Institute of Information Technology

University of Dhaka

SOFTWARE REQUIREMENT SPECIFICATION & ANALYSIS

Smart Customer Care Solution for ISP

Course Code: SE505 Software Project Lab

Submitted by

Mohsin Ibna Amin(BSSE1101)

Fahad Hossain (BSSE 1105)

Supervised by

Prof Mohd. Zulfiquar Hafiz

Designation: Professor
Institute of Information Technology
University of Dhaka

Table of Contents

CHAPTER 1

INTRODUCTION

This chapter is a part of our software requirement specification for the project “Smart Customer Care Solution for ISP”. In this chapter, we focus on the intended audience for this project.

1.1 Purpose

This document briefly describes the Software Requirement Analysis of “Smart Customer Care Solution for ISP”. It contains functional, non-functional, and supporting requirements and establishes a requirements baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered, and organized by topic. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. 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 customers as well as the project managers, designers, developers, and testers.

The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer. The project managers of the developer team will use this SRS to plan milestones and delivery dates and ensure that the developing team is on track during the 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. 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.

1.3 Conclusion

This analysis of the audience 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.

CHAPTER 2

INCEPTION

2.1 Introduction

Inception is the beginning phase of requirements engineering. It defines how a software project gets started and what the scope and nature of the problem to be solved is. The goal of the inception phase is to identify concurrent needs and conflicting requirements among the stakeholders of a software project. At project inception, we established a basic understanding of the effectiveness of preliminary communication and collaboration between the other stakeholders and the software team.

To establish the groundwork, the following factors have been worked on to the inception phase:

- Icebreaking
- List of Stakeholders
- Recognizing Multiple Viewpoints
- Working towards Collaboration
- Requirements Questionnaire

2.1.1 Icebreaking

Icebreaking refers to the fact to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this by talking with them in context-free languages. Their behavior, response to our questions, or willingness to make a change in their shops solely depends on this phase.

2.1.2 List of Stakeholders

Stakeholders refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end-users who interact with the system and everyone else in an organization that might be affected by its installation. At inception, a list of people who will

contribute input as requirements is 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 some

We identified the following stakeholders for our “Smart Customer Care Solution for ISP” project.

- Customer
- Sysadmin of ISP
- Customer care operator

2.1.3 Recognizing Multiple Viewpoints

Different stakeholders demand different features from the software. To satisfy the stakeholders, most of these features should be included in the software.

Customer Viewpoint:

- Easy to maintain internet package
- Easy to find packages
- Manage connection

Sysadmin Viewpoint:

- Easy to find problems
- Get all information in one place

Operator Viewpoint:

- Easy to contact customer
- Manage problems in one place

2.1.4 Working towards Collaboration

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

- Find the common and conflicting requirements.
- Categorize them.
- List the requirements based on stakeholders' priority points.
- Make final decisions about requirements.

Common viewpoints:

- Error-free effective system
- User-friendly
- Easy to maintain the software
- Strong authentication system

Conflicting viewpoints:

- Developing the project in minimum budget
- Creating a ticket generation system

Final requirements:

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

- User-friendly system
- Strong authentication system
- Maximum error-free system
- Restrict access to the functionality of the system based upon user roles.

- Easy to maintain the software

2.1.5 Requirements Questionnaire

At first some context-free questions were asked for identifying the stakeholders. Context-free questions are helpful to identifying some stakeholders who cannot be identified by structural questions. Then questions regarding the software were regarding their demands.

2.2 Conclusion

The Inception phase helped us to establish a basic understanding of the customer care 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.

In our project, we have established a basic understanding of the problem, the nature of the solution that is desired, and the effectiveness of preliminary communication and collaboration between the stakeholders and the software team. More studies and communication will help both sides (developer and client) to understand the future prospect of the project. Our team believes that the full functioning document will help us to define that future prospect.

CHAPTER 3

ELICITATION

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 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-

- Collaborative Requirements Gathering
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

We have met with some stakeholders in the Inception phase such as the customer, sysadmin, customer care operator. These meetings created an indecisive state for us to elicit the requirements. To solve this problem, we have met with the stakeholders (who are playing a vital role in the whole process) again to elicit the requirements.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a quality management technique that translates the needs of the clients into technical requirements for the software. The prime concern of the QFD is customer satisfaction maximization. In order to ensure this, QFD enforces an understanding of what customers describe as ‘valuable’ and then deploys these values throughout the engineering process.

QFD defines three types of requirements:

- Normal Requirements
- Expected Requirements
- Exciting Requirements

3.2.2.1 Normal Requirements

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

1. Simple and user-friendly interface
2. Easily accessible for all
3. Web-based application
4. Login system
5. Allow new members to register
6. Notify members via SMS
7. Search Internet packages
8. Subscribe to package
9. Change internet plan
10. Rate packages
11. Get news and info from ISP
12. Sysadmin view all problems in one place

13. Customer care operator notifies all customers
14. Customers can file a complaint

3.2.2.2 Expected Requirements

The requirements that are implicit in the system might not be brought up during the meeting because of their fundamental nature. Despite not being explicitly mentioned, their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below:

1. Database
2. Login
3. Error-free software
4. User friendly
5. Data backup
6. Interactive and attractive graphical user interface
7. Strong authentication process

3.2.2.3 Exciting Requirements

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

1. Direct message to customer care operator
2. In apps voice call
3. Customer can view their request status

3.2.3 Usage Scenario

1. **Registration:**
    a. **Sysadmin account creation:** To use the system sysadmin has to create an account. The sysadmin will provide a username, mobile number, and job id. A verification code will be sent to the mobile number. If a user can input the correct code an account will be created.
    b. **Customer care operator account creation:** The customer care operator follows the same procedure as a sysadmin to create an account.
    c. **Customer account creation:** To use the system, the customer has to create an account. The customer will provide a username, mobile number, and address where the connection will be given. If all details are correct a verification code will be sent to the mobile number. If a user can input the correct code an account will be created.
2. **Profile**
    a. **Sysadmin profile:** Every account has a profile. In the profile, a user can add a profile picture. The profile will show username, profile picture, mobile number. A sysadmin can update the profile information.
    b. **Customer care operator profile:** The customer care operator's profile will show username, profile picture, and mobile number. Operators can update their profile information.
    c. **Customer profile:** Customers' profiles will show username, profile picture, mobile number, and address.
3. **Update Internet Packages:** Sysadmin will update internet packages. Old packages can be blocked or updated. New internet packages can be added also. The sysadmin will manage existing packages.
4. **Searching and Viewing:** Customer and guest users can search and view existing internet packages.

5. **Select Package:** Valid customers can select packages from the package list. A customer can select a package at a time. The selected package will be subscribed by the customer.
6. **File Complaint:** A customer can file a complaint. Complaint status can be viewed by the customer.
7. **Chat Box:** Customers can chat with customer care operators. In apps voice call is possible if needed. Customers cannot message sysadmin but customer care operators can.
8. **Block request:** Customers can request to block a specific webpage or web URL. A list of blocked websites will be displayed.
9. **Select Problem:** Sysadmin selects problems to solve. Two or more sysadmin cannot select the same problem to solve.

3.2.4 Elicitation Work Product

At first, it has to be known whether the output of the Elicitation task may vary because of the dependency on the size of the system or the product to be built. Here, the Elicitation work product includes:

- Making a bounded statement of scope for our system.
- Making a list of customers, users, and other stakeholders who participated in the requirements elicitation.
- Make a list of requirements that are organized by function and domain constraints that apply to each other.
- A set of usage scenarios that provide insight into the use of the system.
- Description of the system's technical environment.

CHAPTER 4

SCENARIO BASED MODELING

4.1 Introduction

When developing software, user satisfaction is given the highest priority. If we understand how end-users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Thus, requirements are with scenario generation in the form of use cases, activity diagrams, and swim lane diagrams.

4.2 Definition of use case

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end-user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users.

The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as system operators. Every user has one or more goals when using the system.

Primary Actor:
Primary actors interact directly to achieve the required system function and derive the intended benefit from the system. They work directly and frequently with the software.

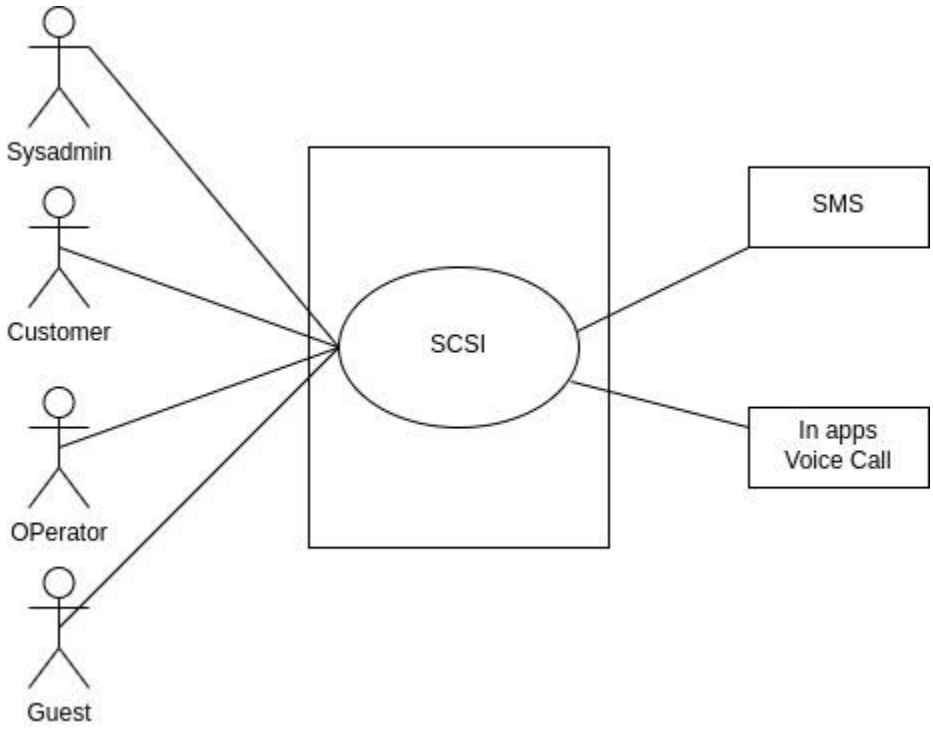
Secondary Actor:
Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

4.3 Use Case Diagrams

Use case diagrams to give a non-technical view of the overall system. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

4.3.1-Level 0

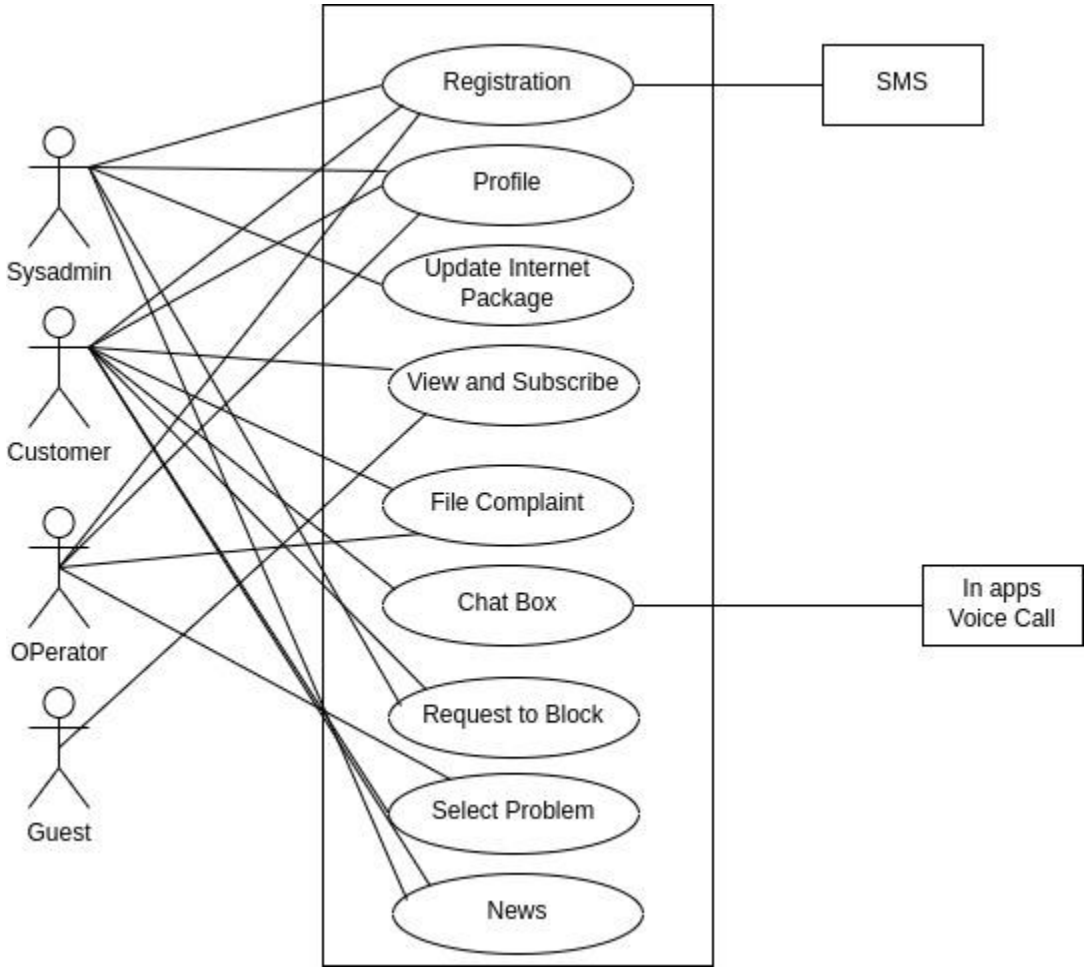
Name: SCSI (Smart Customer Care Solution for ISP)
Primary Actor: Sysadmin, Customer, Operator.
Secondary Actor: Guest



Description:
This is the overall view of SCSI. Sysadmin,operator, customer and guest all are primary actors.They will initiate a task for the system. SMS are external entities. The system will provide inbuilt message and call service.

4.3.2-Level 1

Name: SCSI (Smart Customer Care Solution for ISP)
Primary Actor: Sysadmin, Customer, Operator.
Secondary Actor: Guest

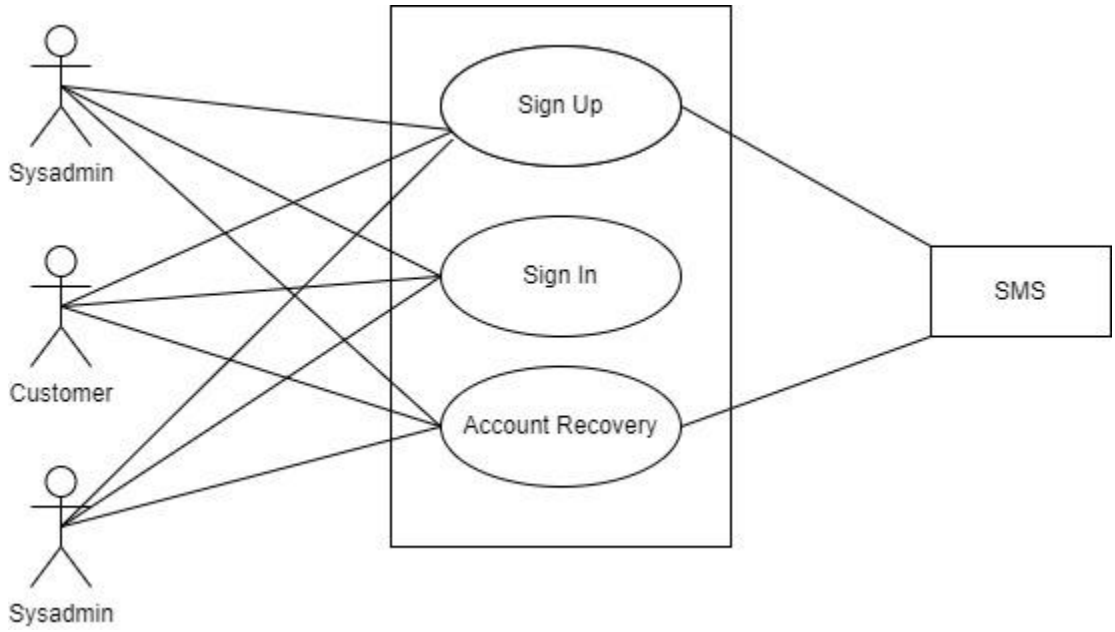


Description:
Registration: Users must register before signing in .
Profile: Every user can manage their profile.
Update Internet Package: Sysadmin update the internet package
View and Subscribe: Customers can view different internet packages and subscribe.

File Complaint: Customers can file complaints regarding internet packages.
ChatBox: Customers and operators can communicate with the chatbox.
Request to block: Customers can request to block a website.
Select Problem: Sysadmin can view the problems and solve them.
News: Sysadmin can create news for the customers.

4.3.3-Level 1.1

Name: Account
Primary Actor: Sysadmin, Customer, Operator.

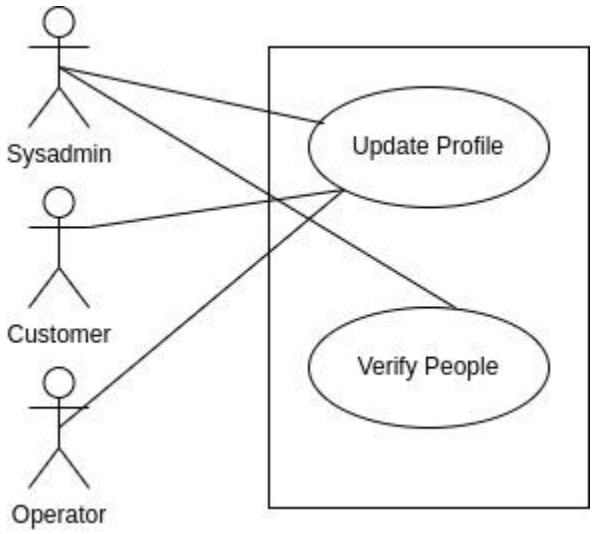


Description:

Sign Up:To create an account, users must provide their information to the system. They must choose a user type and depending on their user type they must provide additional information.
Sign In: After the account is created, users can sign in to their account.
Account Recovery: A user can recover his/her password if forgotten, by using phone number or by contacting the sysadmin manually.

4.3.4-Level 1.2

Name: Profile
Primary Actor: Sysadmin, Customer, Operator.

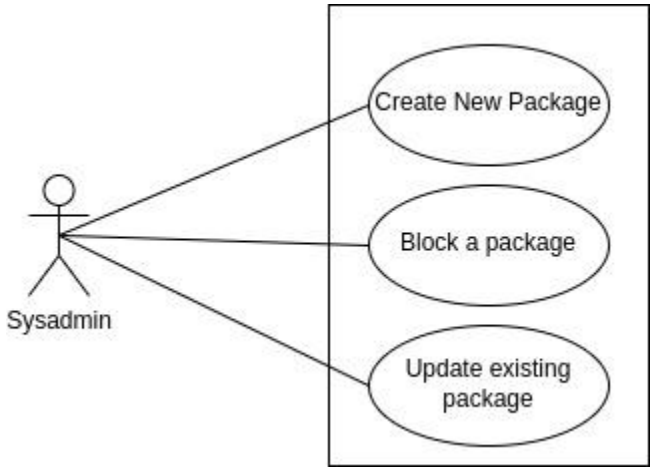


Description:

Update Profile: Customers, sysadmin and operator can update their existing profile.
Verify People: Sysadmin can verify the profile of customers and operators.

4.3.5-Level 1.3

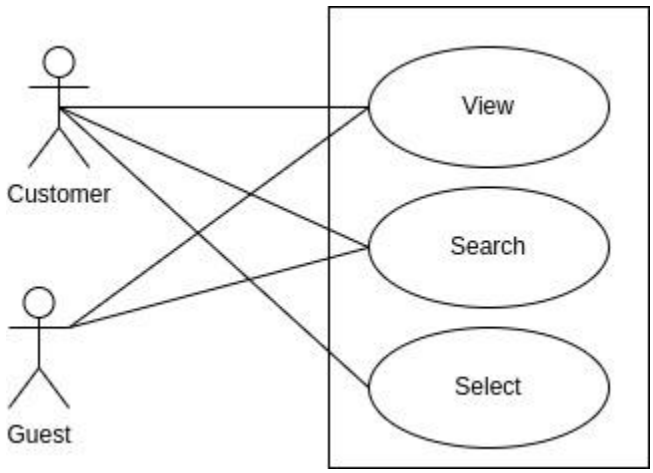
Name: Update Internet Package
Primary Actor: Sysadmin.



Description:
Create New Package: Sysadmin can create new internet packages.
Block a Package: Sysadmin can block a package.
Update existing package: Sysadmin can update packages of customers.

4.3.6-Level 1.4

Name: Select Package
Primary Actor: Customer, Guest.



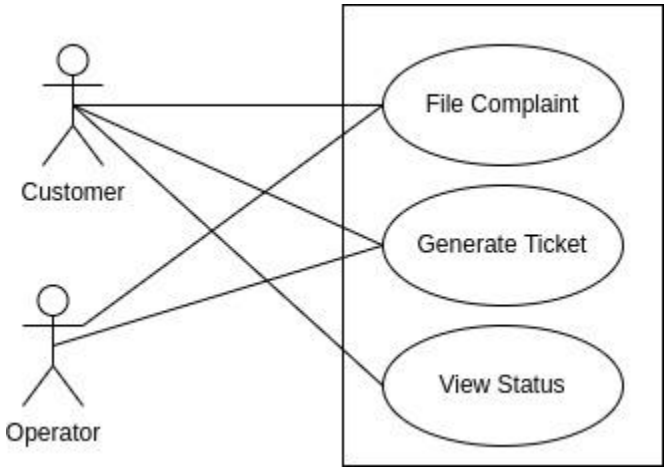
Description:

View: Customers and other viewers can view any package.

Search: Customers and other viewers can search any package.
Select: Customers can select packages to subscribe.

4.3.7-Level 1.5

Name: File Complaint
Primary Actor: Customer, Operator.

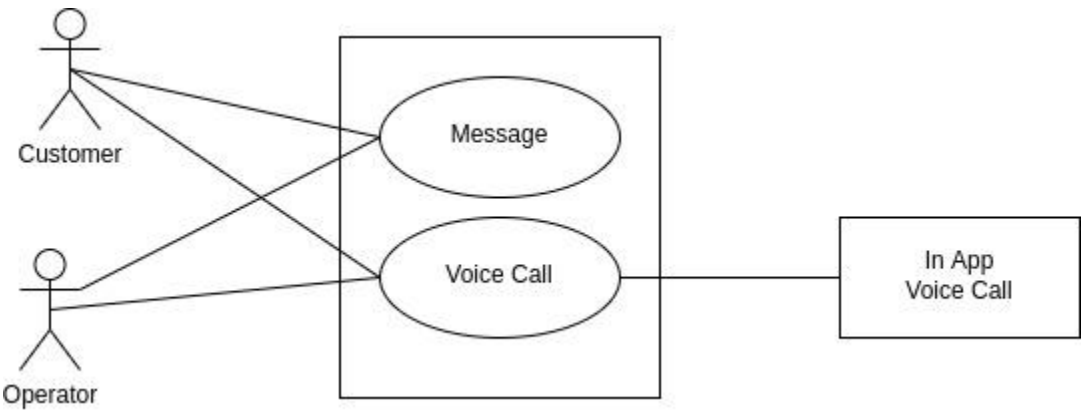


Description:

File complaint: Customers can complain about problems and the operator can also file complaints on behalf of the customers.
Generate Ticket:Customers and operators can generate tickets for the problems.
View Status: Customers can view the status of the tickets.

4.3.8-Level 1.6

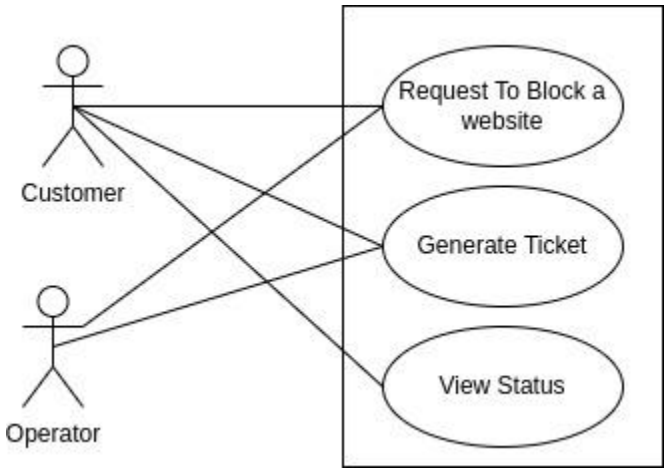
Name: Chat Box
Primary Actor: Customer, Operator.



Description:
Message: Customer and operator can message each other.
Voice Call: Customer and operator can call each other.

4.3.9-Level 1.7

Name: Web Manager
Primary Actor: Customer, Operator.

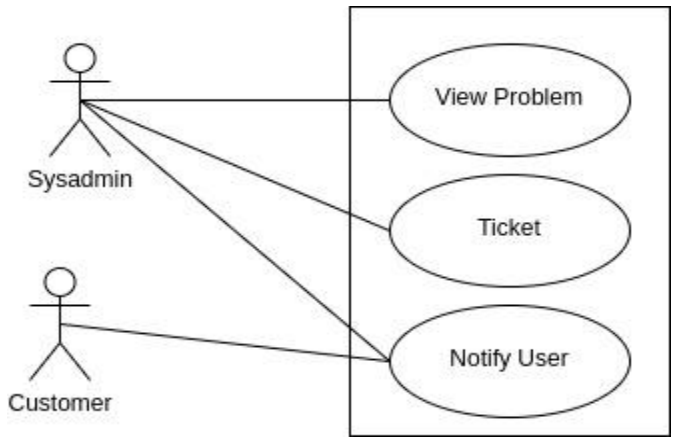


Description:

Request to Block a website: Customer and operator can request to block a website for a particular internet package.
Generate ticket: Customer and operator can generate a ticket for a problem.
View Status: Customers can see the status of the particular ticket.

4.3.10-Level 1.8

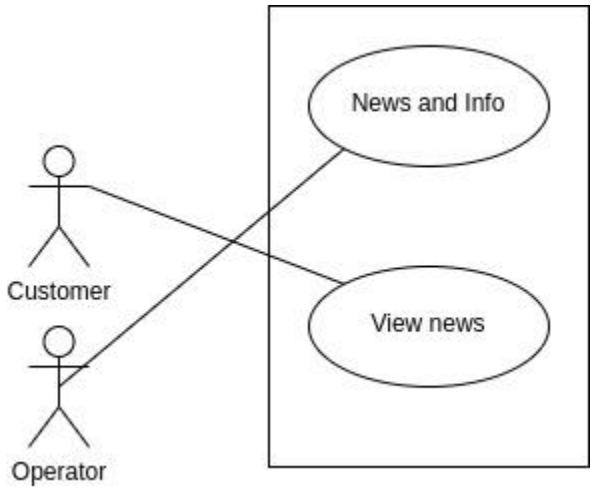
Name: Solving Ticket
Primary Actor: Sysadmin, Customer.



Description:
View Problem: Sysadmin can view the problem.
Ticket: Sysadmin can solve the ticket.
Notify User: Sysadmin can notify the state of the assigned problem associated by customers. Customer can receive the notification.

4.3.11-Level 1.9

Name: News
Primary Actor: Customer, Operator.



Description:

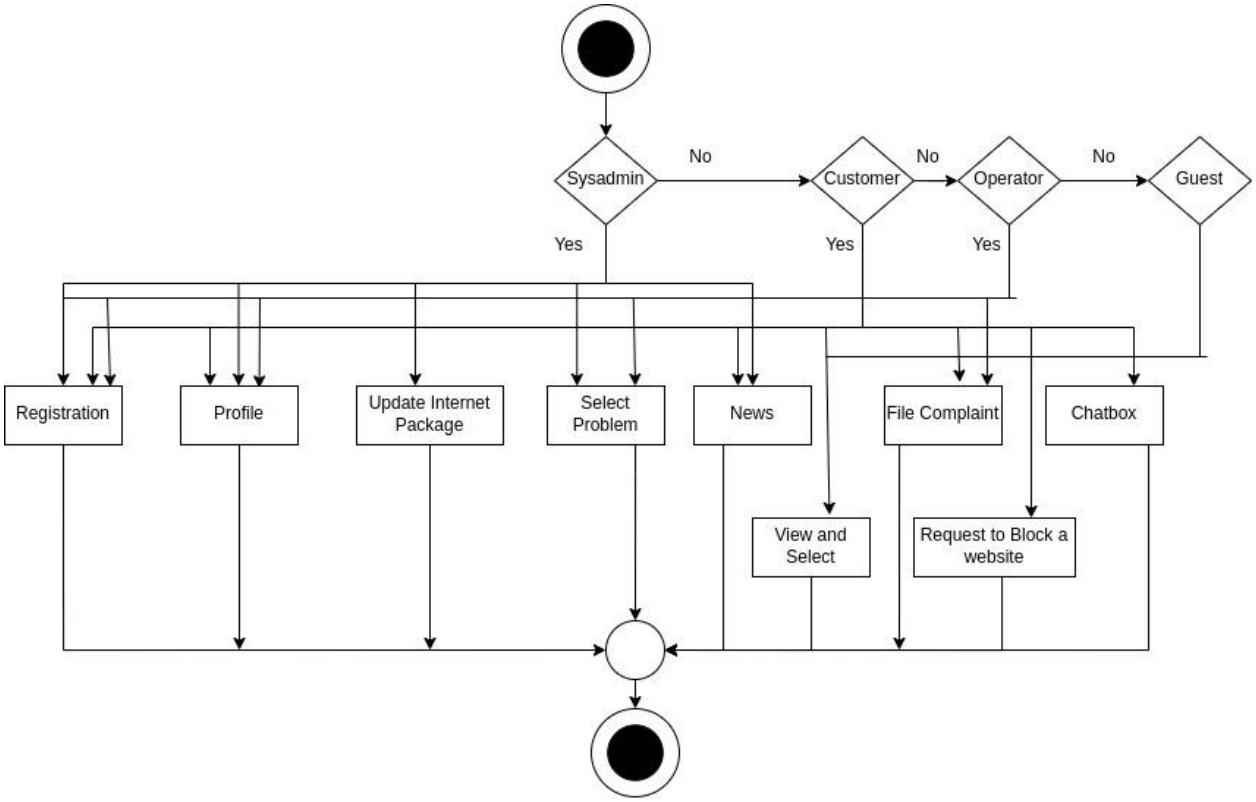
View News: Customers can view the news sent by the operator.
News and Infor: Operators can add news or send information to customers.

4.4 Activity Diagrams

Activity Diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The basic purpose of activity diagrams is to capture the dynamic behavior of the system.

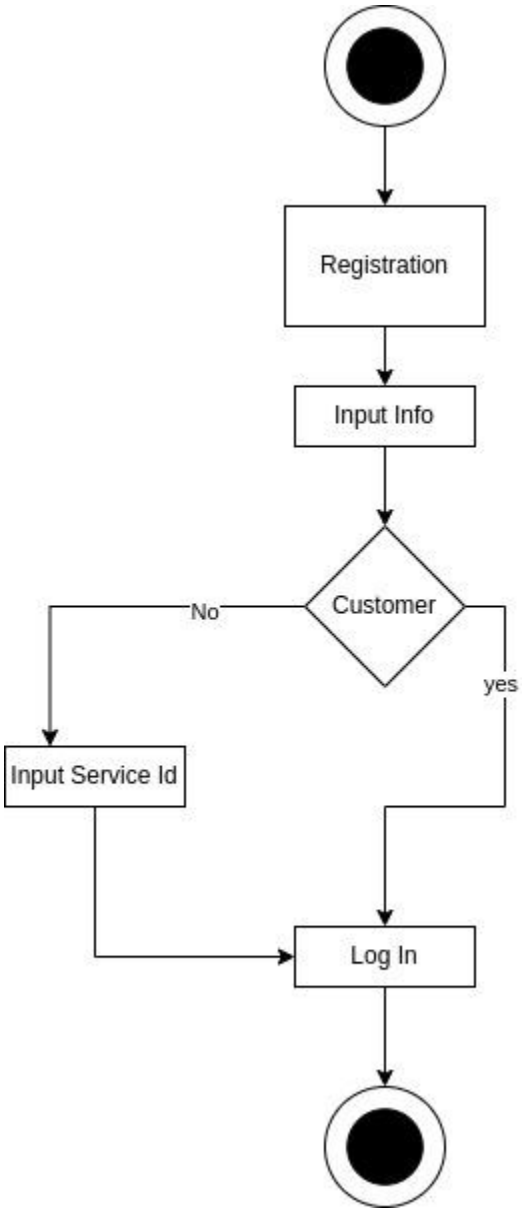
4.4.1-Level 1

Name: SCSI
Reference: Use Case Level 1



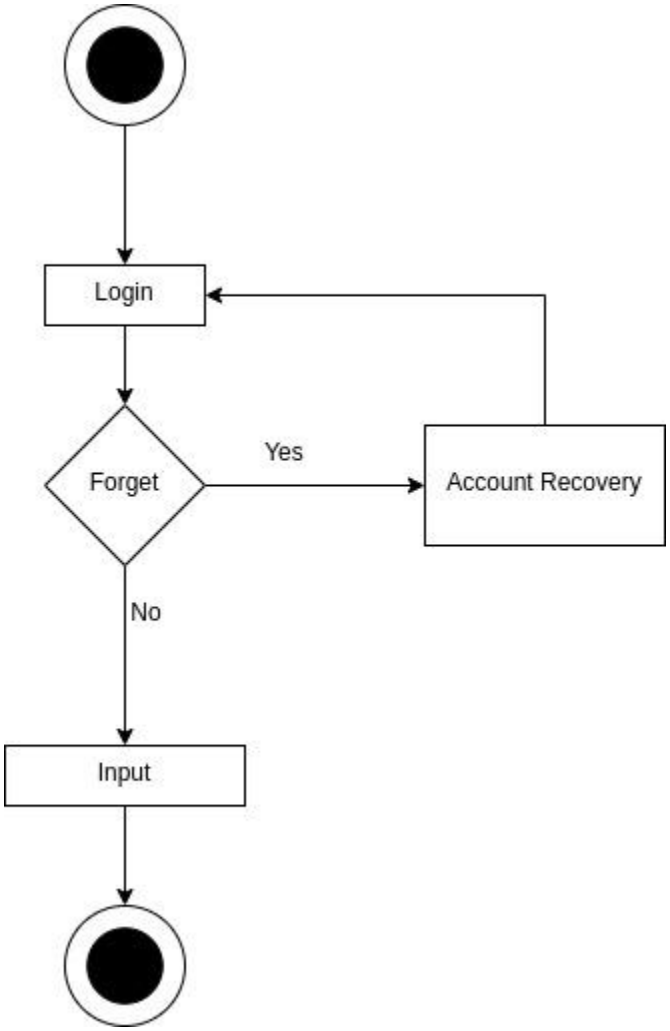
4.4.2-Level 1.1.1

Name: Account Registration
Reference: Use Case Level 1.1



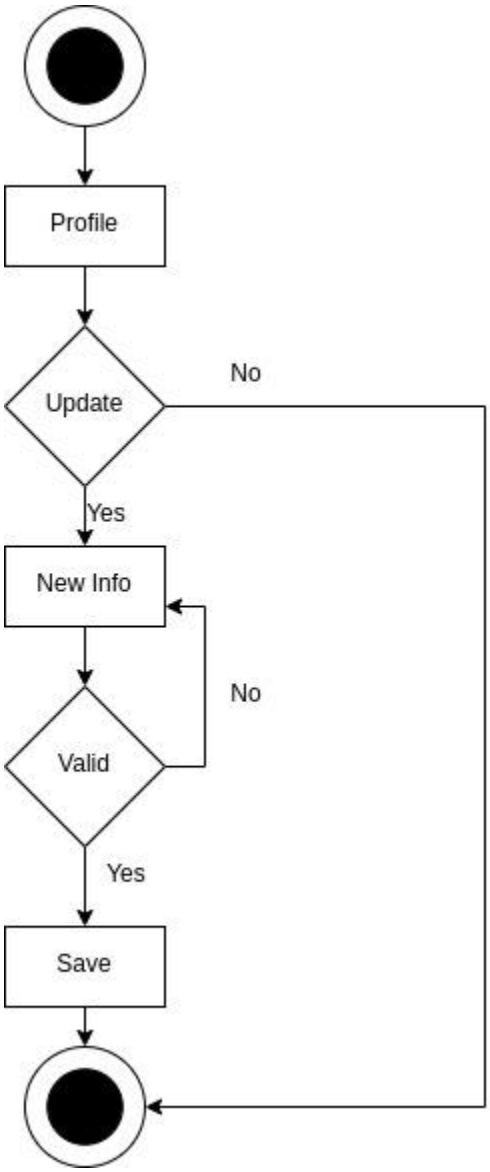
4.4.3-Level 1.1.2

Name: Account and Recovery
Reference: Use Case Level 1.1



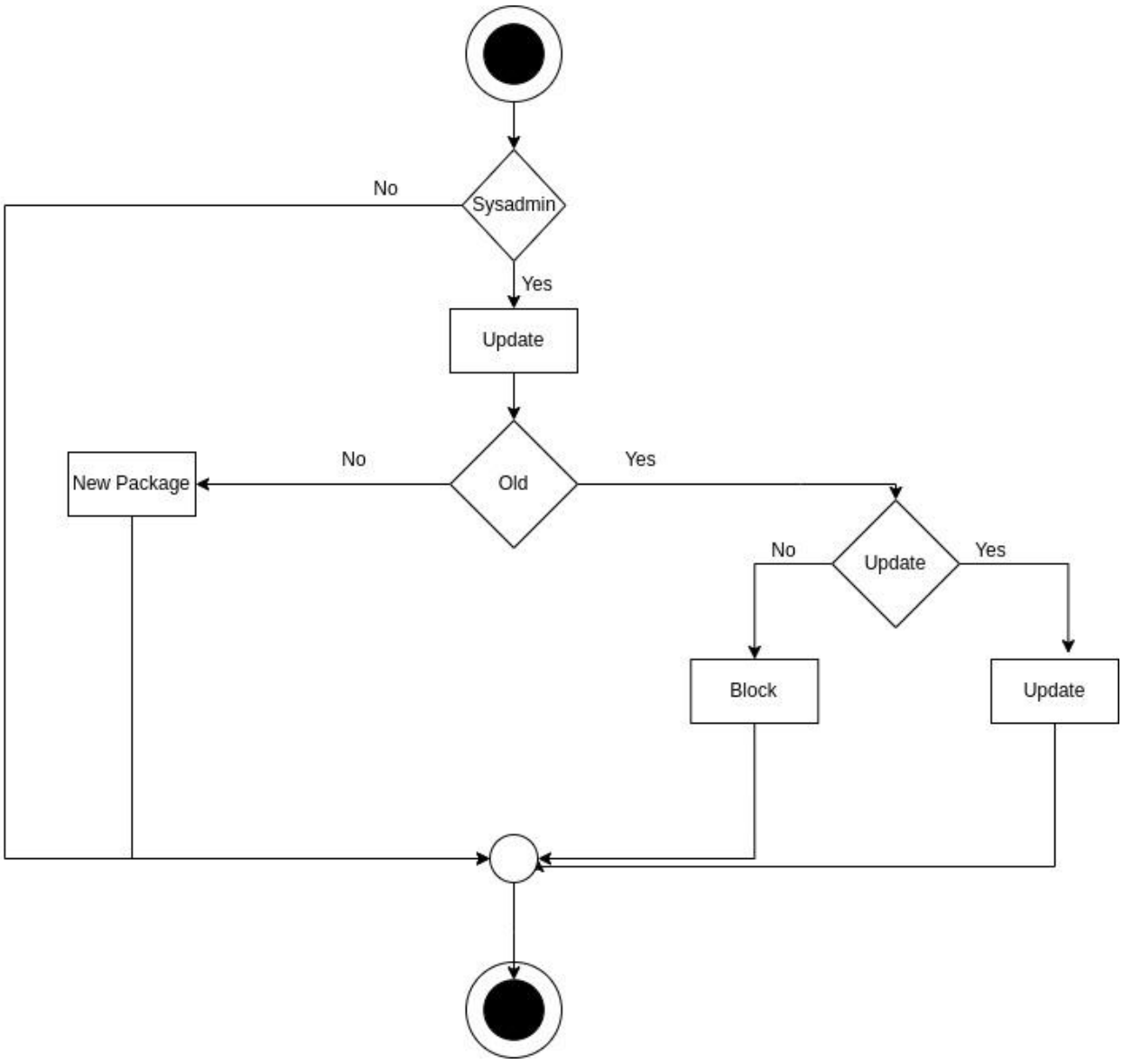
4.4.4-Level 1.2

Name: Profile
Reference: Use Case Level 1.2



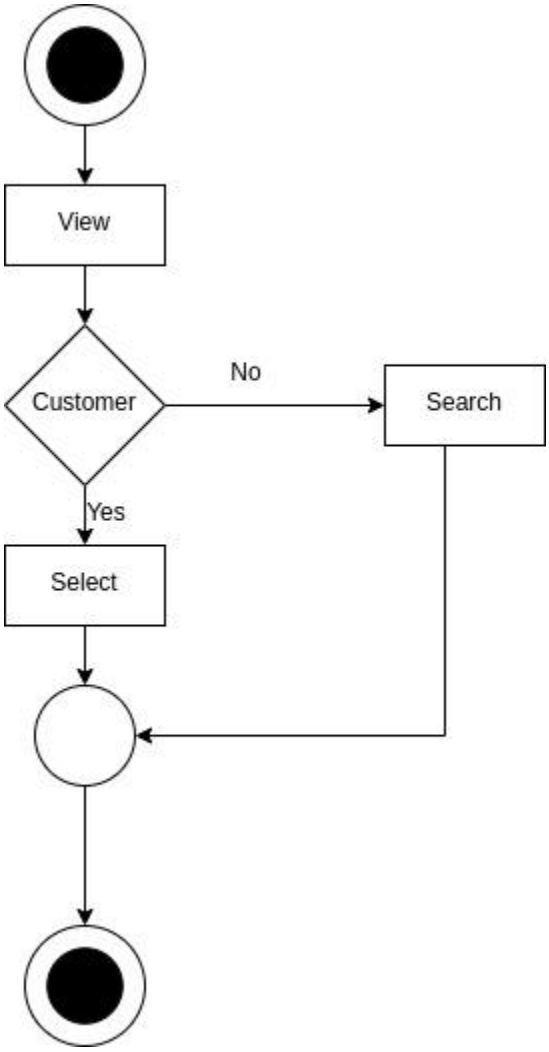
4.4.5-Level 1.3

Name: Update Internet Package
Reference: Use Case Level 1.3



4.4.6-Level 1.4

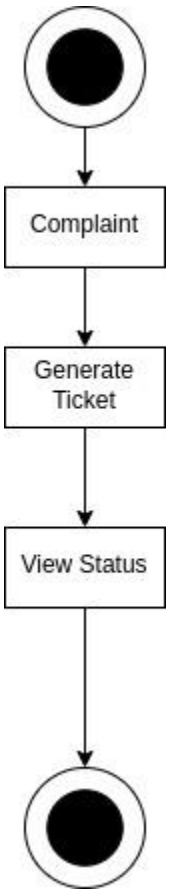
Name: Select Package
Reference: Use Case Level 1.4



4.4.7-Level 1.5

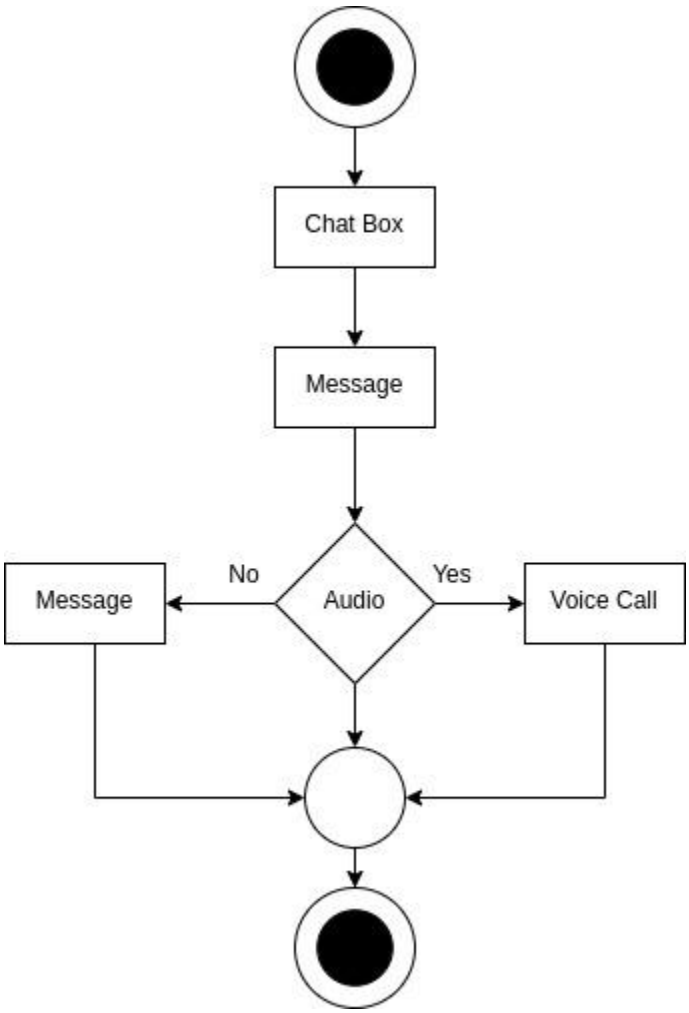
Name: File Complaint

Reference: Use Case Level 1.5



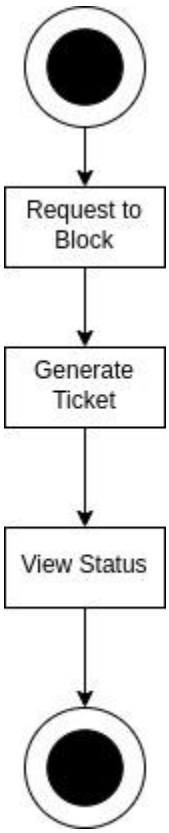
4.4.8-Level 1.6

Name: Chat Box
Reference: Use Case Level 1.6



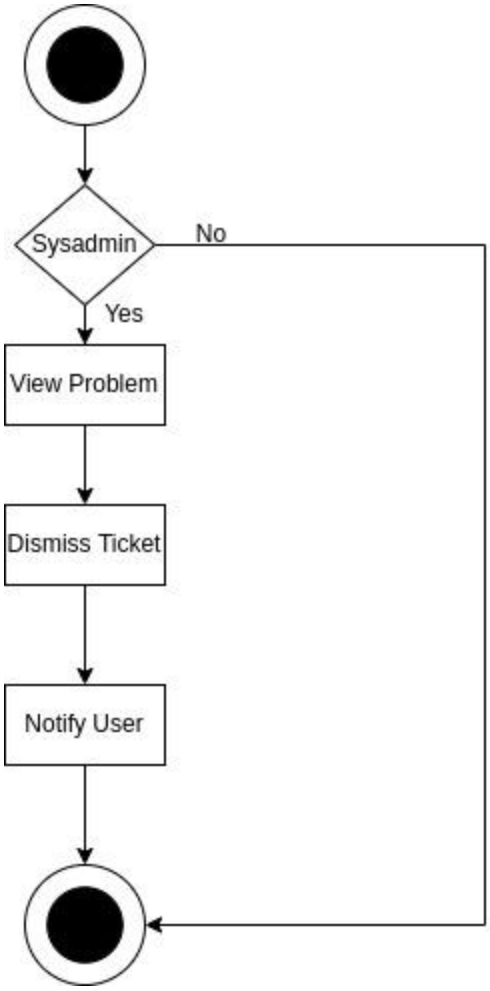
4.4.9-Level 1.7

Name: Web Manager
Reference: Use Case Level 1.7



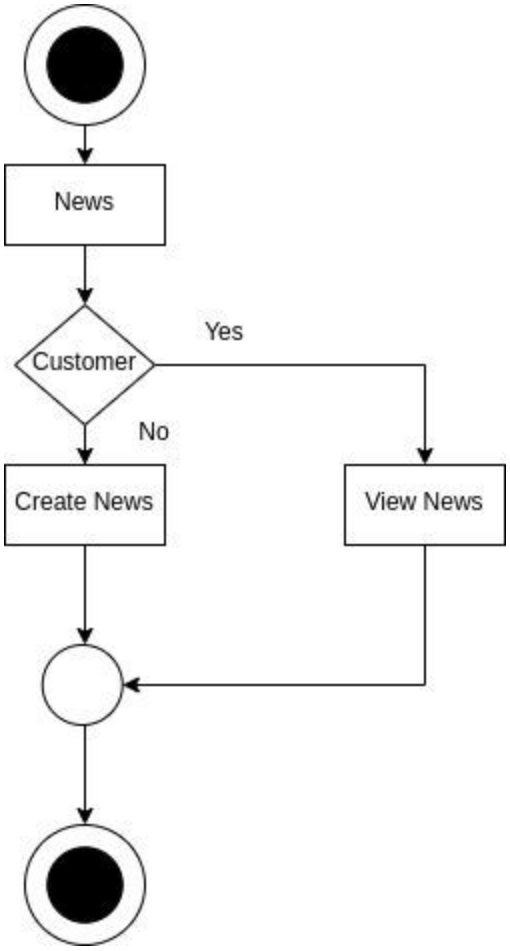
4.4.10-Level 1.8

Name: Solving Ticket
Reference: Use Case Level 1.8



4.4.11-Level 1.9

Name: News
Reference: Use Case Level 1.9


News
Customer
Yes
No
Create News
View News

4.5 Swimlane

A Swimlane diagram is a type of flowchart that outlines who does what in a given process. Based on the analogy of lanes in a pool, a swimlane diagram places process steps within the horizontal or vertical “swimlanes” of a particular department, workgroup, or employee, thus ensuring clarity and accountability. Highlighting connections and communications between these lanes, it can serve as an indicator of waste, redundancy, and inefficiency in a process.

4.5.1-Swimlane ID (SID) 1

Name: SCSI (Smart Customer Care Solution for ISP)
Reference: Use Case & Activity Level 1

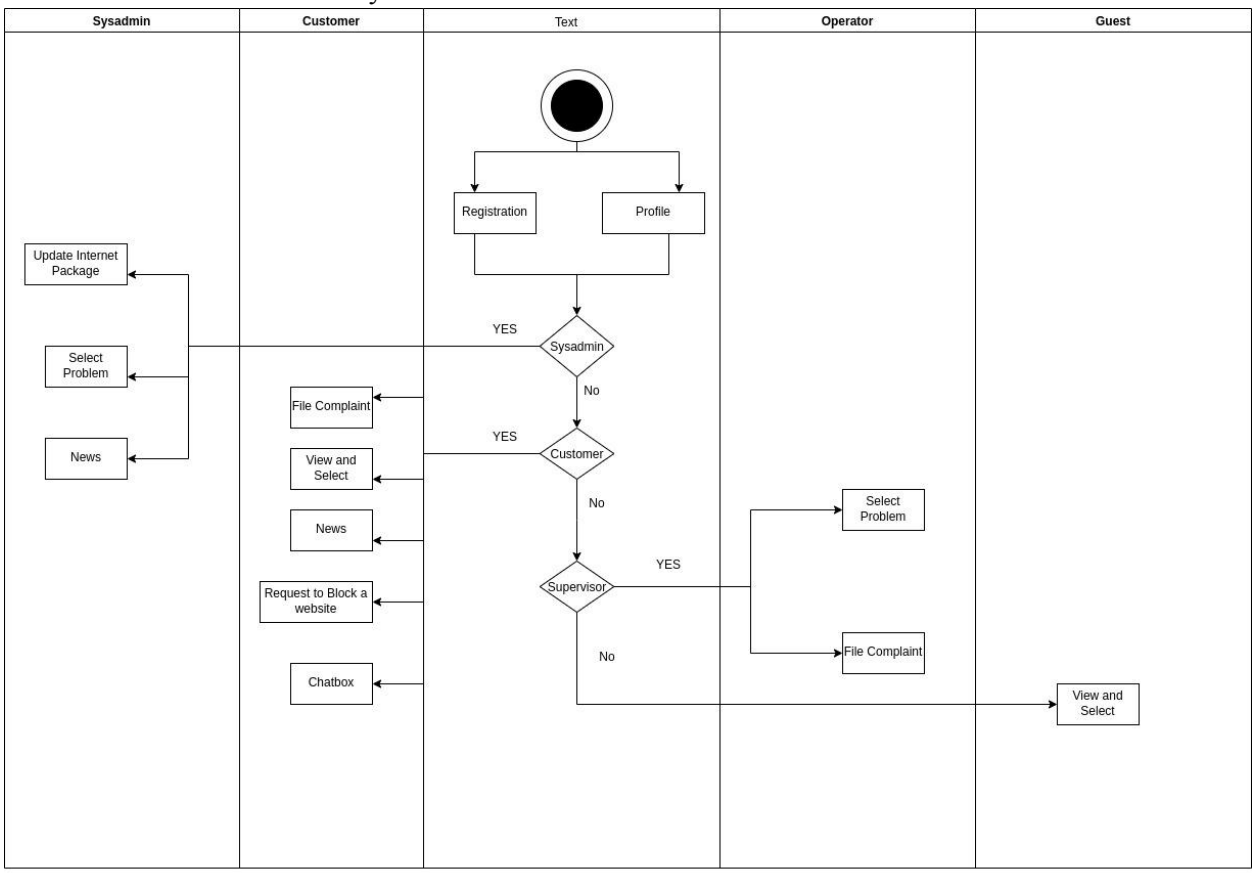
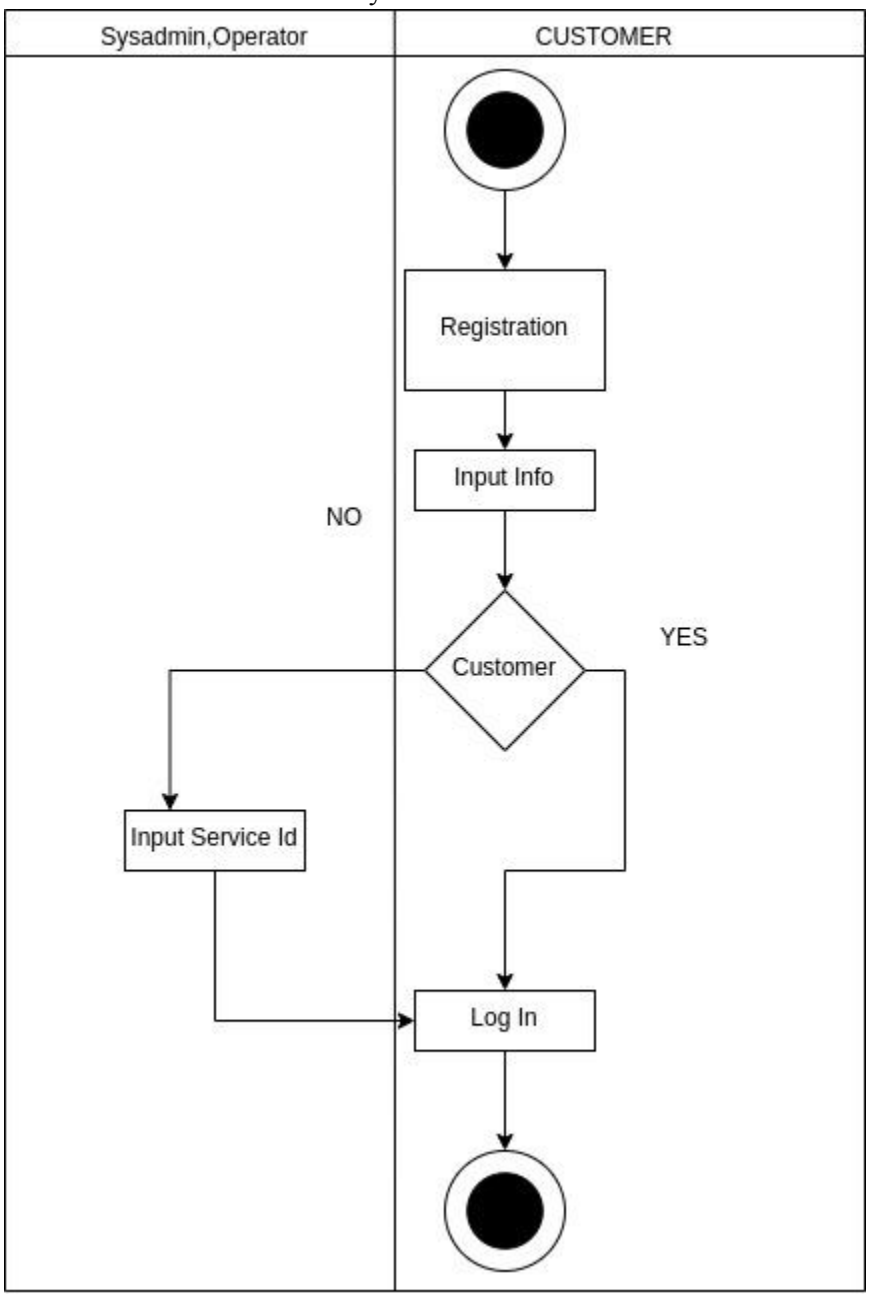


fig 14: level 1 (SPMA)

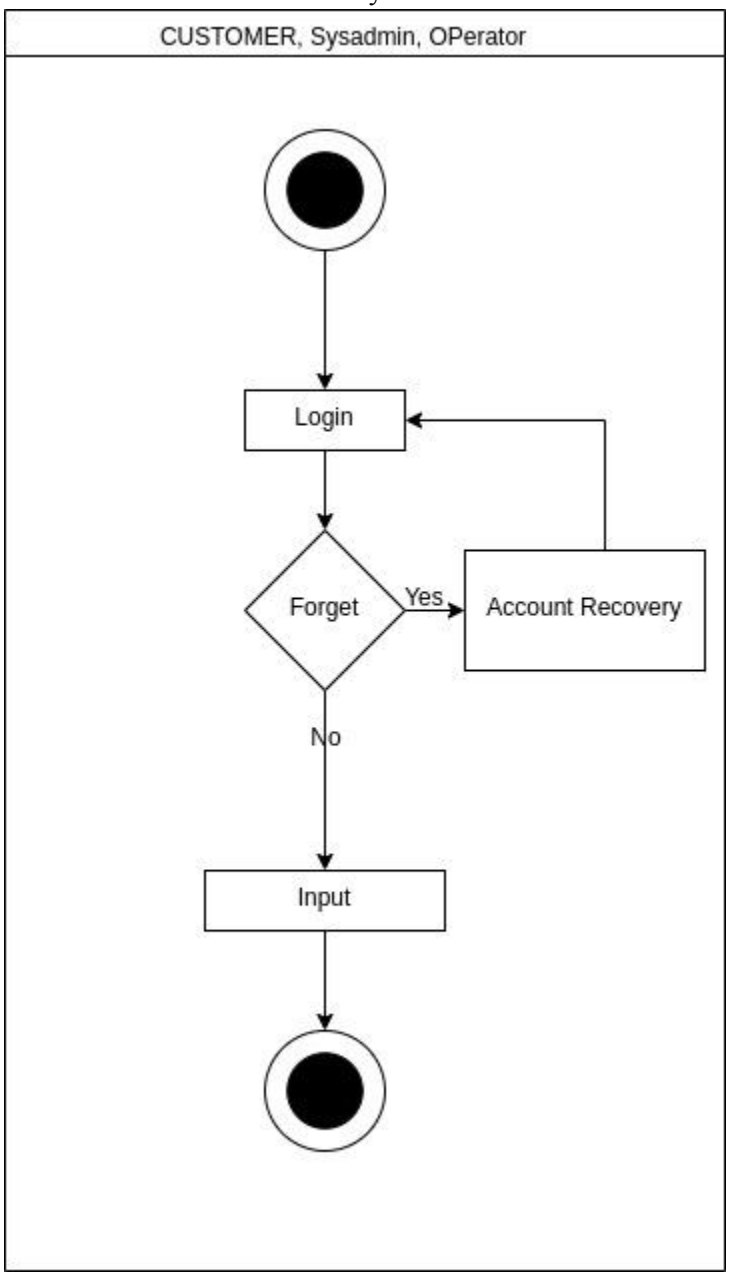
4.5.2-Swimlane ID (SID) 1.1.1

Name: Account and Registration
Reference: Use Case & Activity Level 1.1.1



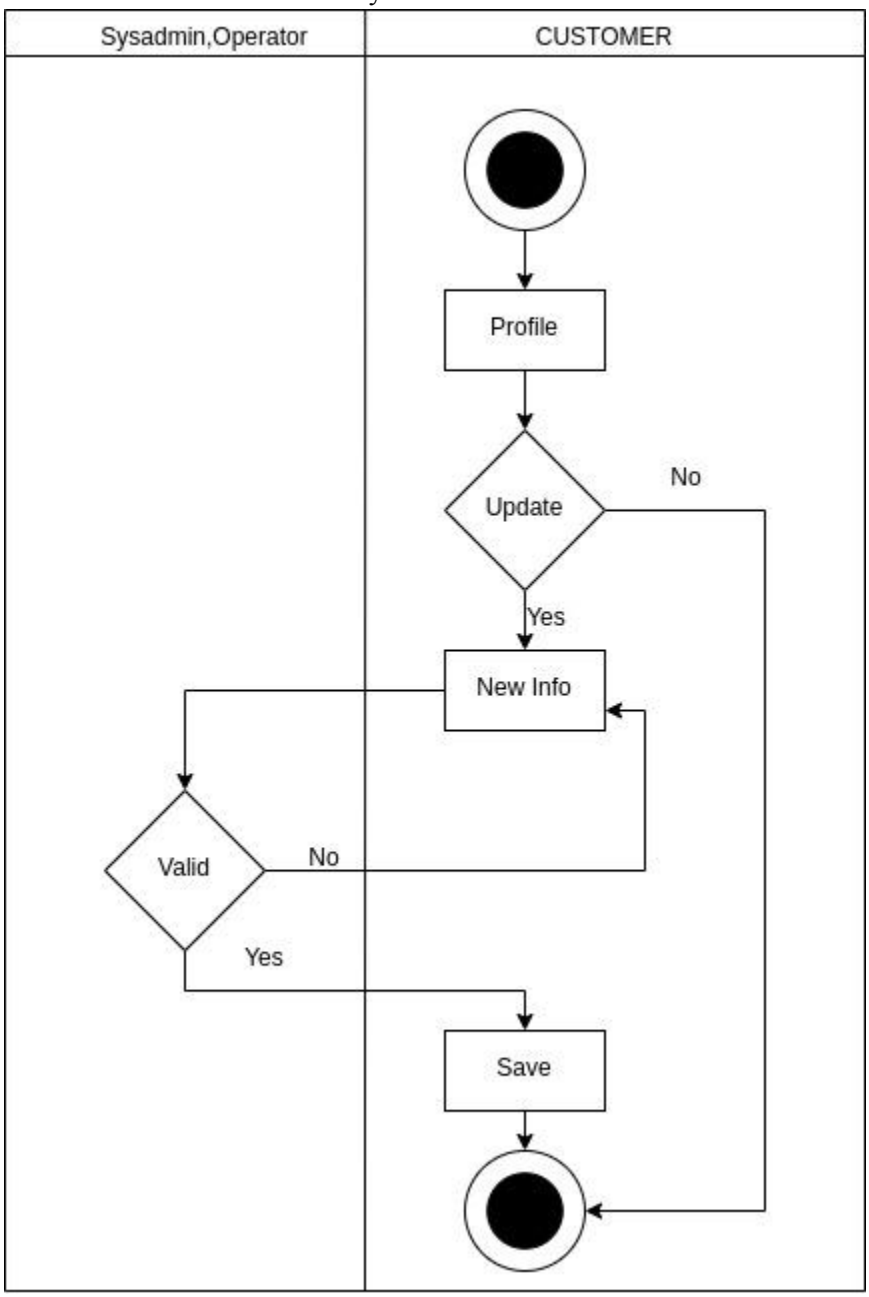
4.5.3-Swimlane ID (SID) 1.1.2

Name: Account and Recovery
Reference: Use Case & Activity Level 1.1.2



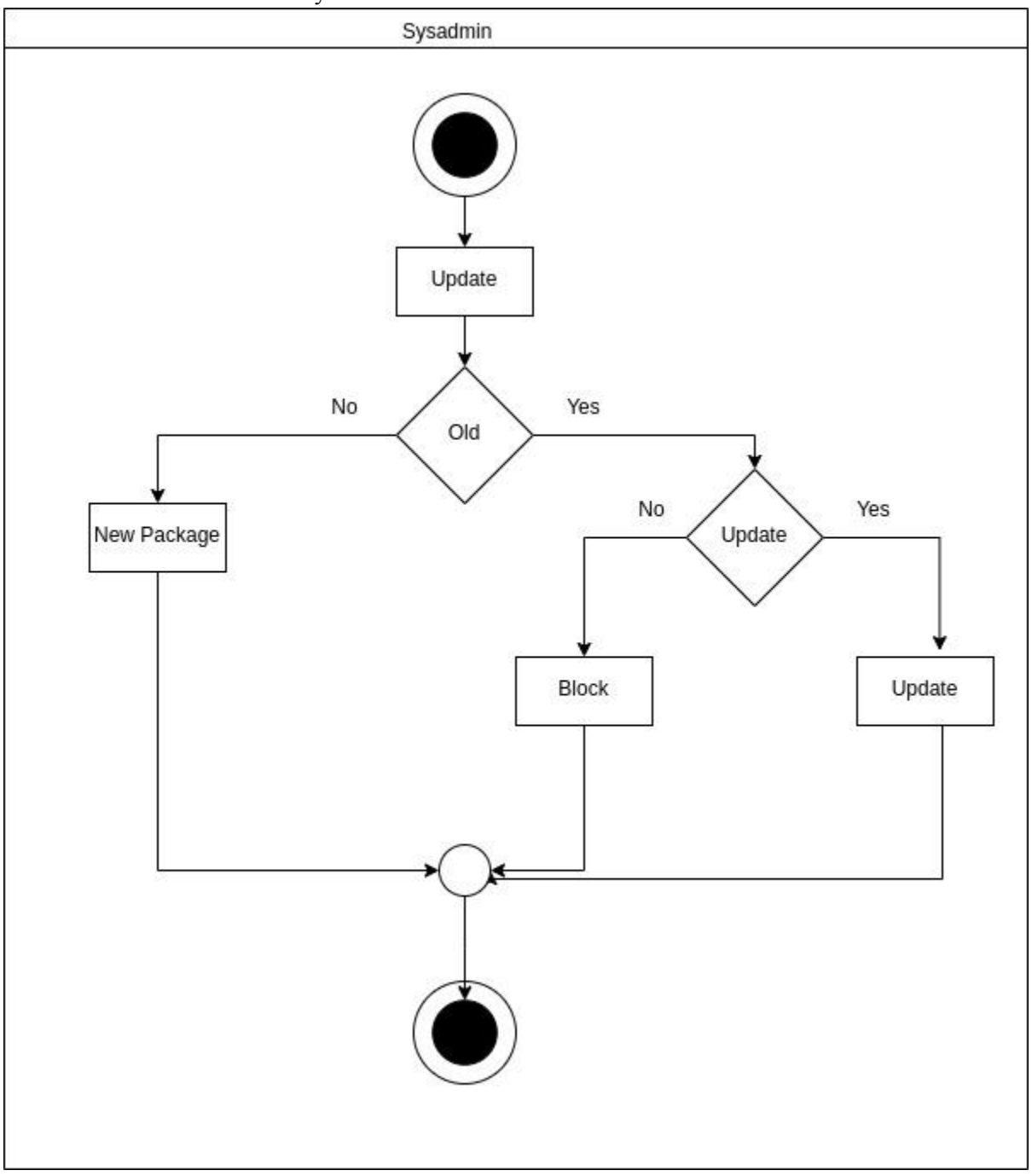
4.5.4-Swimlane ID (SID) 1.2

Name: Profile
Reference: Use Case & Activity Level 1.2



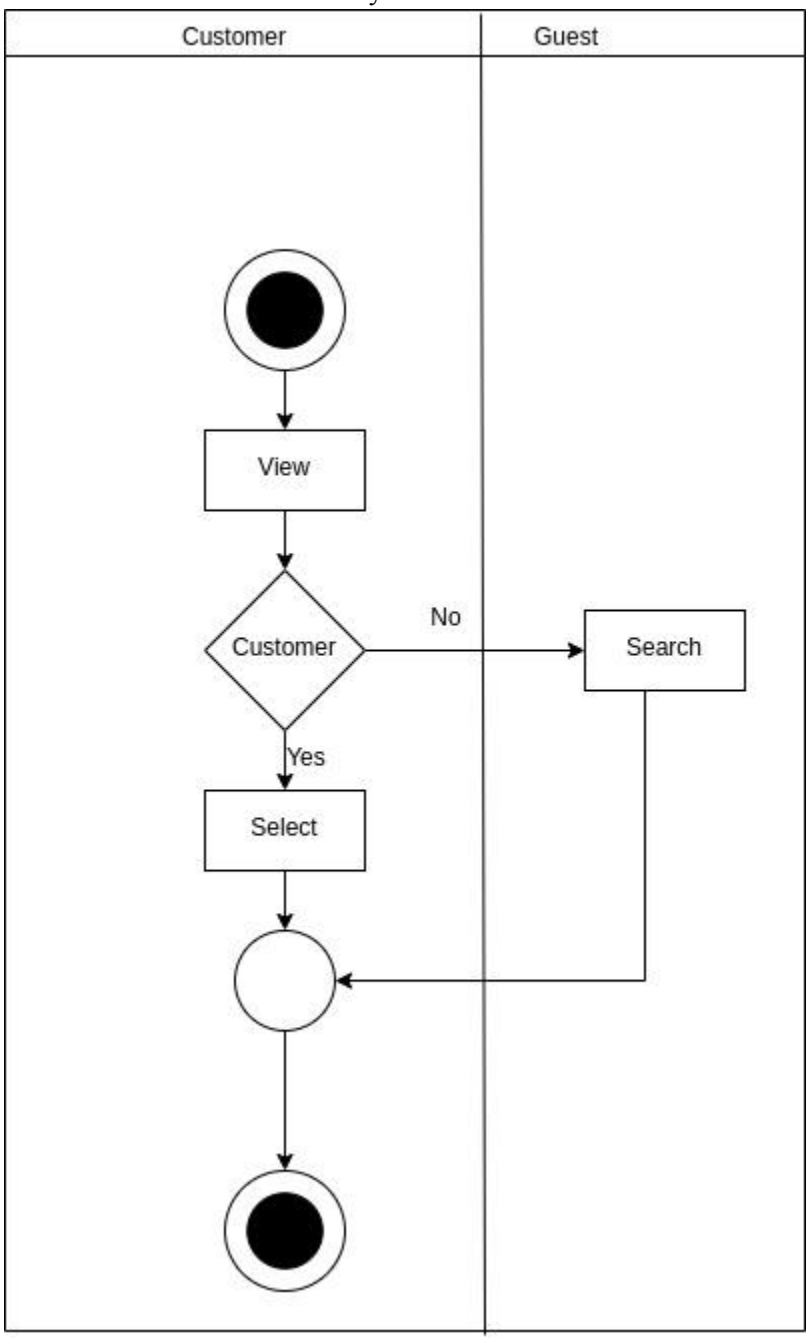
4.5.5-Swimlane ID (SID) 1.3

Name: Update Internet Package
Reference: Use Case & Activity Level 1.3



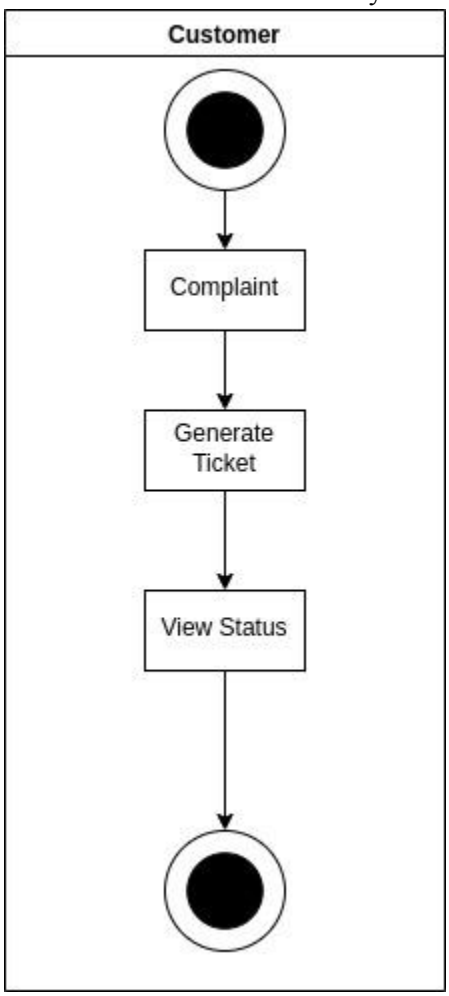
4.5.6-Swimlane ID (SID) 1.4

Name: Select Package
Reference: Use Case & Activity Level 1.4



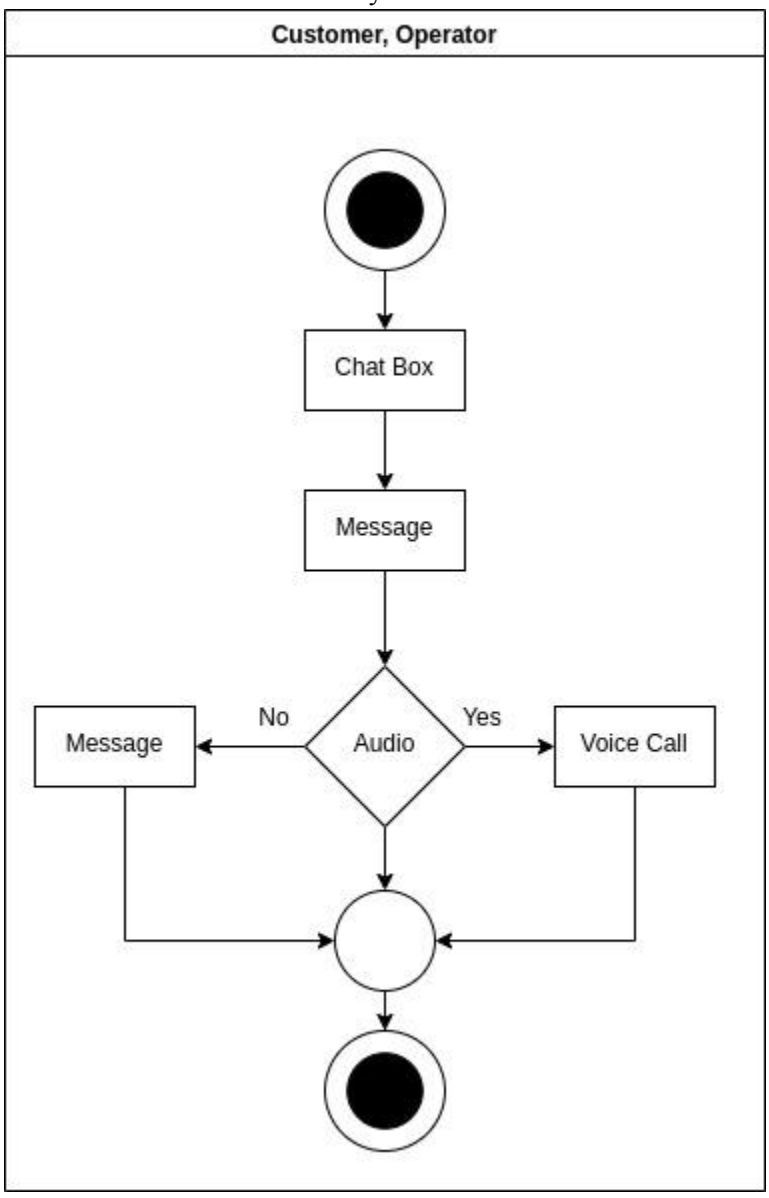
4.5.7-Swimlane ID (SID) 1.5

Name: File Complaint
Reference: Use Case & Activity Level 1.5



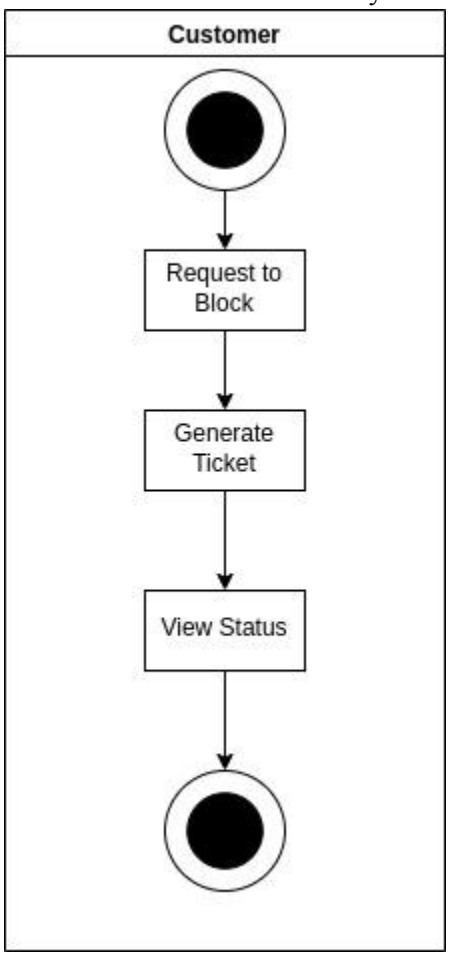
4.5.8-Swimlane ID (SID) 1.6

Name: Chat Box
Reference: Use Case & Activity Level 1.6



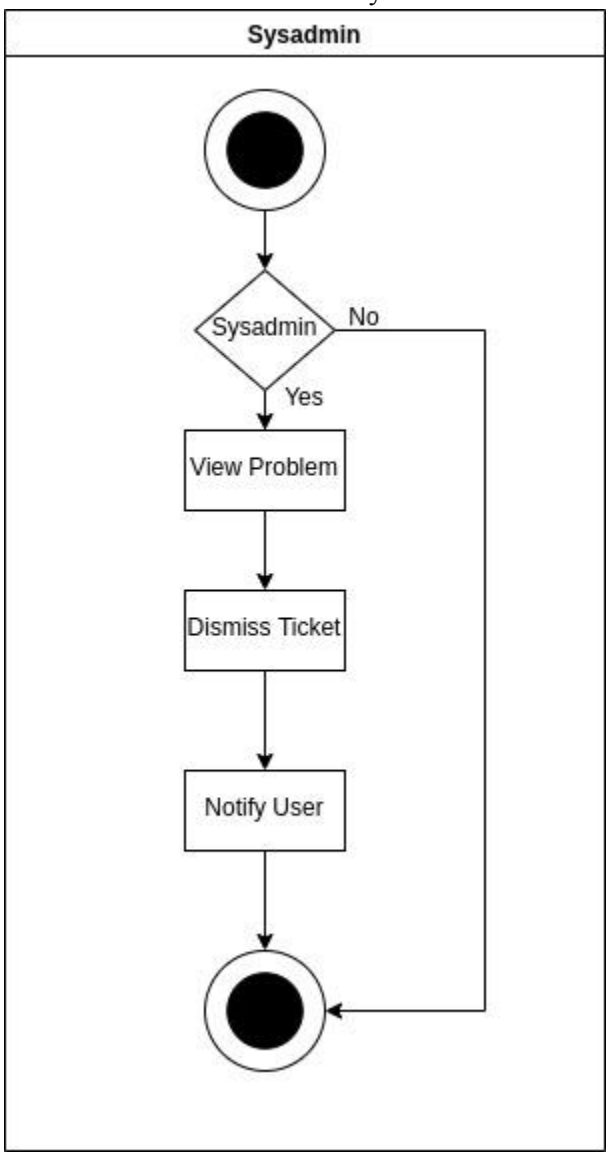
4.5.9-Swimlane ID (SID) 1.7

Name: Web Manager
Reference: Use Case & Activity Level 1.7



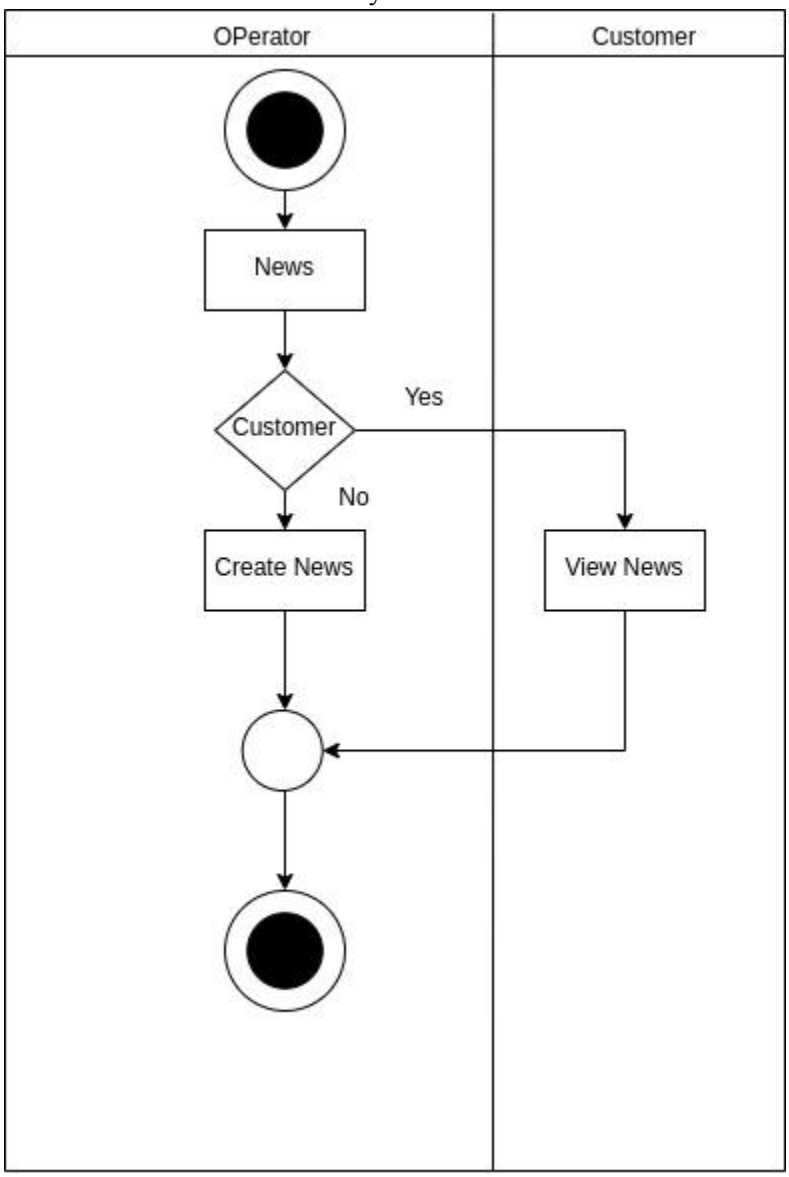
4.5.10-Swimlane ID (SID) 1.8

Name: Solving Ticket
Reference: Use Case & Activity Level 1.8



4.5.11-Swimlane ID (SID) 1.9

Name: NewsInfo
Reference: Use Case & Activity Level 1.9



CHAPTER 5

DATA BASED MODELING

5.1 Introduction

Sometimes software requirements include the necessity to create, extend or interact with a database or complex data structures that need to be constructed and manipulated. The software team chooses to create data models as a part of overall requirements modeling. The entity-relationship diagram (ERD) defines all data objects that are processed within the system, the relationships between the data objects, and the information about how the data objects are entered, stored, transformed, and produced within the system.

5.2 Data Objects

A data object is a collection of one or more data points that create meaning as a whole. In other words, it's a storage or container to store data values. More specifically, data objects are usable, functional, and meaningful artifacts whose form and function are to encode data. It can be used in type checking operations. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place, or a structure.

5.2.1 Noun Identification

We identified all the nouns whether they are in problem space or in solution space from our usage scenario.

Serial	Noun	S/P	Attributes
1	SCSI	P	
2	Customer	S	7,8,9,11,12,14,15,34,35
3	ISP	P	
4	sysadmin	S	7,8,9,10,14,15,34
5	Customer care	P	
6	operator	S	7,8,9,10,14,15,34
7	account	S	

8	username	S	
9	Mobile number	S	
10	Job id	S	
11	code	S	
12	address	S	
13	user	S	
14	profile	S	
15	Profile photo	S	
16	description	S	
17	internet	P	
18	packages	S	34,36,38,44,45,46,47
19	guest	P	
20	valid	S	
21	list	S	
22	complaint	S	
23	status	S	
24	chatbox	S	
25	message	S	
26	Voice call	S	
27	request	S	
28	block	S	
29	webpage	S	
30	web URL	S	
31	problem	S	
32	ticket	S	7,16,22,23,33,35,36,49
33	date	S	
34	name	S	

35	User id	S	
36	id	S	
37	notification	S	
38	rating	S	
39	news	S	
40	info	S	
41	database	P	
42	regenerated	P	
43	blocked	S	
44	speed	S	
45	cost	S	
46	amount	S	
47	duration	S	
48	price	P	
49	type	S	

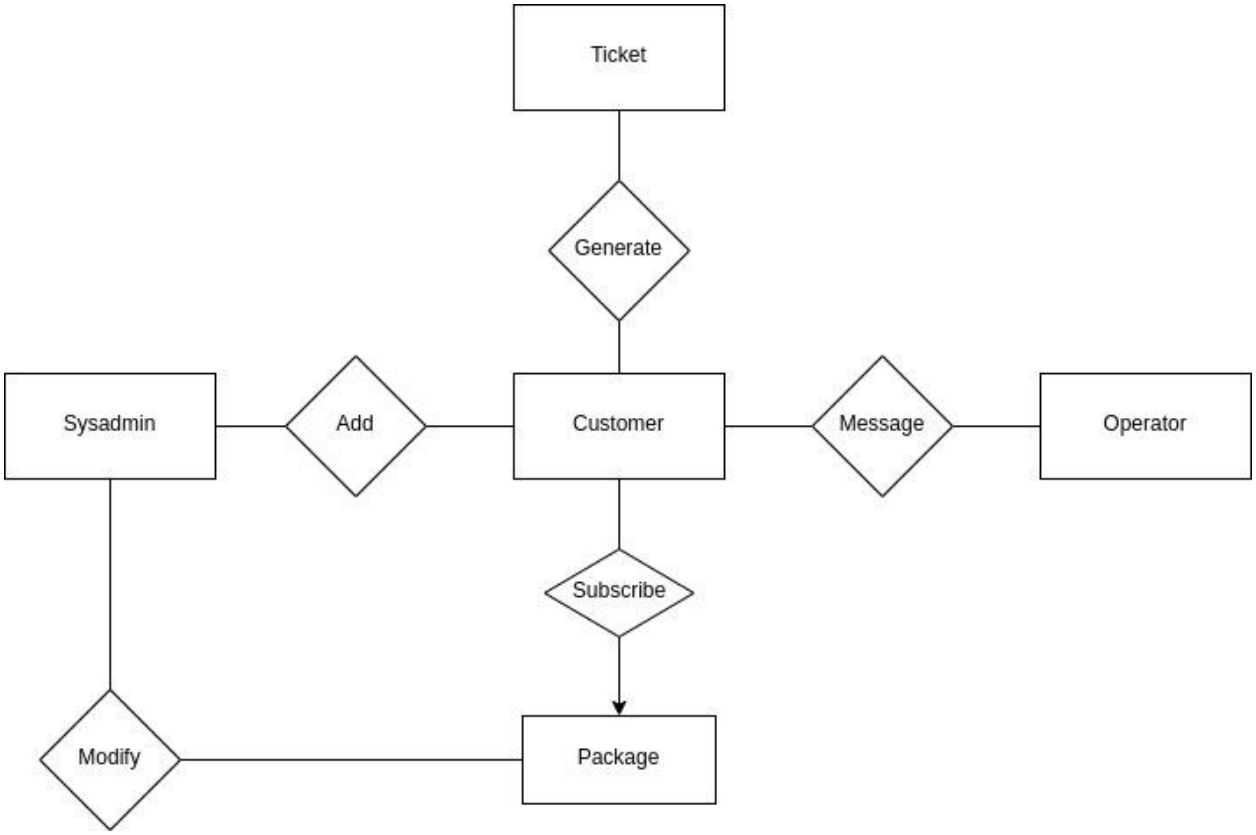
5.2.2 Final Data Objects

Serial	Noun	Attributes
1	Customer	username account mobile number code address profile profile photo name user id package id
2	Sysadmin	username account

		mobile number profile profile photo name job id
3	Operator	username account mobile number profile profile photo name job id
4	Package	name package id rating speed cost duration
5	Ticket	ticket id user id description status date type

5.3 Entity Relationship Diagram

ER Diagram: Definition of ER Diagram : An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.



5.4 Schema Diagram

Customer

Attributes	Type	Size
username	VARCHAR	24
account	VARCHAR	24
mobile number	VARCHAR	15
code	VARCHAR	5
address	VARCHAR	100
profile	VARCHAR	24
profile photo	BLOB	50KB
name	VARCHAR	24
user id	VARCHAR	24
package id	VARCHAR	24

Sysadmin

Attributes	Type	Size
username	VARCHAR	24
account	VARCHAR	24
mobile number	VARCHAR	15
profile	VARCHAR	24
profile photo	BLOB	50KB
name	VARCHAR	24
job id	VARCHAR	24

Operator

Attributes	Type	Size
username	VARCHAR	24
account	VARCHAR	24
mobile number	VARCHAR	15
profile	VARCHAR	24
profile photo	BLOB	50KB
name	VARCHAR	24
operator id	VARCHAR	24

Package

Attributes	Type	Size
name	VARCHAR	24
package id	VARCHAR	24
rating	NUMBER	(1,2)
speed	VARCHAR	5
cost	VARCHAR	5
duration	VARCHAR	5

Ticket

Attributes	Type	Size
ticket id	VARCHAR	24
user id	VARCHAR	24
description	VARCHAR	200
status	VARCHAR	10
date	DATETIME	
type	VARCHAR	10

SysadminAddCustomer

Attributes	Type	Size
sysadminId customerId	VARCHAR VARCHAR	24 24

GenerateTicket

Attributes	Type	Size
ticketId userId	VARCHAR VARCHAR	24 24

ManagePackage

Attributes	Type	Size
packageId sysadminId	VARCHAR VARCHAR	24 24

Message

Attributes	Type	Size
customerId operatorId content timestamp	VARCHAR VARCHAR VARCHAR DATETIME	24 24 100

CHAPTER 6

CLASS BASED MODELING

6.1 Introduction

Class-based modeling identifies classes, attributes, and relationships that the system will use. It represents the object. The system manipulates the operations.

The elements of the class-based model consist of classes and objects, attributes, operations, class – responsibility - collaborator (CRC) models.

Classes are determined using underlining each noun or noun clause and entering it into the simple table. Attributes are the set of data objects that are defining a complete class within the context of the problem. The operations define the behavior of an object.

Noun List

Serial	Noun
1	Customer
2	sysadmin
3	operator
4	account
5	username
6	Mobile number
7	Job id
8	code
9	address
10	user
11	profile
12	Profile photo
13	description

14	packages
15	valid
16	list
17	complaint
18	status
19	chatbox
20	message
21	Voice call
22	request
23	block
24	webpage
25	web URL
26	problem
28	ticket
29	date
30	name
32	User id
33	id
34	notification
35	rating
36	news
37	info
39	blocked
40	speed
41	cost
42	amount
43	duration

44	type

Verb List

Serial	Verb
1	create
2	generate
3	provide
4	sent
5	input
6	add
7	show
8	update
9	update package
10	show
11	view
12	manage
13	search
14	select
15	subscribe
16	file complaint
17	request
18	display
19	notify
20	call
21	login
22	signin

6.2 Identifying and Analysis Classes

Classes are identified by underlining each noun or noun phrase and plotting it into a simple table. If the class (noun) is required to implement a solution, then it becomes a part of the solution space. Otherwise, if the noun is used only to describe a solution, it is regarded as a part of the problem space. Once all the nouns have been isolated, General classification and Selection are done.

6.3 General Classification

Nouns belonging to the solution space should exhibit any of the following criteria to be considered as a class. The 7 general characteristics are stated below:

1. **External entities** (e.g., other systems, devices, people) that produce or consume information to be used by a computer-based system.
2. **Things** (e.g., reports, displays, letters, signals) that are part of the information domain for the problem.
3. **Occurrences or events** (e.g., a property transfer or the completion of a series of robot movements) that occur within the context of system operation.
4. **Roles** (e.g., manager, engineer, salesperson) played by people who interact with the system.
5. **Organizational units** (e.g., division, group, team) that are relevant to an application.
6. **Places** (e.g., manufacturing floor or loading dock) that establish the context of the problem and the overall function of the system.
7. **Structures** (e.g., sensors, four-wheeled vehicles, or computers) that define a class of objects or related classes of objects.

Serial	Noun	General classification
1	Customer	4,5,7
2	sysadmin	4,5,7
3	operator	4,5,7
4	account	2
5	username	2
6	Mobile number	2
7	Job id	2
8	code	2
9	address	6

10	user	2
11	profile	2
12	Profile photo	2
13	description	2
14	packages	2,5,7
15	valid	
16	list	2
17	complaint	
18	status	2
19	chatbox	2
20	message	2
21	Voice call	2
22	request	3,7
23	block	3
24	webpage	2
25	web URL	2
26	problem	
28	ticket	2,5,7
29	date	2
30	name	2
32	User id	2
33	id	2
34	notification	3
35	rating	2
36	news	2
37	info	2
39	blocked	2

40	speed	2
41	cost	
42	amount	2
43	duration	2
44	type	2
45	database	1
46	unblock	2

6.4 Selection Criteria

The six selection characteristics that should be used as you consider each potential class for inclusion in the analysis model:

1. **Retained information:** The potential class will be useful during analysis only if information about it must be remembered so that the system can Function.
2. **Needed services:** The potential class must have a set of identifiable operations that can change the value of its attributes in some way.
3. **Multiple attributes:** During requirement analysis, the focus should be on “major” information; a class with a single attribute may, in fact, be useful during design, but is probably better represented as an attribute of another class during the analysis activity.
4. **Common attributes:** A set of attributes can be defined for the potential class and these attributes apply to all instances of the class.
5. **Common operations:** A set of operations can be defined for the potential class and these operations apply to all instances of the class.
6. **Essential requirements:** External entities that appear in the problem space and produce or consume information essential to the operation of any solution for the system will almost always be defined as classes in the requirements model.

Serial	Noun	Selected criteria
1	Sysadmin	1,2,3,4,5
2	Customer	1,2,3,4,5
3	Operator	1,2,3,4,5
4	Package	2,3,4,5

5	Ticket	2,3,4,5
6	Notification	3,6
7	Database	6
8	WebsiteManager	1,2,3,4,5
9	NewsInfo	1,2,4,5
10	ChatBox	6

6.5 Attribute and Method Selection

Sysadmin

Attribute	Method
-username -account -mobileNumber -profile -profilePhoto -name -job id	+signUp() +logIn() +updateInfo() +updatePhoto() +forgotPassoword() +accountRecovery() +signOut() +addCustomer() +addPackage() +selectTicket()

Customer

Attribute	Method
-username -account -mobile number -code -address -profile -profile photo -name -user id -package id	+signUp() +logIn() +updateInfo() +updatePhoto() +forgotPassoword() +accountRecovery() +signOut() +generateTicket() +subscribePackage() +viewNews() +viewStatus() +messageOperator() +fileComplaint()

Operator

Attribute	Method
-username -account -mobile number -profile -profile photo -name -operator id	+signUp() +logIn() +updateInfo() +updatePhoto() +forgotPassoword() +accountRecovery() +signOut() +generateTicket() +contactCustomer() +notifyCustomer()

Package

Attribute	Method
-name -package id -rating -speed -cost -duration	+updatePackage() +showPackages() +getRating() +subscribePackage() +searchPackage() +selectPackage()

Ticket

Attribute	Method
-ticket id -user id -description -status -date -type	+generateTicket() +viewStatus() +getType()

Notification

Attribute	Method
-id -userId -operatorId -content -time	+notifyUser() +setContent() +notifyAllCustomer()

Database

Attribute	Method
-id -tableNmae -operation	+create() +update() +delete()

WebManager

Attribute	Method
-userId -webURL -blockingStatus	+requestBlock() +requestUnblock() +blockWebsite() +unblockWebsite() +viewStatus()

NewsInfo

Attribute	Method
-operatorId -newsId -date -content -heading	+createNews() +notifyUser() +setContent() +setHeading()

ChatBox

Attribute	Method
-messageId -userId -operatorId -content -timestamp	+message() +voiceCall() +getUserId() +getOperatorId() +setContent()

6.6 Finalizing Classes

Customer, Sysadmin, Operator have common methods and attributes. So we merge them to Account. Here Account is a super class for Customer, Sysadmin and Operator.

We have found the following classes:

1. Account

2. Customer
3. Sysadmin
4. Operator
5. Package
6. Ticket
7. Notification
8. Database
9. WebManager
10. NewsInfo
11. ChatBox

6.7 CRC Card

Account

Responsibilities	Collaborator
● Sign up ● Login ● Update profile ● Account recovery ● Forget Password	Sysadmin Customer Operator Database

Sysadmin

Responsibilities	Collaborator
● Add new customer ● Remove customer ● Add packages ● Update packages ● Solving ticket ● Block website	Customer Package Ticket WebManager

Customer

Responsibilities	Collaborator
● View packages	Sysadmin

• Subscribe package • Rate packages • Generate ticket • Request to block website • Change package • Chat with operator	Operator Ticket Package Notification Database WebManager ChatBox

Operator

Responsibilities	Collaborator
• View users • Generate ticket • Chat with customer • Notify users	Customer Notification Ticket ChatBox

Package

Responsibilities	Collaborator
• Add new packages • Update packages • Get reating • Show packages	Sysadmin Customer

Ticket

Responsibilities	Collaborator
• Generate ticket • Select ticket • Solve ticket	Sysadmin Customer Operator Database

Notification

Responsibilities	Collaborator
• Push notification • View notification	Customer Operator Sysadmin

Database

Responsibilities	Collaborator
• Update database • Create database • Delete database	Sysadmin Customer Operator Package Ticket

WebManager

Responsibilities	Collaborator
• Block website • Unblock website	Sysadmin Customer Database

NewsInfo

Responsibilities	Collaborator
• Notify users • Show news	Operator Customer

ChatBox

Responsibilities	Collaborator
• Message • Voice call	Customer Operator Database

6.8 Class Card

Account

Attribute	Method
-username -account -mobileNumber -profile -profilePhoto -name	+signUp() +logIn() +updateInfo() +updatePhoto() +forgotPassoword() +accountRecovery() +signOut()

Responsibilities	Collaborator
• Sign up • Login • Update profile • Account recovery • Forget Password	Sysadmin Customer Operator Database

Sysadmin

Attribute	Method
-jobId	+addCustomer() +addPackage() +selectTicket() +solveTicket() +updateTicketStatus()
Responsibilities	**Collaborator**
• Add new customer • Remove customer • Add packages • Update packages • Solving ticket • Block website	Customer Package Ticket WebManager

Customer

Attribute	Method
-userId -address -packageId	+generateTicket() +subscribePackage() +viewNews() +viewStatus() +messageOperator() +fileComplaint()
Responsibilities	**Collaborator**
• View packages • Subscribe package • Rate packages • Generate ticket • Request to block website • Change package • Chat with operator	Sysadmin Operator Ticket Package Notification Database WebManager ChatBox

Operator

Attribute	Method
-operatorId	+generateTicket() +contactCustomer() +notifyCustomer()
Responsibilities	**Collaborator**
• View users • Generate ticket • Chat with customer • Notify users	Customer Notification Ticket ChatBox

Package

Attribute	Method
-name -package id -rating -speed -cost -duration	+updatePackage() +showPackages() +getRating() +subscribePackage() +searchPackage() +selectPackage()
Responsibilities	**Collaborator**
• Add new packages • Update packages • Get reating • Show packages	Sysadmin Customer

Ticket

Attribute	Method
-ticket id -user id -description -status -date -type	+generateTicket() +viewStatus() +getType()
Responsibilities	**Collaborator**
• Generate ticket • Select ticket • Solve ticket	Sysadmin Customer Operator Database

Notification

Attribute	Method
-id -userId -operatorId -content -time	+notifyUser() +setContent() +notifyAllCustomer()
Responsibilities	**Collaborator**
• Push notification • View notification	Customer Operator Sysadmin

Database

Attribute	Method
-id -tableNmae -operation	+create() +update() +delete()
Responsibilities	**Collaborator**
• Update database • Create database • Delete database	Sysadmin Customer Operator Package Ticket

WebManager

Attribute	Method
-userId -webURL -blockingStatus	+requestBlock() +requestUnblock() +blockWebsite() +unblockWebsite() +viewStatus()
Responsibilities	**Collaborator**
• Block website • Unblock website	Sysadmin Customer Database

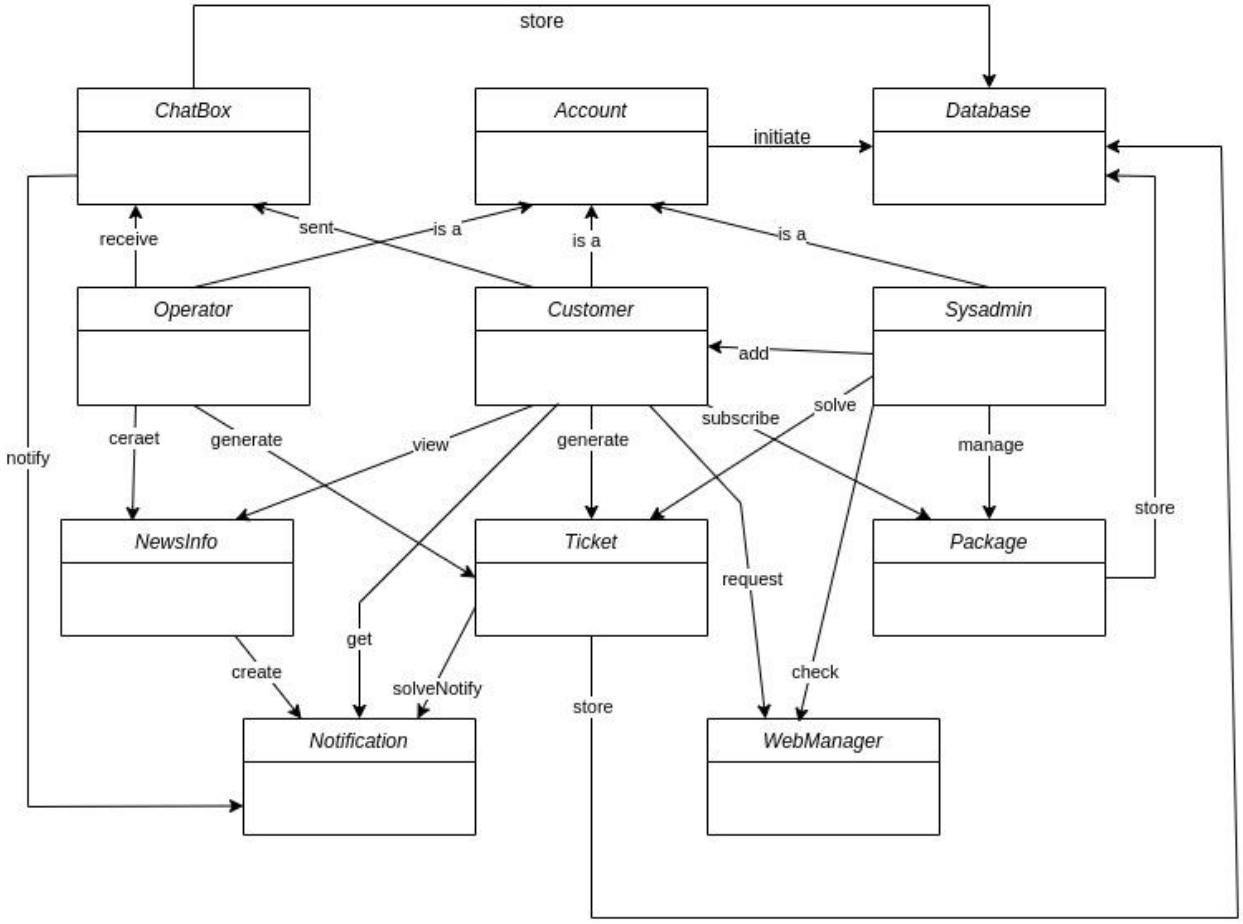
NewsInfo

Attribute	Method
-operatorId -newsId -date -content -heading	+createNews() +notifyUser() +setContent() +setHeading()
Responsibilities	**Collaborator**
• Notify users • Show news	Operator Customer

ChatBox

Attribute	Method
-messageId -userId -operatorId -content -timestamp	+message() +voiceCall() +getUserId() +getOperatorId() +setContent()
Responsibilities	**Collaborator**
• Message • Voice call	Customer Operator Database

6.9 Class Diagram



CHAPTER 7

BEHAVIORAL MODELING

7.1 State Transition Diagram

State diagram represents active states for each class of events (triggers). For this we identified all the events, their initiators and collaborators.

7.1.1 Identifying Event

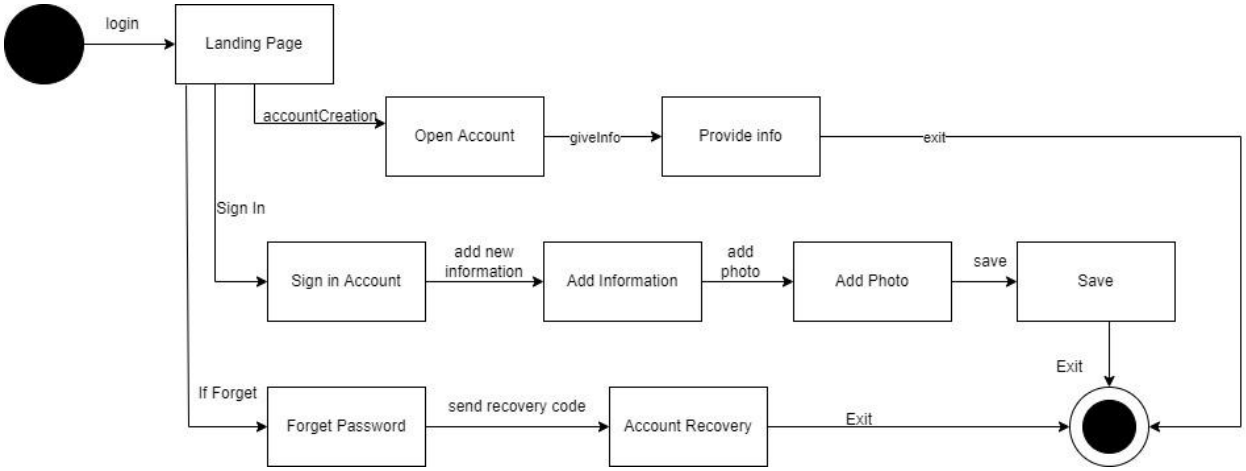
Serial	Events	State Name	Initiator	Collaborator
1	Open an account	CreatingAccount	Account	Sysadmin, Customer, Operator, Database
2	Sign in account	SigninAccount	Account	Sysadmin, Customer, Operator, Database
3	Sign out account	SignOutAccount	Account	Sysadmin, Customer, Operator
4	Add information	AddInformation	Account	Sysadmin, Customer, Operator, Database
5	Add photo	AddPhoto	Account	Sysadmin, Customer, Operator, Database
6	Account recovery	AccountRecovery	Account	Sysadmin, Customer, Operator, Database
7	Add customer	AddCustomer	Sysadmin	Customer
8	Add packages	AddPackage	Sysadmin	Package

9	Solve ticket	SolveTicket	Sysadmin	Ticket
10	Generate ticket	GenerateTicket	Customer, Operator	Ticket
11	Subscribe package	SubscribePackage	Customer	Package, Sysadmin
12	File complaint	FileComplaint	Customer	Sysadmin, Operator, Ticket
13	Message Operator	MessageOperator	Customer	Operator, ChatBox
14	View news	ViewNews	Customer	
15	Update package	UpdatePackage	Package	Sysadmin
16	Show packages	ShowPackages	Package	Customer, Sysadmin
17	Rating	GetRating	Package	Customer
18	Search packages	SearchPackage	Package	Customer
19	Select package	SelectPackage	Package	Customer
20	Contact customer	ContactCustomer	Operator	Customer, Chatbox
21	Notify users	NotifyUsers	Operator	Customer, Notification
22	Create ticket	CreateTicket	Ticket	Customer, Operator
23	Show status	ShowStatus	Ticket	Customer
24	Get type	Get type	Ticket	Sysadmin, Customer
25	Set content	SetContent	Notification	Sysadmin, Customer, Operator
26	Create news	CreateNews	NewsInfo	Operator
27	ViewNews	ViewNews	NewsInfo	Customer
28	Message	Message	ChatBox	Customer, Operator
29	Voice call	VoiceCall	ChatBox	Customer,

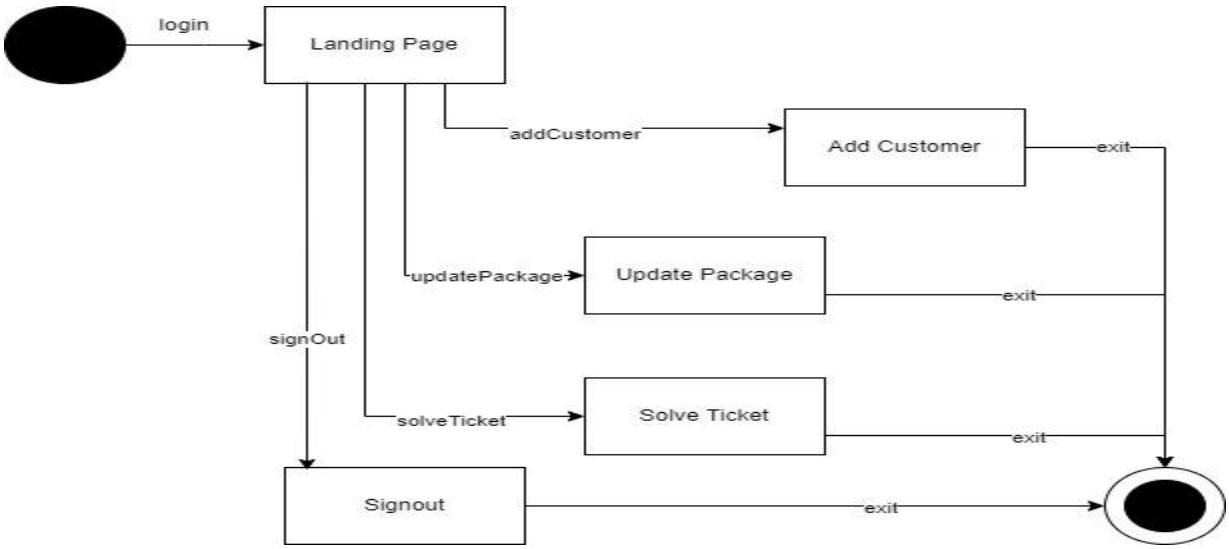
				Operator
30	Request to block	RequestBlocking	WebManager	Customer
31	Request to unblock	RequestUnblocking	WebManager	Customer
32	Block website	BlockWebsite	WebManager	Sysadmin
33	Unblock website	UnblockWebsite	WebManager	Sysadmin
34	View status	ViewStatus	WebManager	Customer, Sysadmin

7.1.2 State Transition Diagrams

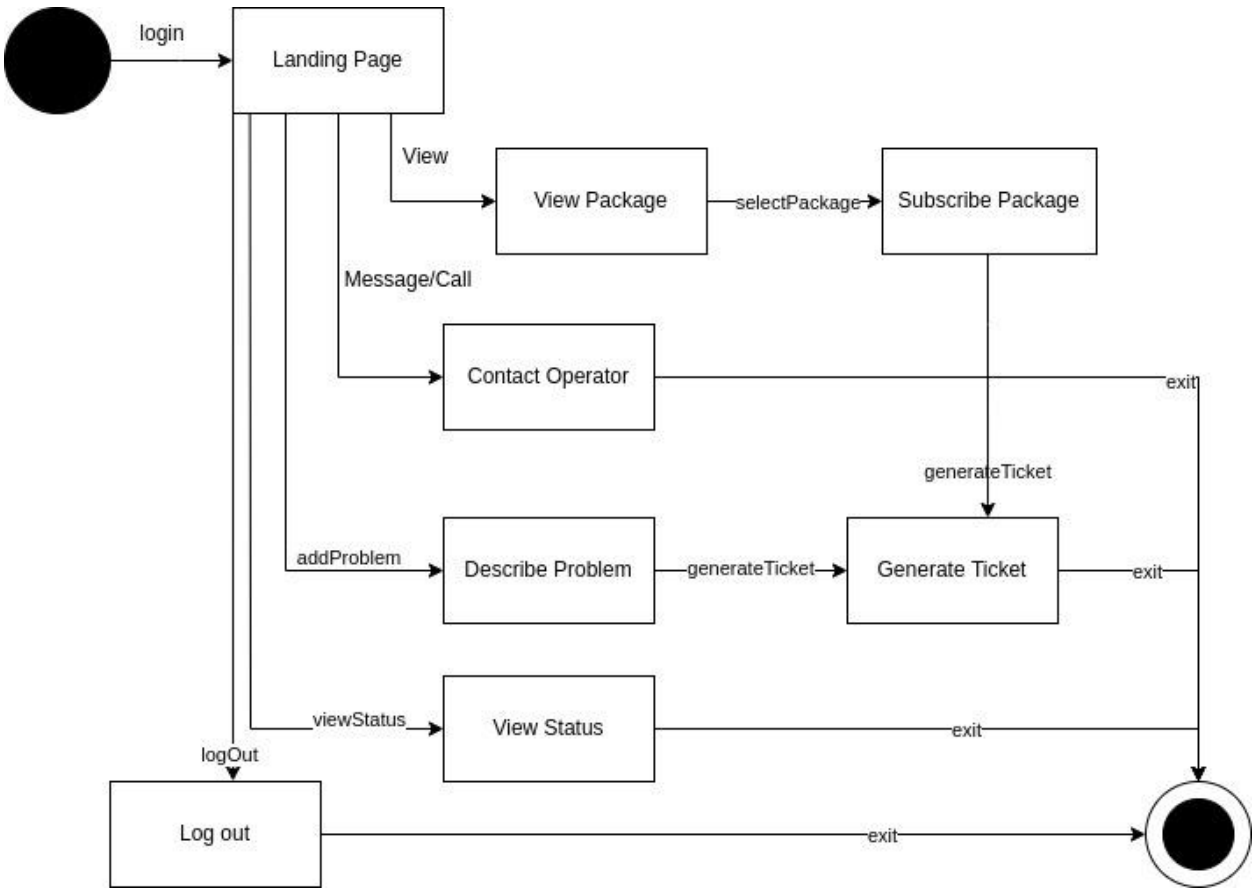
7.1.2.1 ID:1
Name :Account



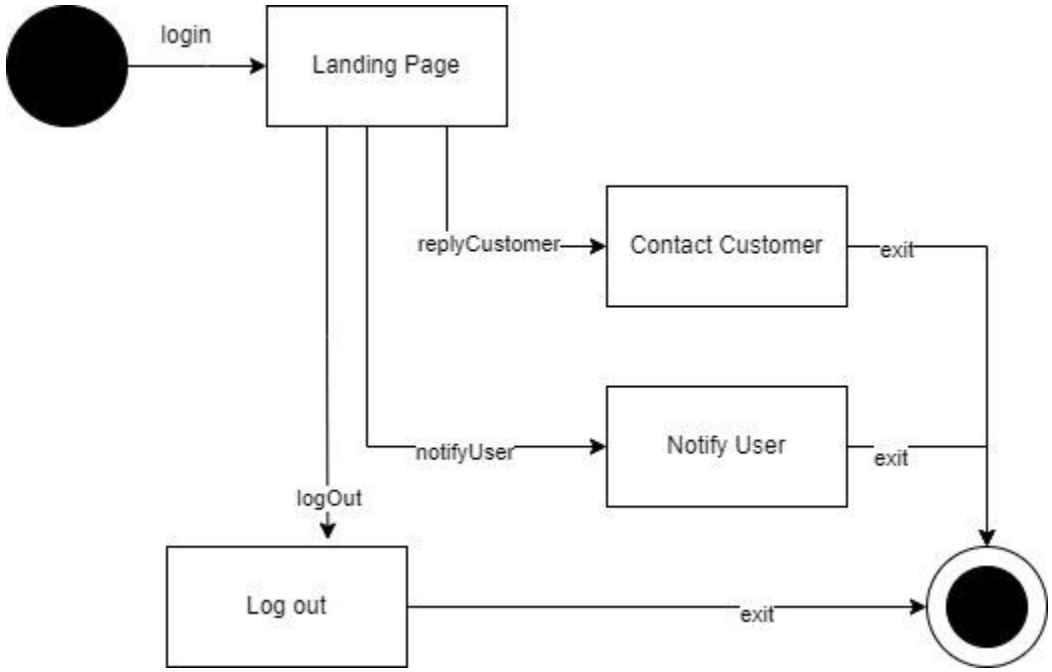
7.1.2.1 ID:2
Name :Sysadmin



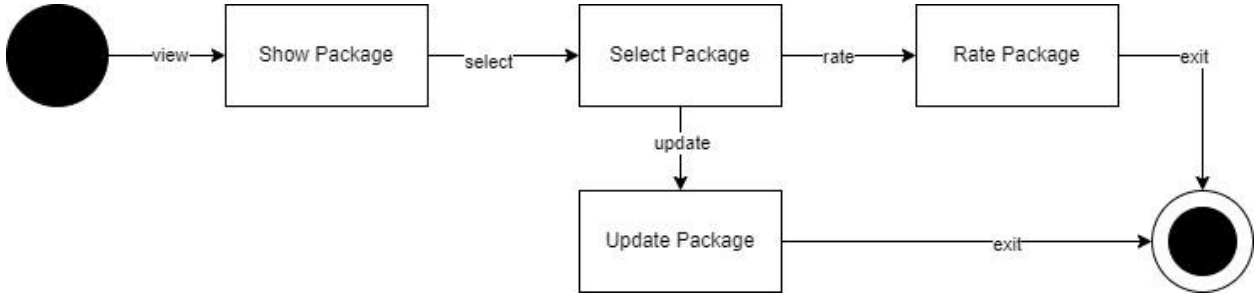
7.1.2.1 ID:3
Name :Customer



7.1.2.1 ID:4
Name :Operator

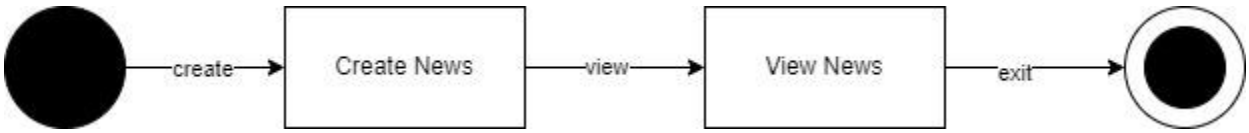


7.1.2.1 ID:5
Name :Package

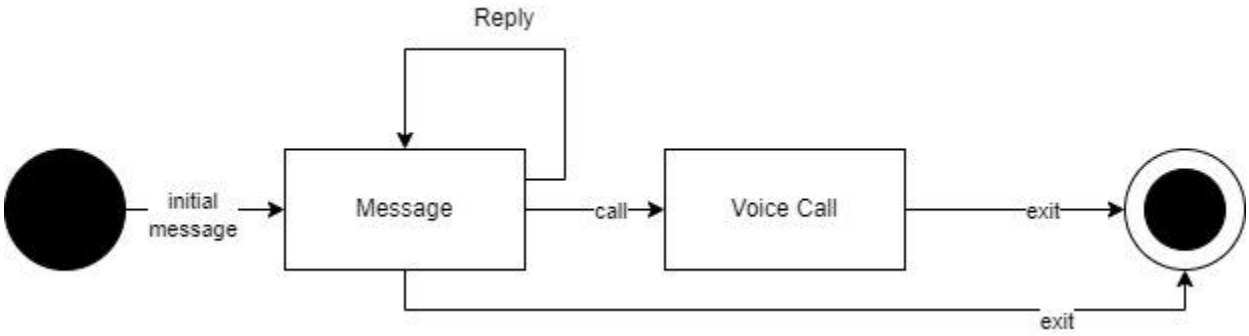


7.1.2.1 ID:6
Name :Ticket

7.1.2.1 ID:7
Name :News

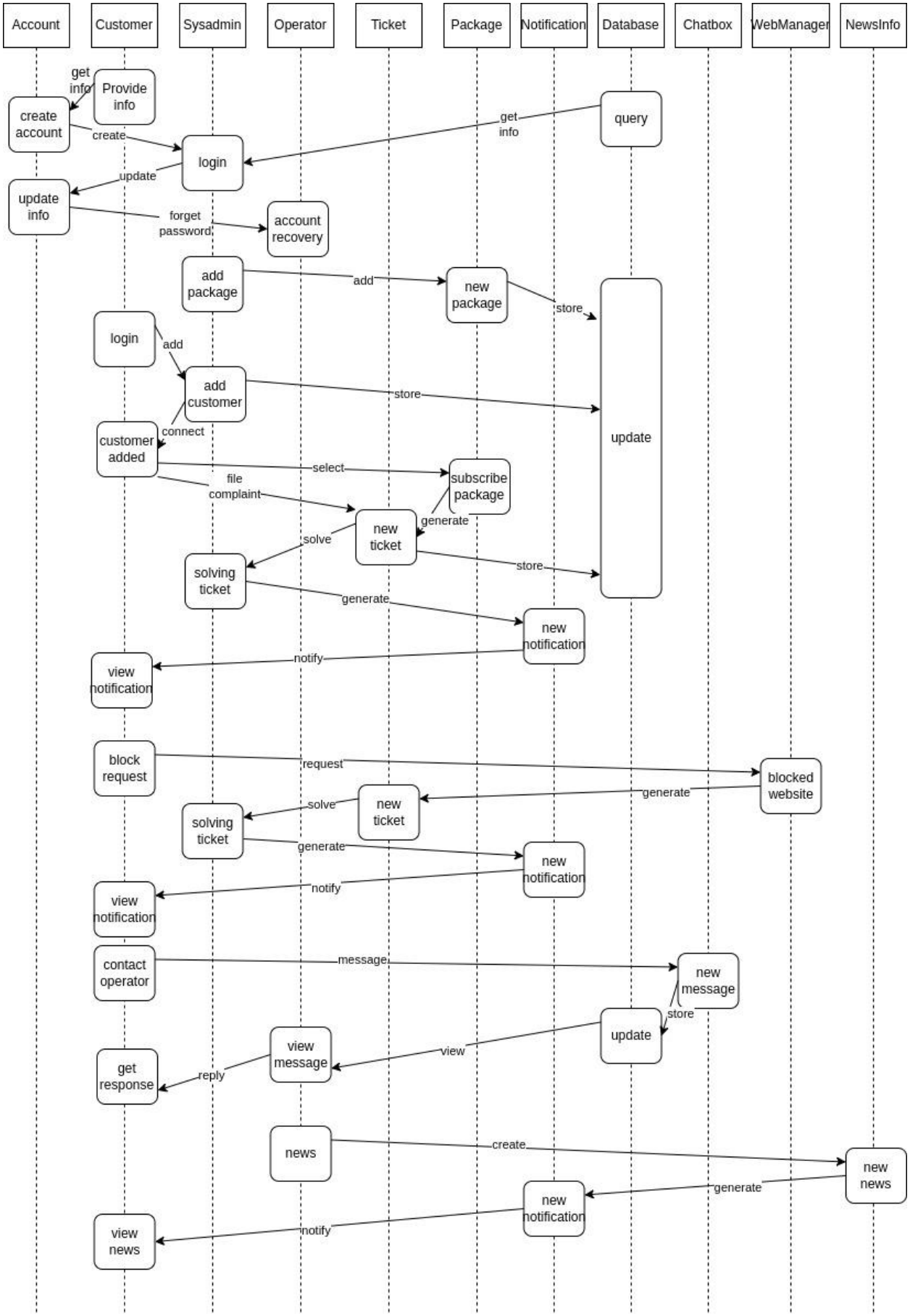


7.1.2.1 ID:8
Name :Chatbox



7.2 Sequence Diagram

Another type of representation in behavior modeling is Sequence Diagrams.


Account
Customer
Sysadmin
Operator
Ticket
Package
Notification
Database
Chatbox
WebManager
NewsInfo
get info
Provide info
create account
create
login
query
get info
update
update info
forget password
account recovery
add package
add
new package
store
update
login
add
add customer
store
connect
customer added
select
subscribe package
file complaint
generate
new ticket
solve
solving ticket
store
generate
new notification
notify
view notification
block request
request
blocked website
generate
solve
new ticket
solving ticket
generate
new notification
notify
view notification
contact operator
message
new message
store
update
view
view message
reply
get response
news
create
new news
generate
new notification
notify
view news

CHAPTER 8

CONCLUSION

We are pleased to submit the final SRS report on this project. Throughout the project we have maintained our QFD. Our srs documentation reflects the stakeholders points of view. We have tried our best to avoid duplication or repetition of processes, data or steps.It will be very easy to conduct the whole project using this SRS.