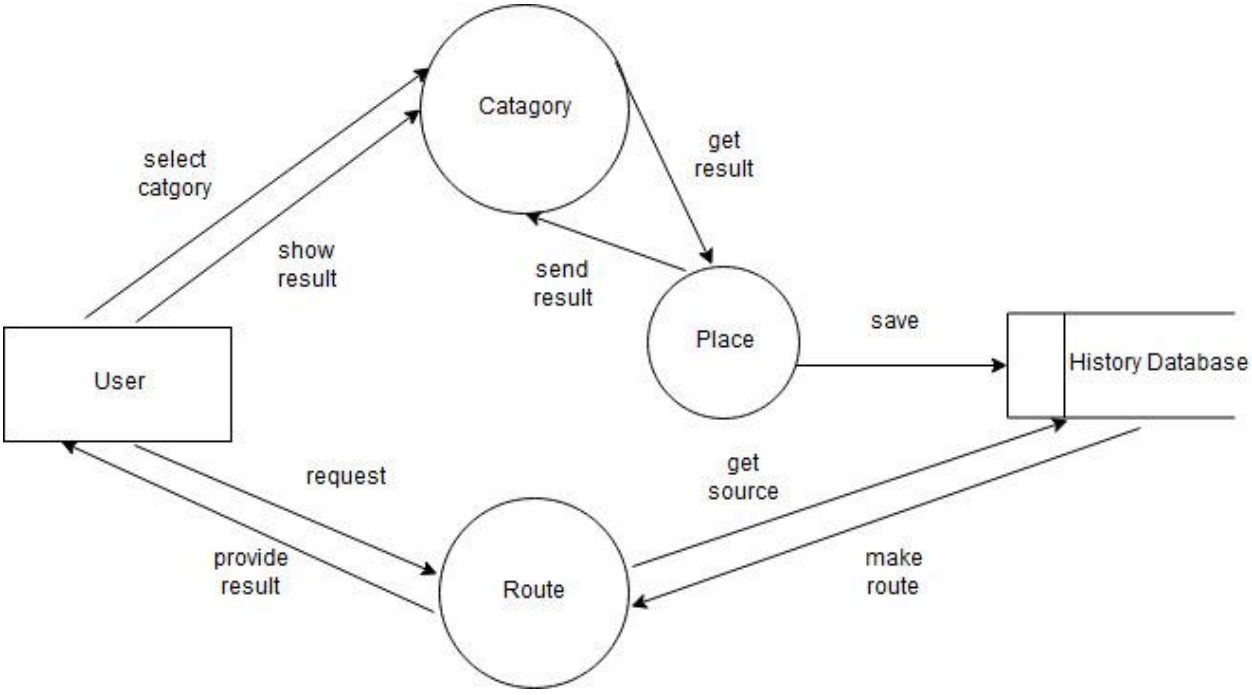


Fig 44: level 2.4-1 DFD

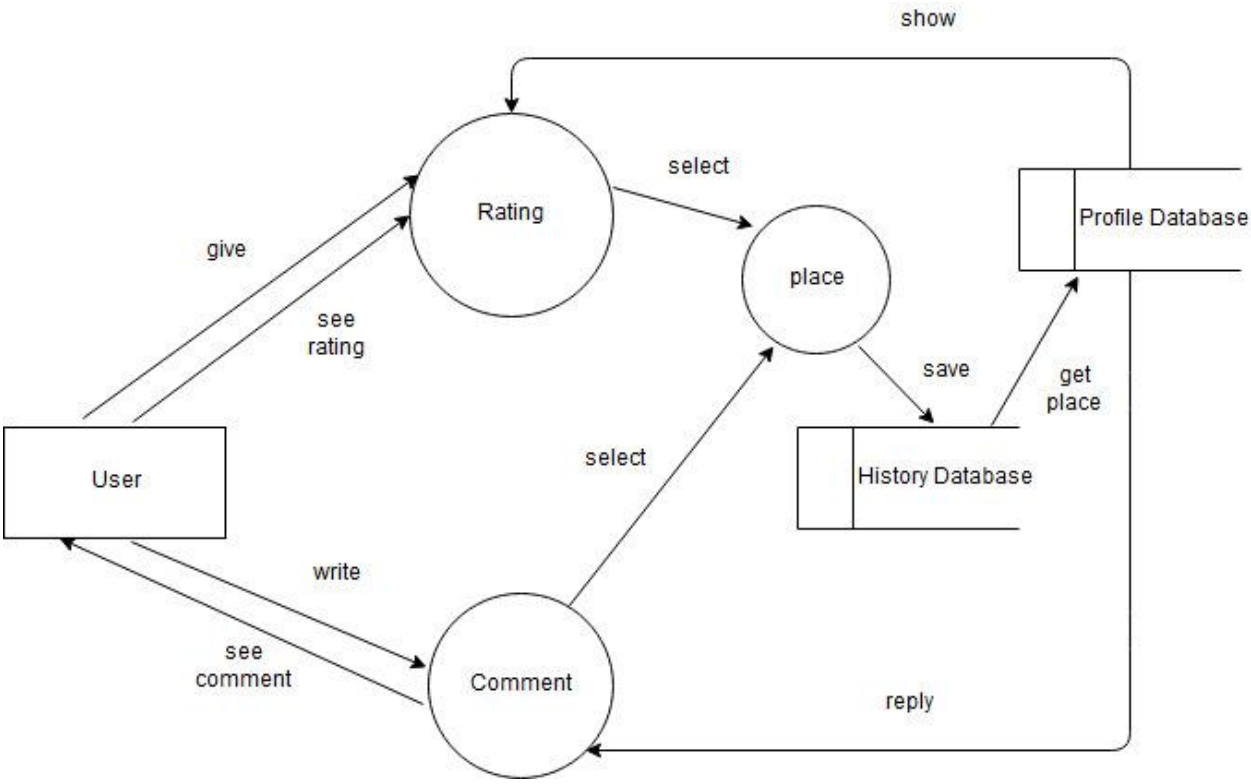


Fig 45: level 2.5-1 DFD

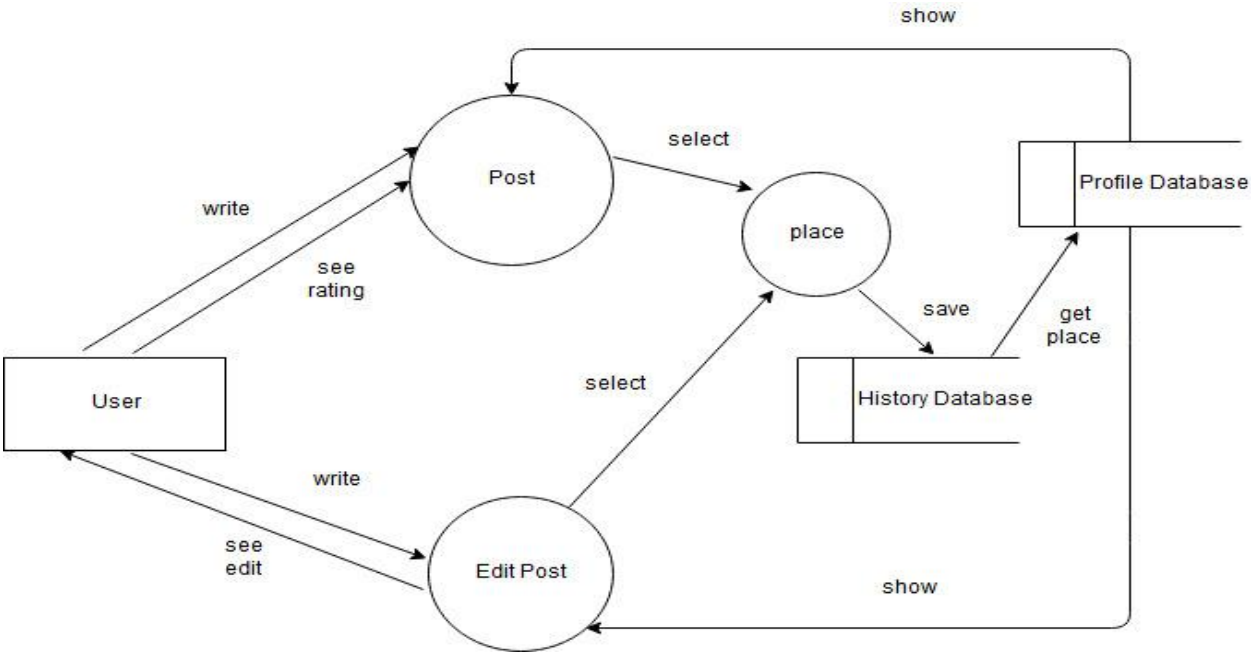


Fig 46: level 2.6-1 DFD

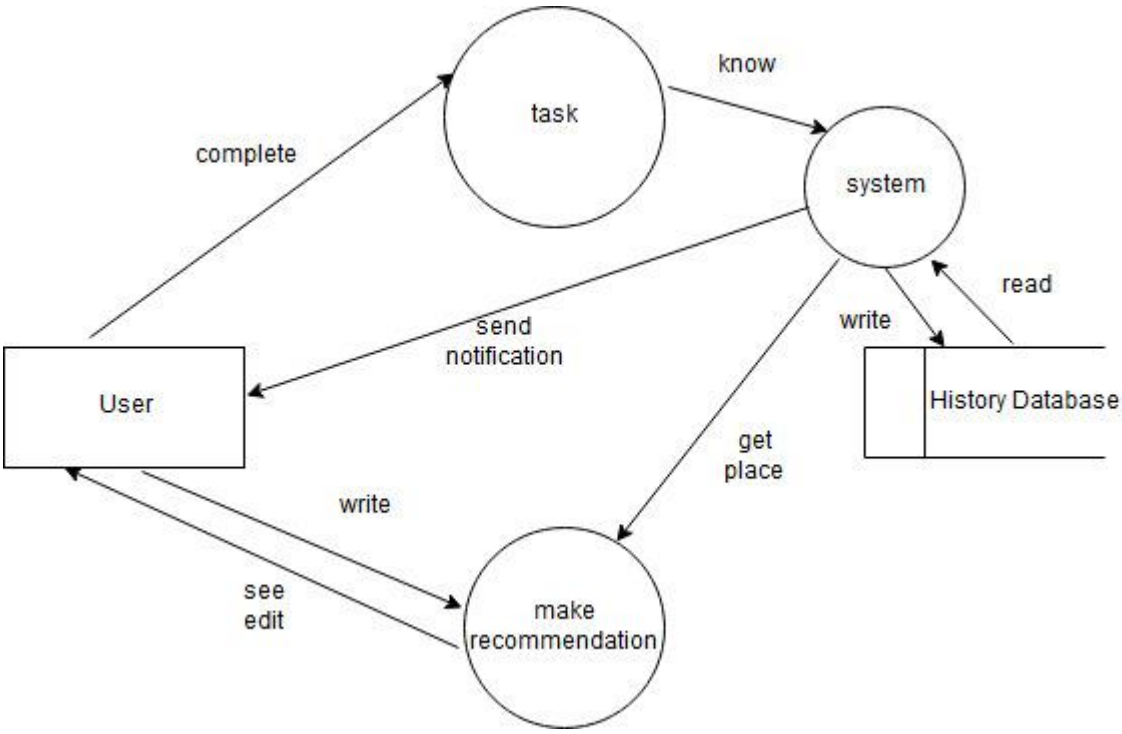


Fig 47: level 2.7-1 DFD

Chapter 8: Behavioral Model

The behavioral model indicates how software will respond to external events.

8.1 State Diagram

State diagram represents active states for each class the events (triggers).

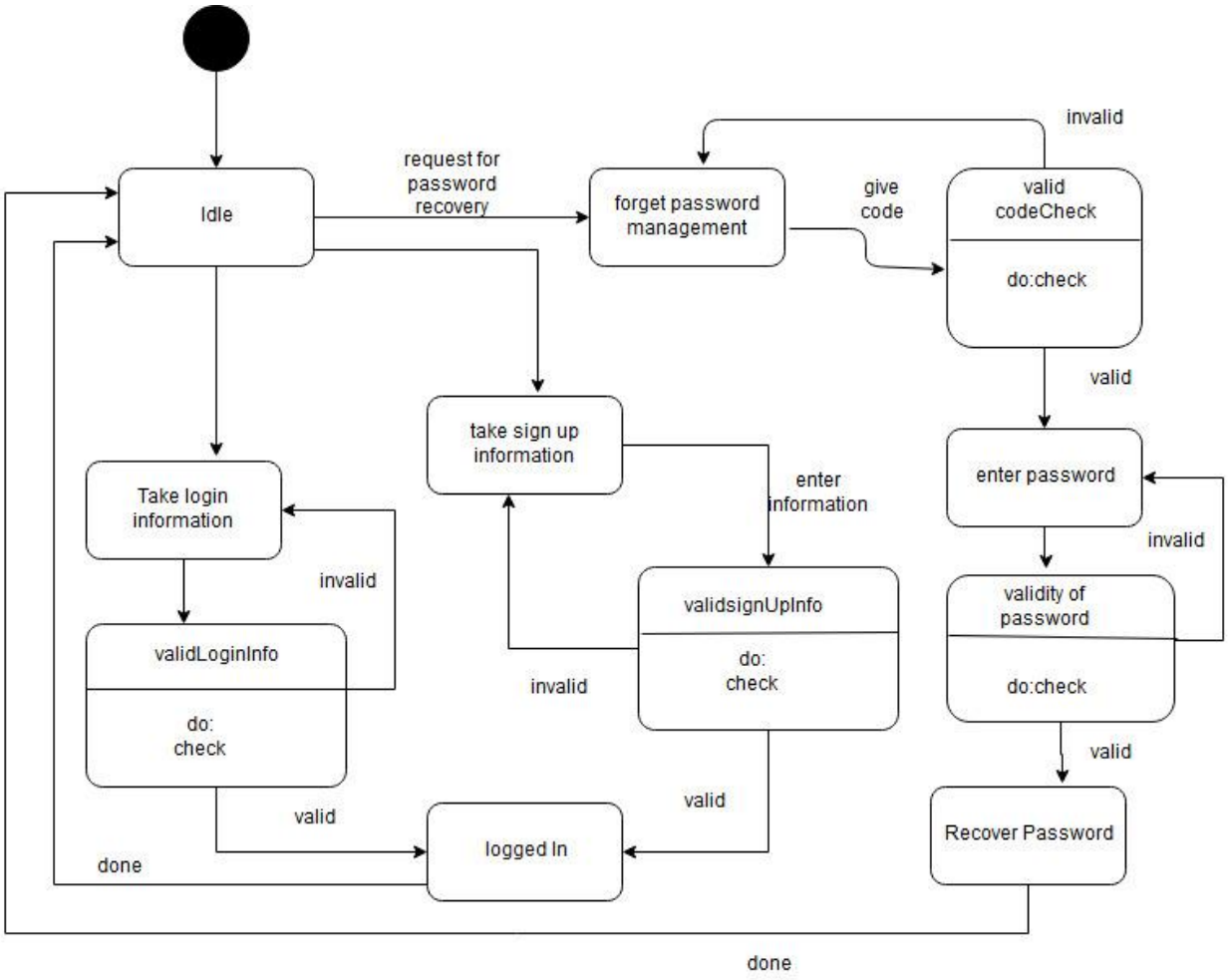


Fig 48: : State Transition for user Class

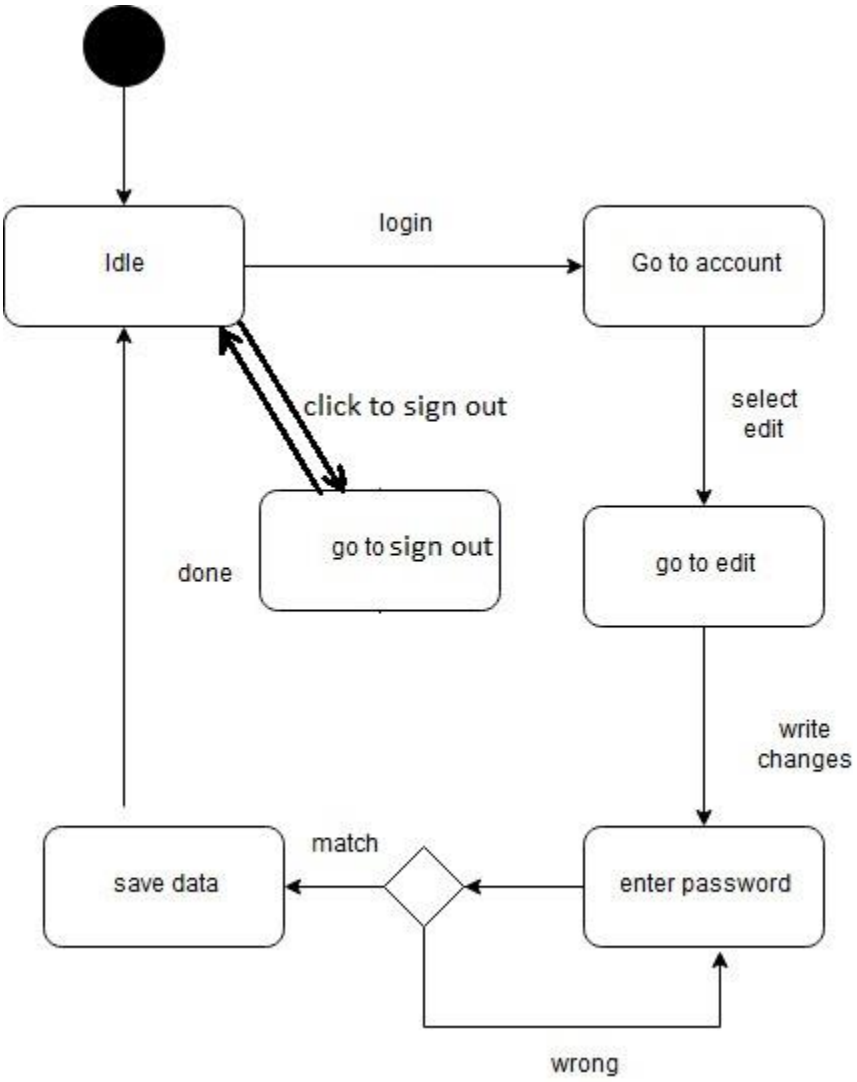


Fig 49: : State Transition for accountClass

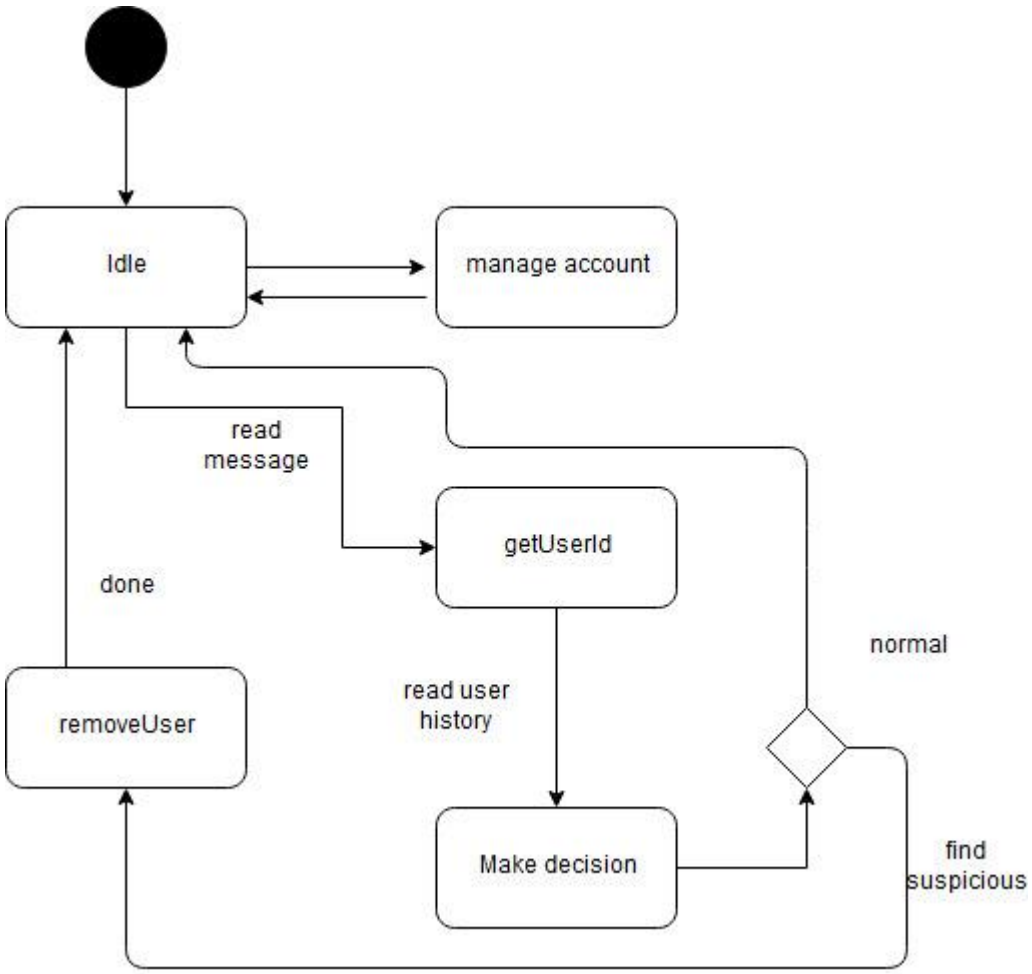


Fig 50: State Transition for adminClass

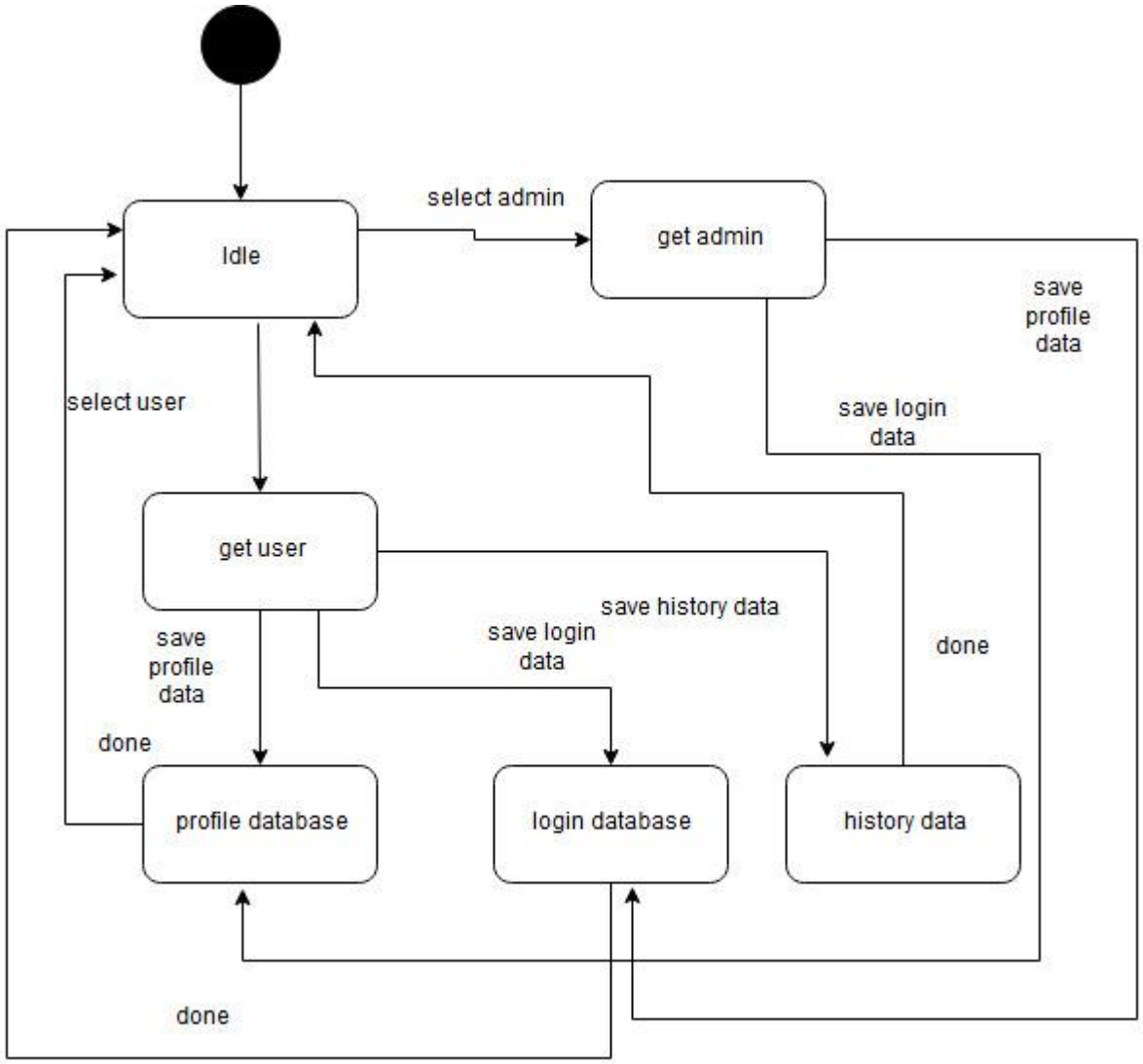


Fig 51: State Transition for databaseClass

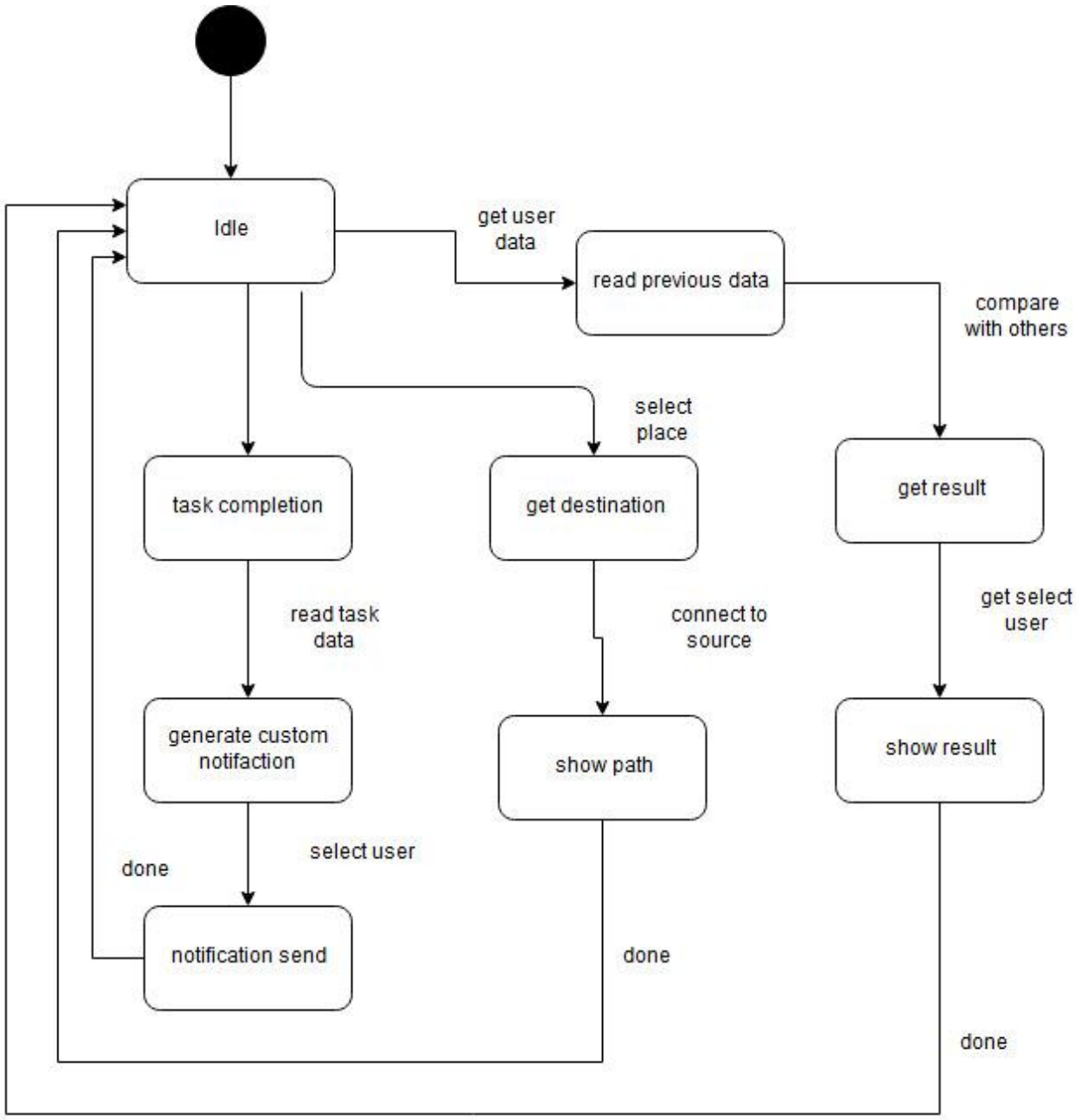


Fig 52: State Transition for systemClass

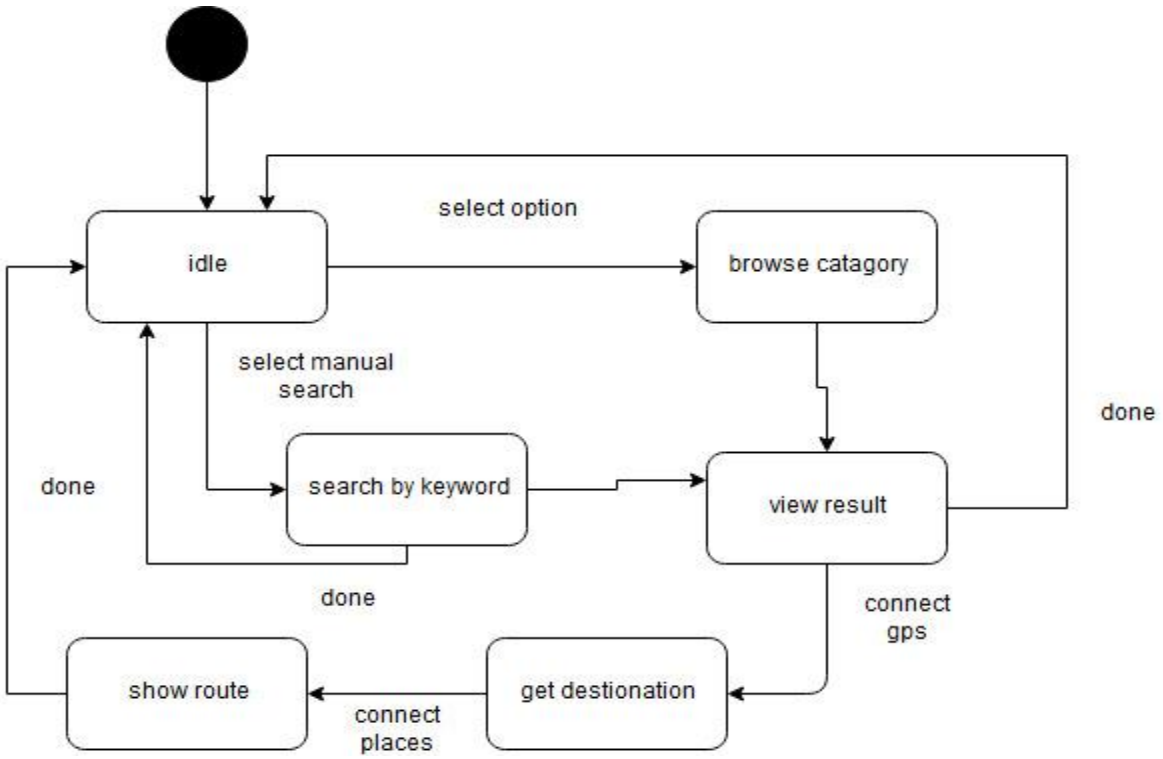


Fig 53: State Transition for searchClass

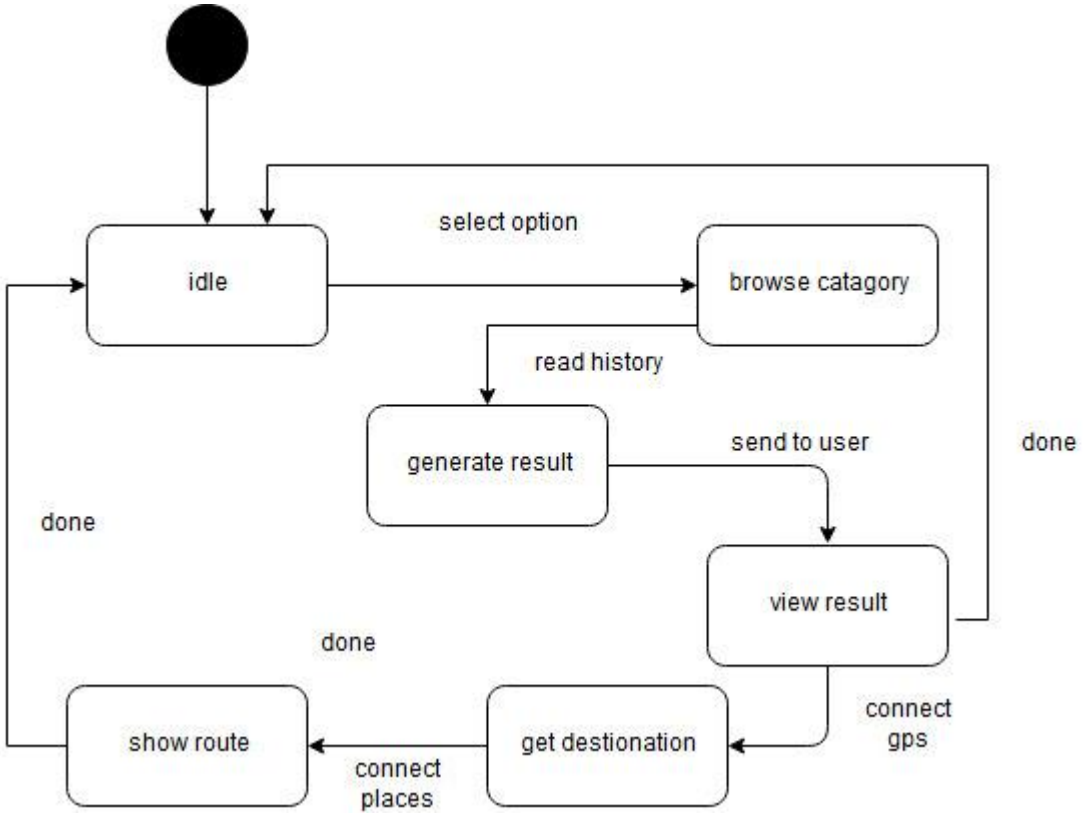


Fig 54: State Transition for recommendClass

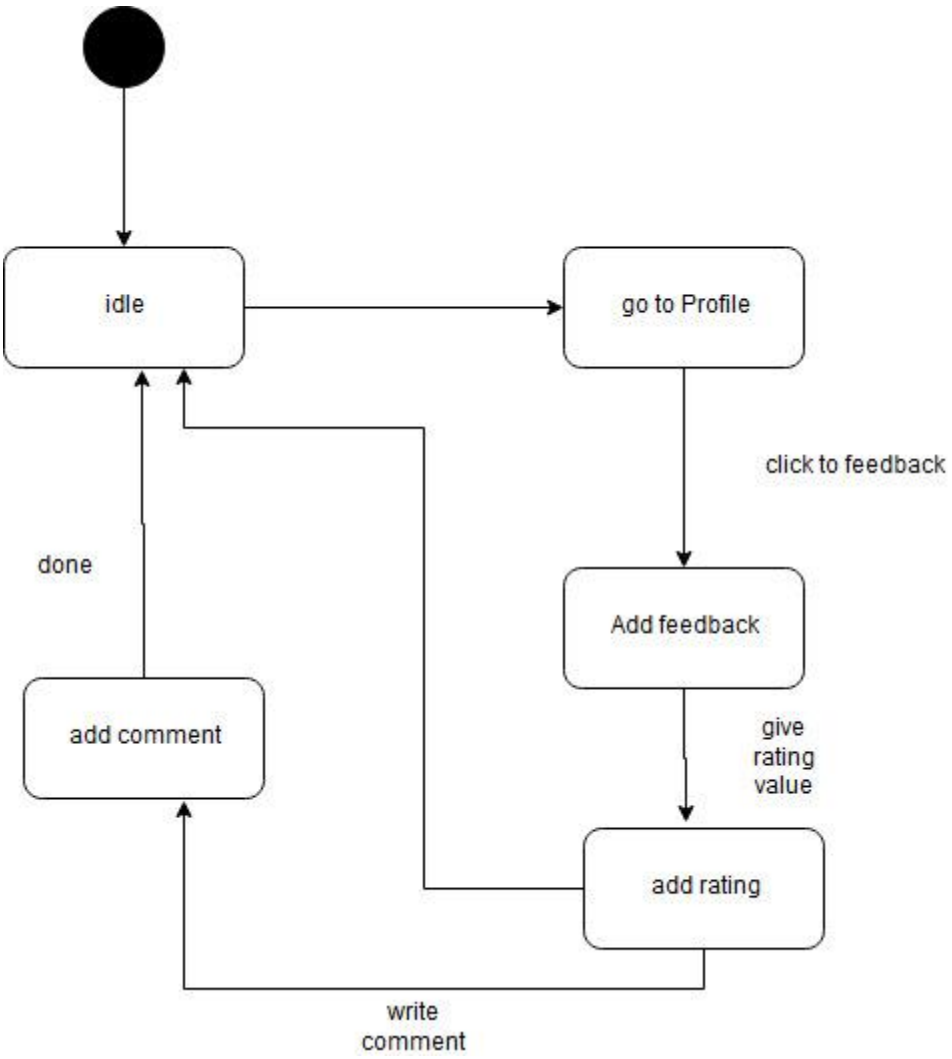


Fig 55: State Transition for feedbackClass

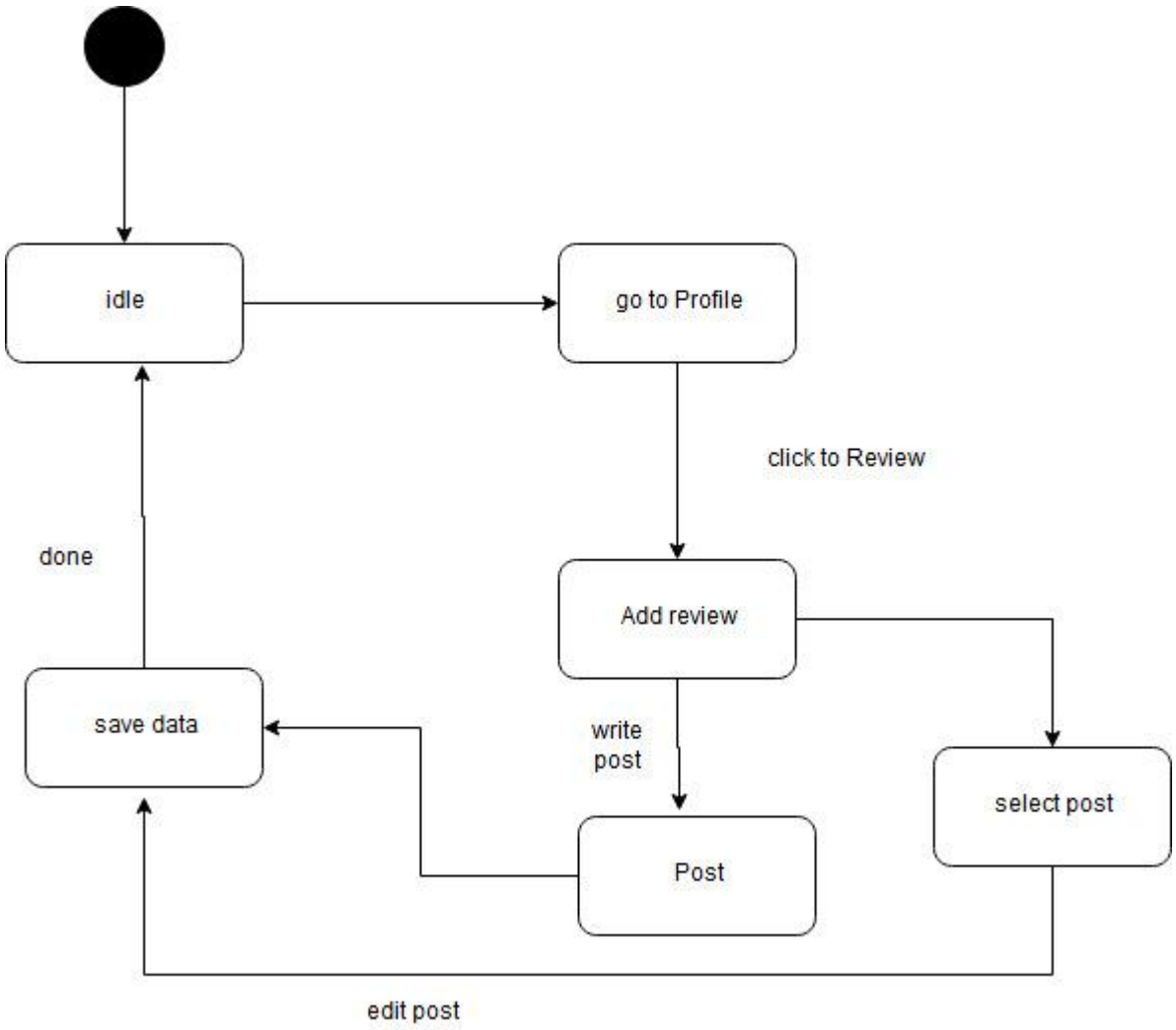


Fig 56: State Transition for reviewClass

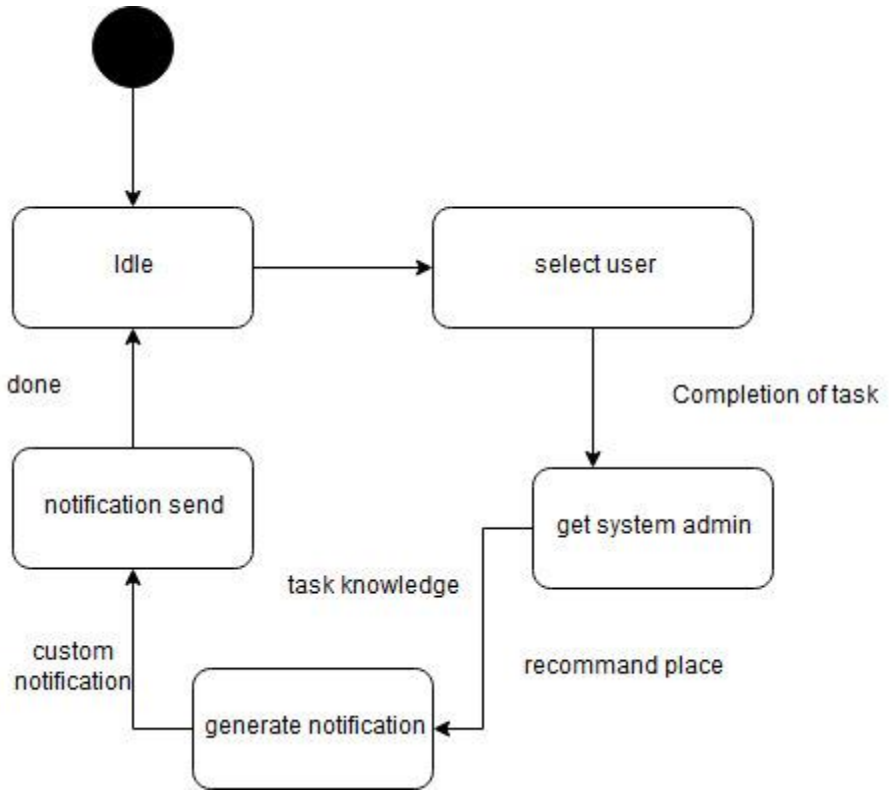


Fig 57: State Transition for notificationClass

8.2 Sequence Diagram

Sequence diagram indicates how events cause transitions from object to object

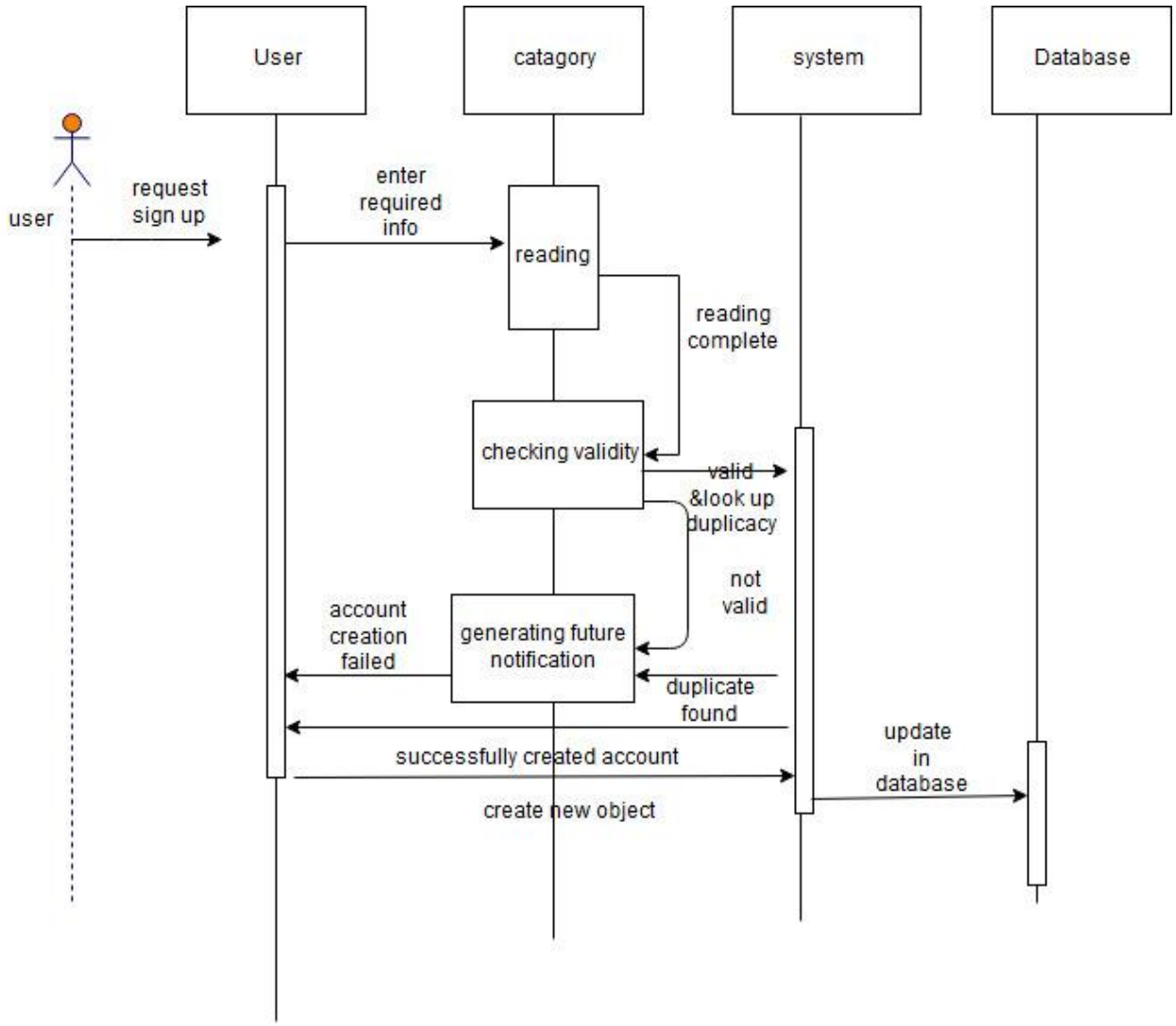


Fig 58: Sequence Diagram for user sign up

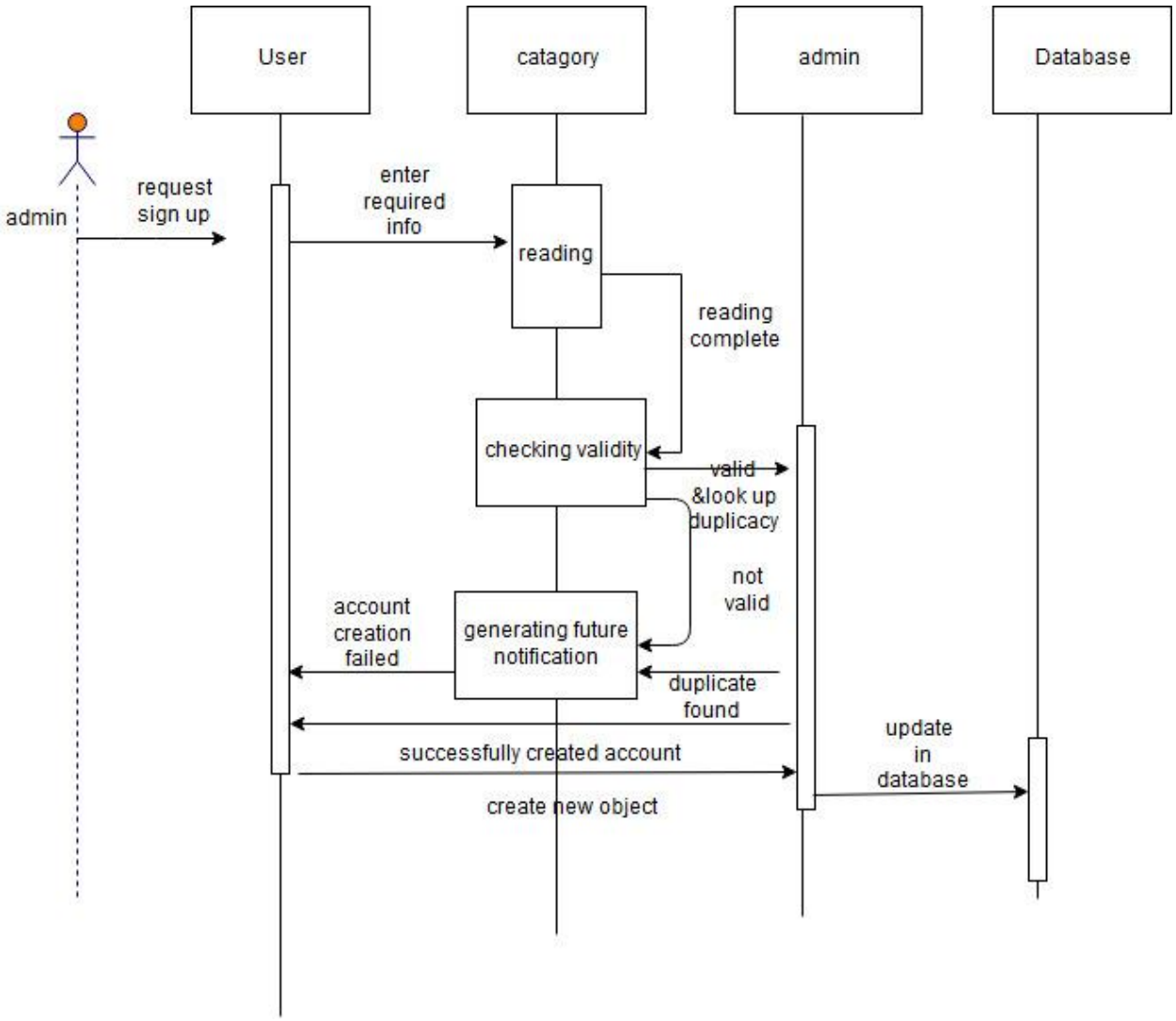


Fig 59: Sequence Diagram for admin sign up

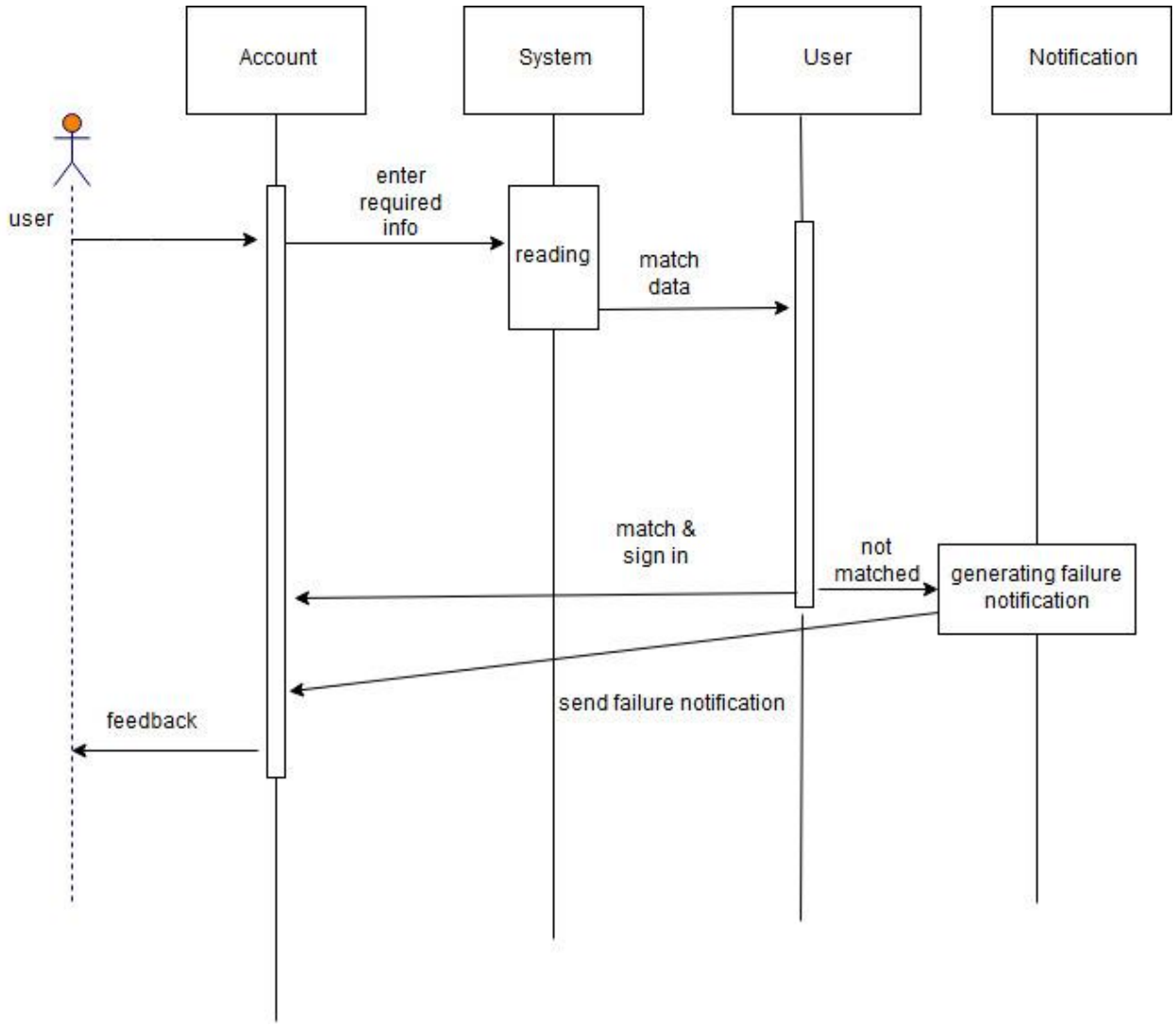


Fig 60: Sequence Diagram for user login

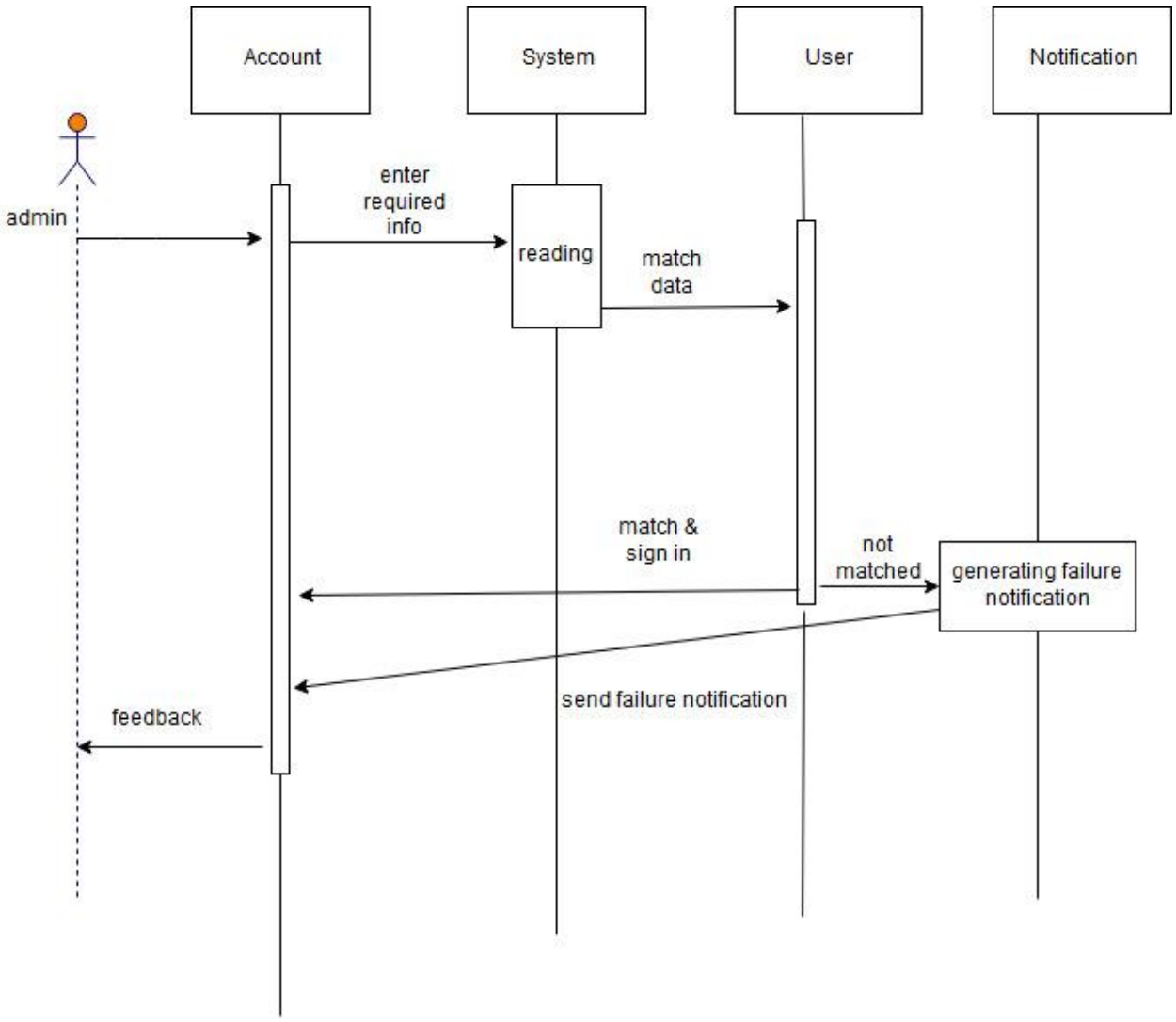


Fig 61: Sequence Diagram for admin login

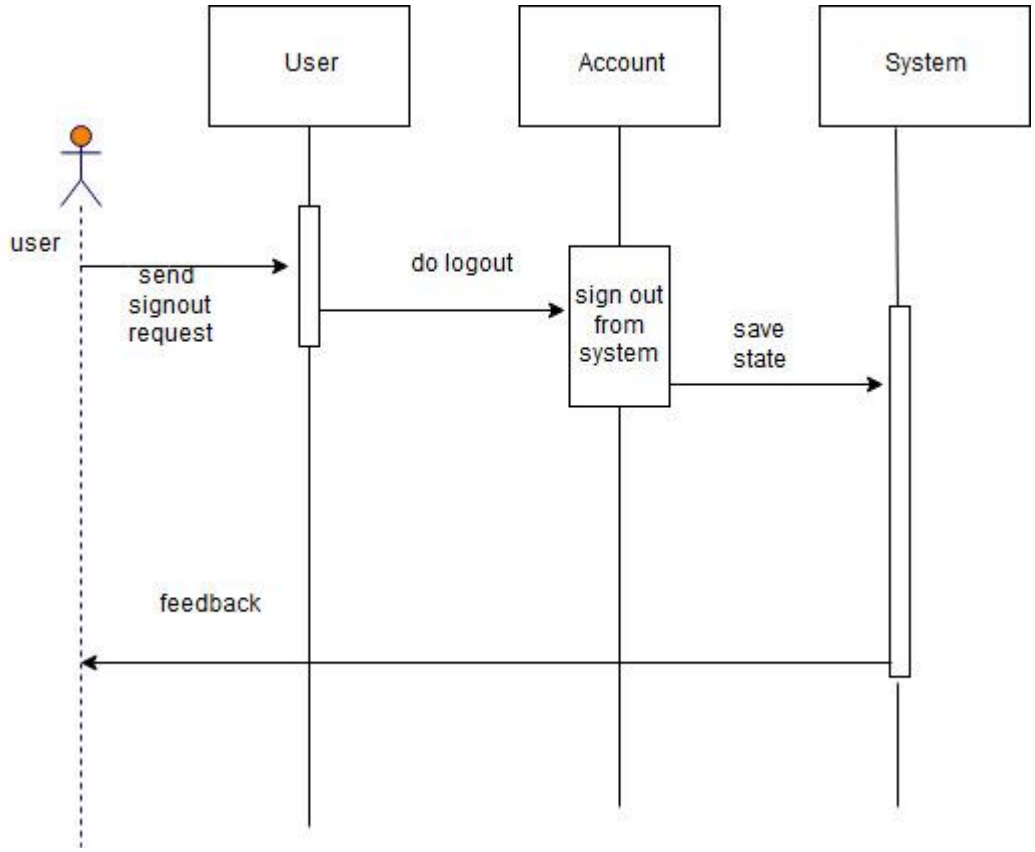


Fig 62: Sequence Diagram for user sign out

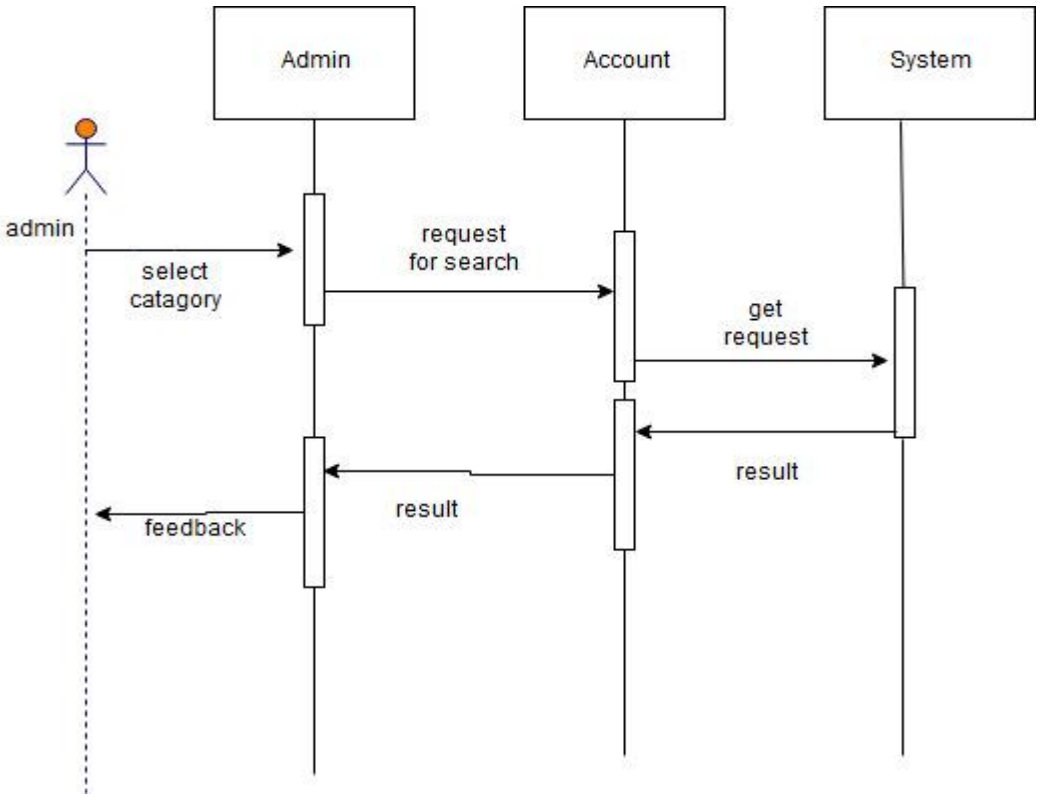


Fig 63: Sequence Diagram for admin sign out

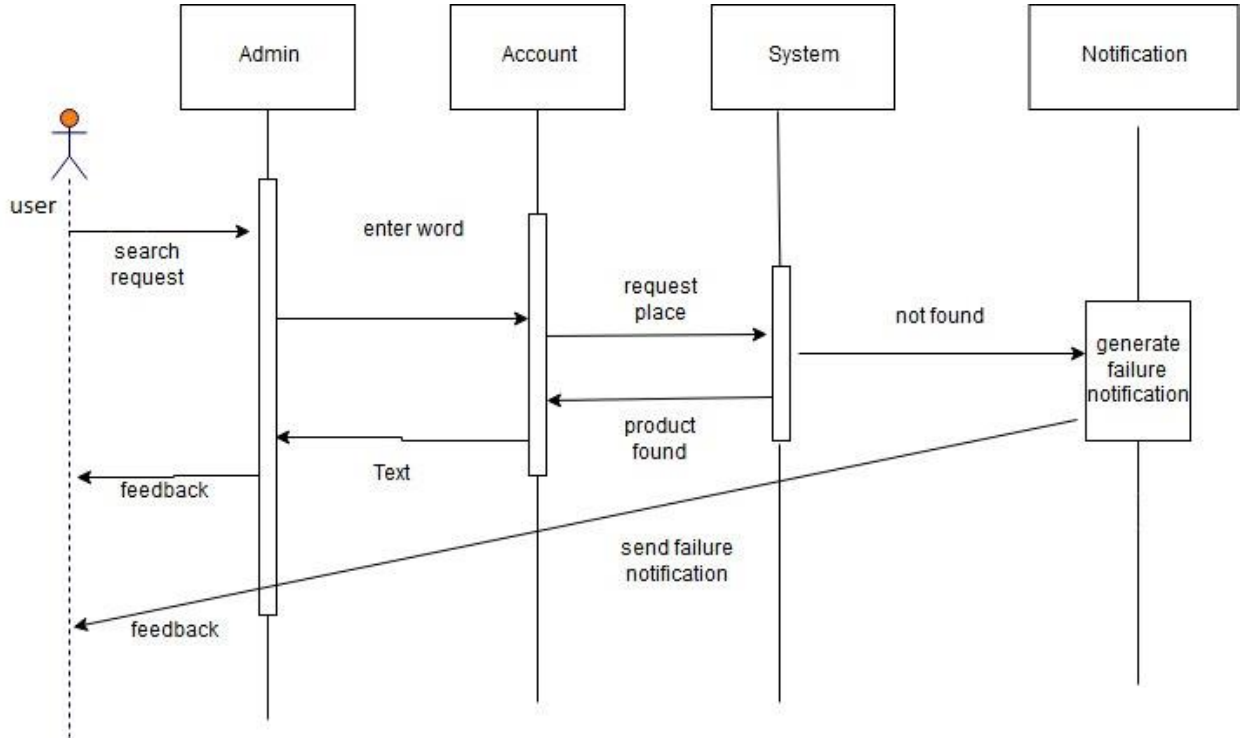


Fig 64: Sequence Diagram for manual Search

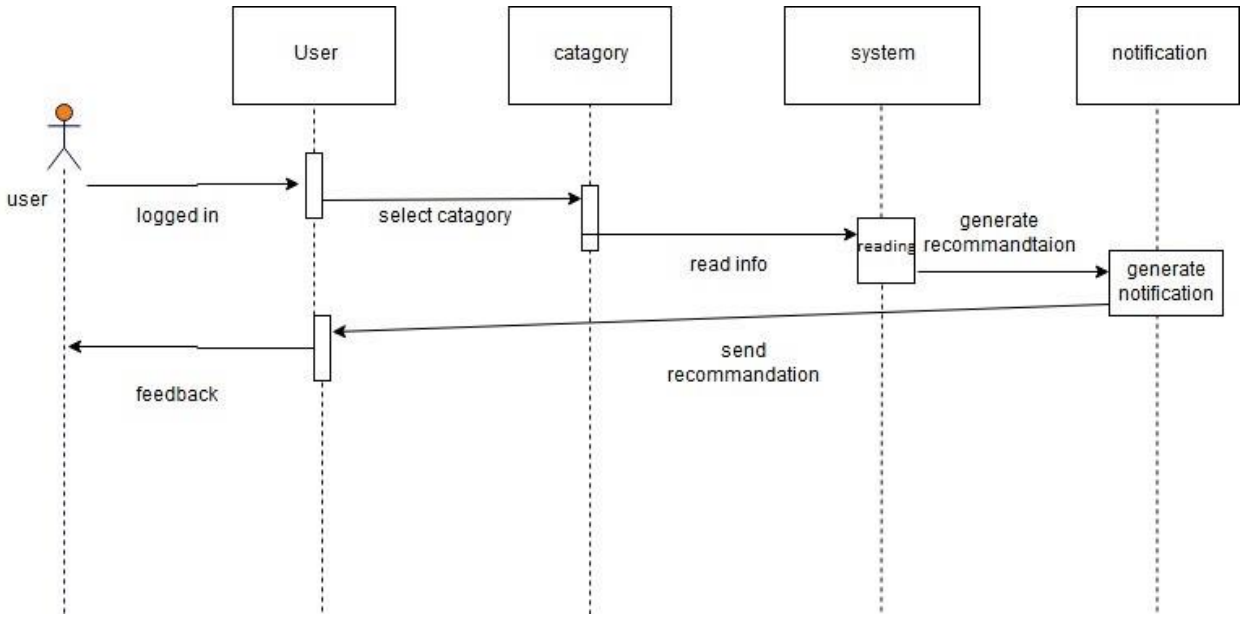


Fig 65: Sequence Diagram for category Search

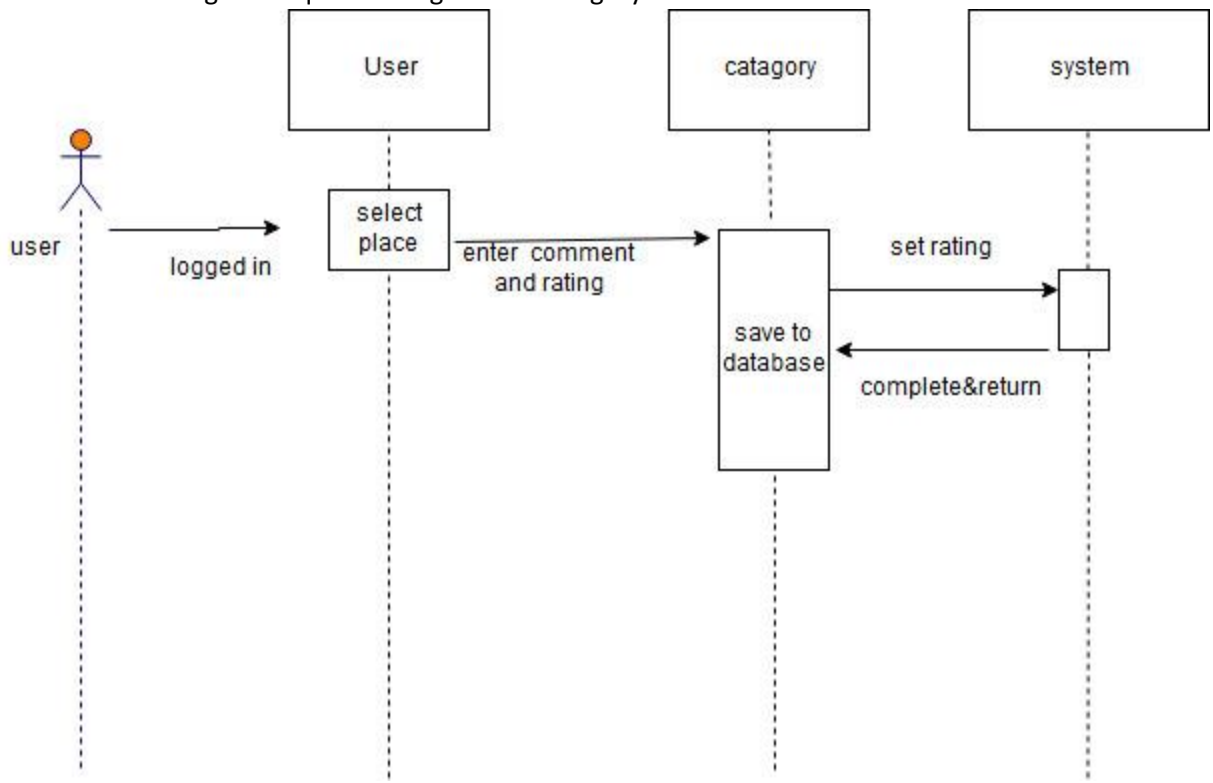


Fig 66: Sequence Diagram for recommendation

Chapter 9: Conclusion

From this SRS report on Smart Tourist Guide, the readers will get a clear and easy view of the overall system of management system of a tourist guide application. This SRS document can be used effectively to maintain the software development cycle. It will be very easy to conduct the whole project using SRS. We tried our level best to remove dependencies and make an effective and fully designed SRS.

Reference:

[1] Pressman, Roger S. Software Engineering: A practitioner's Approach (7th Edition)