TRADING OF OLD MOTOR VEHICLES

Software Requirement Specification

Trading Of Old Motor Vehicles

Submitted to

Mohammed Shafiul Alam Khan

SPL-II Coordinator

Institute of Information Technology

University of Dhaka

Submitted by

Ibrahim Khalil	BSSE0804
Imam Hossain Kawsar	BSSE0816

Group ORCHID

5th Semester, BSSE 8th Batch

Institute of Information Technology

University of Dhaka

Submission Date: 20 March 2018

Letter of Transmittal

Mohammed Shafiul Alam Khan

SPL-II Coordinator

Institute of Information Technology,

University of Dhaka.

Subject: Submission of term report on "Trading Of Old Motor Vehicles".

Sir,

With due respect, we are submitting the report on the above topic assigned us for SPL-II. 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,

Ibrahim Khalil 0804

Imam Hossain Kawsar 0816

3rd year, 5th semester, 8th batch

Institute of Information Technology

University of Dhaka

Session: 2015-16

Acknowledgement

We are highly indebted for getting such a tremendous opportunity to prepare the report on Trading Of Old Motor Vehicles. We would like to thank whole-heartedly our supervisor, Nadia Nahar, Lecturer, Institute of Information Technology, University of Dhaka, for giving us guidelines about how we can prepare this report. In completing this report, we have collected various important data and information from various cars shops. We are thankful to all of the works cited.

Abstract

This study is made for understanding the online trade of old motor vehicles. The scope of the study is to analyze the trade of old motor vehicles and to know the functions and drawbacks and design the SRS (software requirements and specification) of the system. The object of the study is to develop an SRS of trading of old motor vehicles. This study also describes the current system of the cars shops online system.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter is a part of the software requirement specification for the project 'Trading of Old Motor Vehicles'. In this chapter intended audience for the project are focused on.

1.1 Purpose

This document briefly describes the Software Requirement Analysis of online system of selling and buying old motor vehicles. It contains functional, non-functional, and supporting requirements and establishes a requirement baseline for the development of the system. The requirements specified in the SRS are independent, uniquely numbered and organized by topic. The SRS serves as an official mean 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 in intended for several audiences including the customers as well as the project managers, designers, developers and testers.

- The customers will use this SRS to verify that the developer team has created a product that is acceptable to the customers.
- This 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 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 a software that will fulfill all 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

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

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:

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

2.1.1 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 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?”

The following stakeholders were identified for the Trading of Old Motor Vehicles system.

Shop Owner: Shop Owner owns the shop and operates the business. If he works alone then, he does not demand day to day transaction.

Employee: Employee is an individual who works part-time or full-time under a contract of employment, whether oral or written, express or implied, and has recognized rights and duties. He carries out the instructions of the owner. After the day, he needs to clear out the transactions.

Customer: Customer is an individual who buys or sometimes sell his/her motor vehicles. He/she can view the product and comment for any product. He/she can be authenticated by the system.

Dealer: In this system, anyone can act as a dealer. To be a dealer, an account in the system is required. Only shop owner can view the dealer's product list until the confirmation about the product.

Software Developer: Software developer also stakeholders. They develop and maintain the whole system.

2.1.2 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.

Shop Owner's viewpoint

- Error free system
- Limited budget
- Strong authentication system
- Keeping all transaction records
- Easy to use
- Keeping a backup of the data

Employee's viewpoint

- User friendly
- Error free system
- Getting accurate information of total received cash and profit of a day
- Being notified about the pre-ordered vehicles
- Strong authentication system

Dealer's viewpoint

- Easy to send car information
- Notify the shop owner about the sending car for the sell
- Ranking system should be maintained

Customers' viewpoint

- Getting a receipt of the purchase
- View the correct information about the motor vehicles
- Warranty for the maintenance

Developer's viewpoint

- Easy to built
- Error free effective software
- No ambiguous requirement
- Getting a decent amount of money for project budget

2.1.3 Working towards collaboration

While working with different stakeholders, some conflicting and common viewpoints can be noticed. For this reason, final requirements can be gotten by collaborating the viewpoints.

Common viewpoints

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

Conflicting viewpoints

- Developing the project in minimum budget
- Notify the shop owner about the car
- Warranty for the maintain

Final Requirements

- Error free effective system
- User friendly
- Easy to maintain the software
- Strong authentication system
- Keeping a data backup
- Keeping a well manageable transaction record

2.1.4 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 to know their demands. The questionnaires are included in the appendix section.

2.2 Conclusion

In this inception phase, a basic understanding of the problem was developed and a preliminary nature of the solution was obtained. The requirements which are identified in this phase, will be used later for further steps of requirement engineering.

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-

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

3.2.1 Collaborative Requirements Gathering

The meetings with the stakeholders created an indecisive state to elicit the requirements. To solve this problem, more than one meeting was held with the stakeholders. A slightly different scenario from these approaches has been found following activities have been completed to accomplish this task.

- The meetings were conducted with the shop owner, employees and suppliers. They were questioned about their requirements and expectations from the Grocery Management System.
- They were asked about the problems they were facing with the current manual system.
- Lastly final requirement list was selected from the meetings.

3.2.2 Quality Function Deployment

Quality Function Deployment (QFD) is a technique that translates the needs of the customer into technical requirements for software. Ultimately the goal of QFD is to translate subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology that concentrates on maximizing customer satisfaction from the software engineering process. The requirements, which are given below, are identified successfully by the QFD.

3.2.2.1 Normal Requirements

The 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 the project.

- Getting information about daily, weekly, monthly and yearly transaction.
- Storing information regarding equipment and other product maintenance
- Storing profit records
- Using limited budget for making the software
- Keeping track of remaining cars and bikes
- Storing accurate records of transactions
- Identifying profitable and unpopular products
- Get notified about cars and bikes fulfill some condition

3.2.2.2 Expected Requirements

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described.

- Error free software
- Strong authentication system
- User friendly
- Effective system
- No ambiguous feature
- Data back up

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 this project.

- Customers can contact shop owner directly through mail and text messages
- User will be notified through mail and text message when their prefer cars or bikes are available.

3.2.3 Usage scenario

Authentication

Sign Up

In the TOMVS, there is an authentication part, where it allows the user to access the secure part of the system. Shop owners, employees, suppliers and customers are the users of this system. To access some part of the system or perform some task, a user requires an account and for that he/she must fill up a form. Users have to give first name, last name, email id, address, mobile number, password, confirm password and account type (types control the access to the system). User can fill either email id or

mobile number or both. No user name can contain any number, punctuation mark or any special character and the length of the name should be between 2 characters and 30 characters. There will be duplicity and validity (syntax) checking for email and mobile number. If all the information is correct, user will be sent a verification code to his/her email address (if he/she provides only email address) or mobile number (if he/she provides only mobile number) or both (if he/she provides both email id and mobile number). If the account type is "Employee", "Supplier" or "Owner", he/she will have to wait for any of the current owner's approval. There will be one preinstalled account as "Owner" type.

Sign In

If any user has an account, he/she can log in to the system. To log in, user has to give his/her email id or mobile number and password. The email id or mobile number and password will be verified. If the verification is successful, the user can log in to the system successfully. If the password is wrong, there is a retry option. If the retry count is 3, the user will be blocked for 30 seconds. After block time has finished user can attempt to log in to the system again. User has an option to logout from the system.

Recover Password

User can recover his/her password. To recover password, he/she has to give his/her email id or mobile number that he/she has used to sign up. If the provided email id or mobile number is valid, his/her old password will be send to his/her email address or mobile number.

Search

There are three kinds of search in the TOMVS: User search, Product search and Transactional search.

User search

If user is an owner or employee, he/she can search owner information, employee information, supplier information and customer information. If

user is a supplier or customer, he/she cannot search another supplier or customer information but only owner or employee information.

To search a specific type of user, username has to be given. If username (first name, last name) is valid, all stored information of entered username will show. After that, user can also be searched by using user id. If user Id is valid, all stored information of that id will show.

Product search

Purpose of this part is to show a product is exist or not in the shop. All kind of user can search product. To search product user has to give product name (can be substring). If product name is exist, all stored information of entered product will show. After that, product information can be searched by using product id. If product id is valid, all stored information of entered product id information will show.

Transactional search

Only owner and employee can perform transactional search. For every transaction type, there will be a search option. To search any transaction, user has to give transaction type. If type exists, all stored information of this type will show. After that, user can search using transaction id. If the transaction id is valid, all stored information of the entered transaction id will be shown.

Order and maintenance

Existing Product order

After logged in to the system, user can order to any products through the website interface. When s/he orders, all the information of that product will be added to the ordered list. System must want to know from the customer the date by when s/he wants to buy that product. If user doesn’t take any action within this period, order will be cancelled automatically. A product can be ordered by multiple customers. But first customer gets the higher priority. If the first customer cancels the order or doesn’t effort to buy

within the date, second customer will get the chance to buy. The process will continue for all users who orders later.

Special Order

User can desire to have a special configured car/bike that is not available in the shop on that time. So, system has an option for the user to order of any car/bike. To achieve this, user has to describe her/his desired product information, that is brand name, model, original buying year, fuel, engine, edition of the model, condition of the product, mileage price condition, color, features (exterior, interior, equipment) and picture of the product. If any time product is available on the shop for sale, that user will be notified.

Vehicle Management

Vehicle management is divided into two subsystems. One is for car management and another is for motorbike management.

Car Management

Shop owner can include and exclude car. If a new car is added to the car list, the description of the car: brand name, model, originalBuyingYear, fuel, engine, editionOfTheModel, condiotionOfTheCar, mileage, price, priceCondiotion, color, features (exterior, interior, equipment) and pictureOfTheCar are the fields needed to be filled out by the shop owner. System then generates a car ID for each car. When a car is added to the advertisement section by the supplier, s/he has to fill out all these fields. User has an opportunity to provide review about a product. All the products are shown on the basis of review popularity.

Motorbike Management

Shop owner can include and exclude bike. If a new bike is added to the bike list, the description of the bike: brand name, model, originalBuyingYear, fuel, engine, editionOfTheModel, condiotionOfThebike, mileage, price, priceCondiotion, color, features (exterior, interior, equipment) and pictureOfThebike are the fields needed to be filled out by the shop owner. System then generates a bike ID for each car. When a bike

is added to the advertisement section by the supplier, s/he has to fill out all these fields. User has an opportunity to provide review about a product. All the products are shown on the basis of review popularity.

Notification

User might be notified for several reasons.

Ordered product is available

User can make a pre-order of products that is not available in the shop on that time. So, when the time his/her ordered product is available, s/he will be notified instantly.

Discount Time

Shop Owner can offer a discount to all the users occasionally or for the purpose of business. This discount message is sent to all the users through notification on their account.

Transactional Report

After completing a transaction, user will be notified the transactional information on their account.

Maintenance Log

System provides an opportunity to the user to report to the system if they face with any problem of using product before expiring their warranty date.

When customer is eligible to get the free servicing, s/he will be requested by the owner through a notification. After recovering they problem, system again sends a notification to their account.

Advertisement

User can advertise their product on the website. When s/he upload all information about her/his product, a notification is sent to the shop owner. Besides, if shop owner chooses her/his product to buy, a notification will be sent to his account.

Equipment Management

There is a subsystem for the equipment of car and motorbike. Shop owner can add an equipment in the equipment list. To add an equipment: name, price, quantity, unit price, date, time are stored.

Transaction

Transaction is one of the most important part of the TOMVS. In this subsystem, any kind of transaction will be recorded. Transaction can be two types e.g. sell and buy. Three kinds of product can be sell or buy. They are car, motor-bike and equipment. For any transaction some information has to be stored. They are: transaction id, transaction type(buy or sell), product type(car, motor-bike or equipment), date, time, client name(can be supplier or buyer). Transaction can searched by transaction id, transaction type, date range.

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 statement of our requirements for the Trading of Old Motor Vehicles system Making a bounded statement of scope for our system.
- Making a list of customers, users and other stakeholders who participated in the requirements elicitation.
- Making 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

This chapter describes the scenario based model for the Trading of Old Motor Vehicles System.

4.1 Introduction

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If the software developer team understands how end users (and other actors) want to interact with a system, they will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios 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 the system operators. Every user has one or more goals when using system.

Primary Actor

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

Secondary Actor

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

4.3 Use Case Diagrams

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

4.3.1 Level 0 - TOMVS

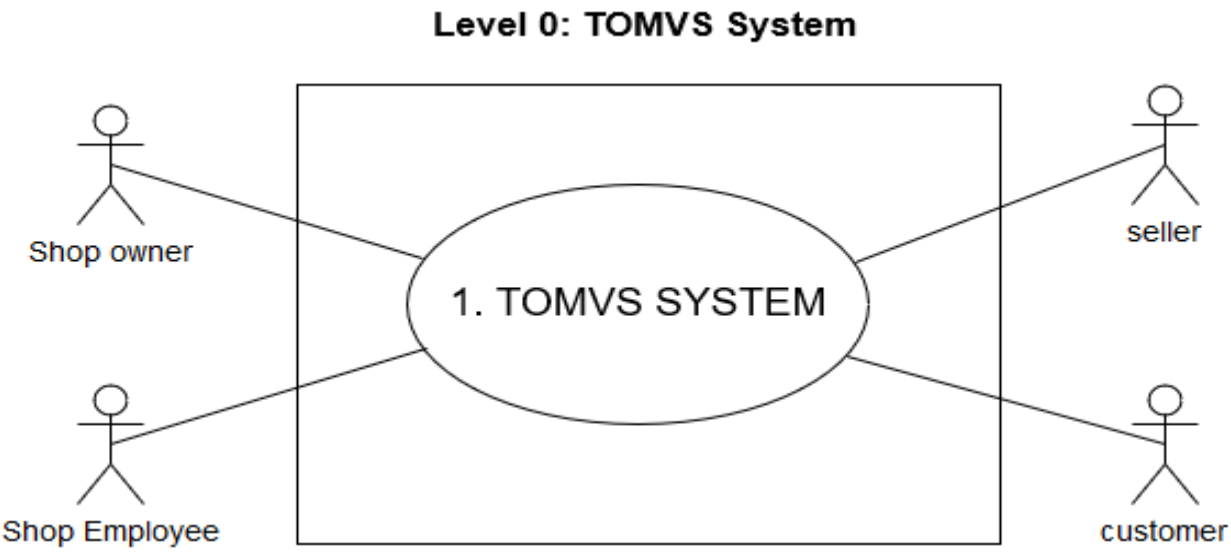


Figure 1: Level 0 -Trading of Old Motor Vehicle System

Actors

1. Shop Owner
2. Employee
3. seller
4. Customer

Description

There are four actors in this system who will use the system directly. Primary actors are those who will play action and get replies from the system. Secondary actors only produce or consume information.

4.3.2 Level 1– Subsystems

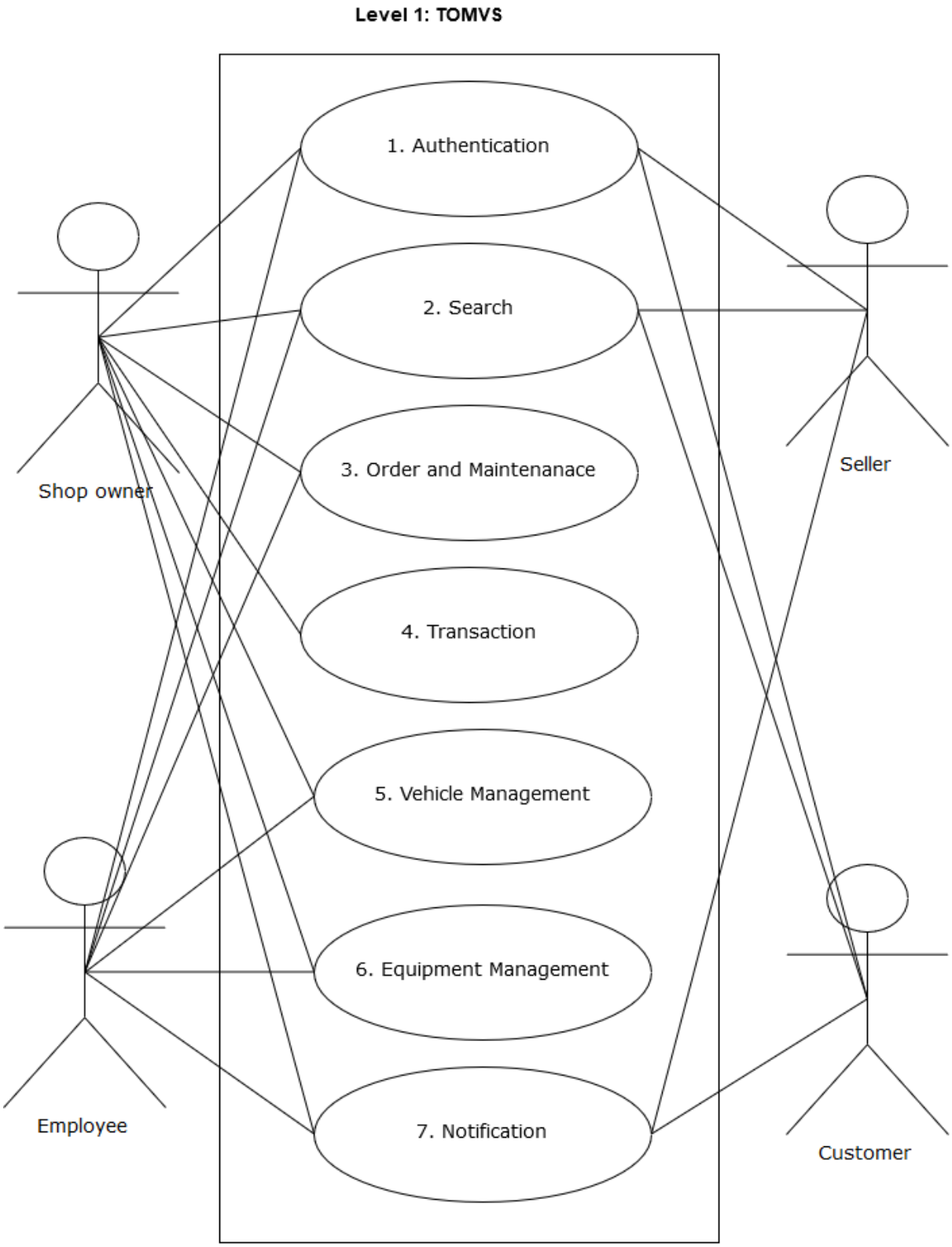


Figure 2: Level 1 - Subsystems

Actors

1. Shop Owner
2. Employee
3. Customer
4. Seller

Description

There are seven subsystems in the Trading of Old Motor Vehicle System. They are:

1. Authentication
2. Search
3. Order and Maintenance
4. Transaction
5. Vehicle Management
6. Equipment Management
7. Notification

4.3.3 Level 1.1 – Authentication

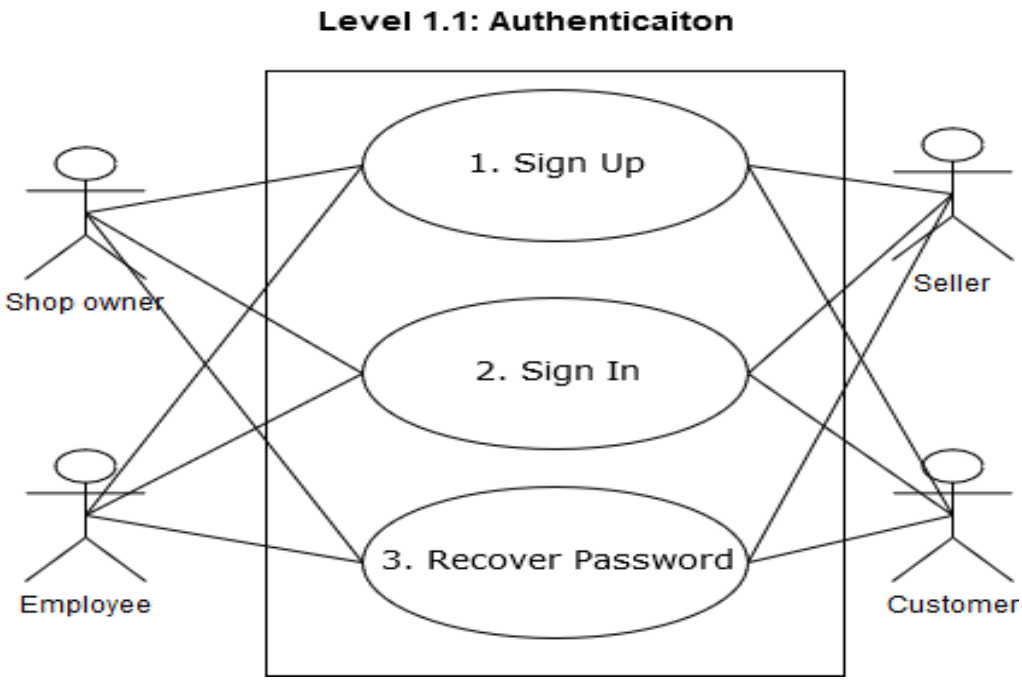


Figure 3: Level 1.1 - Authentication

Actor

1. Shop Owner
2. Employee
3. Seller
4. Customer

Description

Sign up: While signing up, the user has to give her/his first name, last name, email Id (checking syntax and duplicity), address, phone number (validity check, duplicity check), password (Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters), confirm password, account type (types control the access to the system). If the given information is all correct, it is stored. If the account type is "Employee" s/he needs to wait for the owner's approval. Thus the users' account is created.

Sign In: Those who are registered can log in to the system. In order to sign in, the user has to give his/her email id (exist in the storage) and password. If the password is wrong, there is a retry option. If her/his attempt goes failed for three consecutive time, the user will be blocked for 30 seconds. After block time has finished, the user again gets the chance to attempt to log in to the system.

Recover Password: User can recover his/her password. While recovering password, s/he has to give her/his email id. If email id is valid, a recovery password will be sent to her/his email address.

If any user wants to change her/his information, there is an option where s/he can update her/his information except the account type. User has an option to log out from the system.

Action reply

Sign up

Action: User enters information (first name, last name, email id, phone number, password, confirm password, address, account type).

Reply: System checks validity of information. If the entered information is correct, information will be stored.

Sign in

Action: User enter email id and password.

Reply: System checks validity. If valid, user can successfully sign in. If not valid, system count users' number of retry and take action as blocking the user account for 30 seconds.

Action: User can choose recover password option.

Reply: System allows user to recover password.

Recover password

Action: User enters the email id to send his/her password.

Reply: System checks validity of entered information. If valid, user's password will be sent to email id.

4.3.4 Level 1.2 – Search

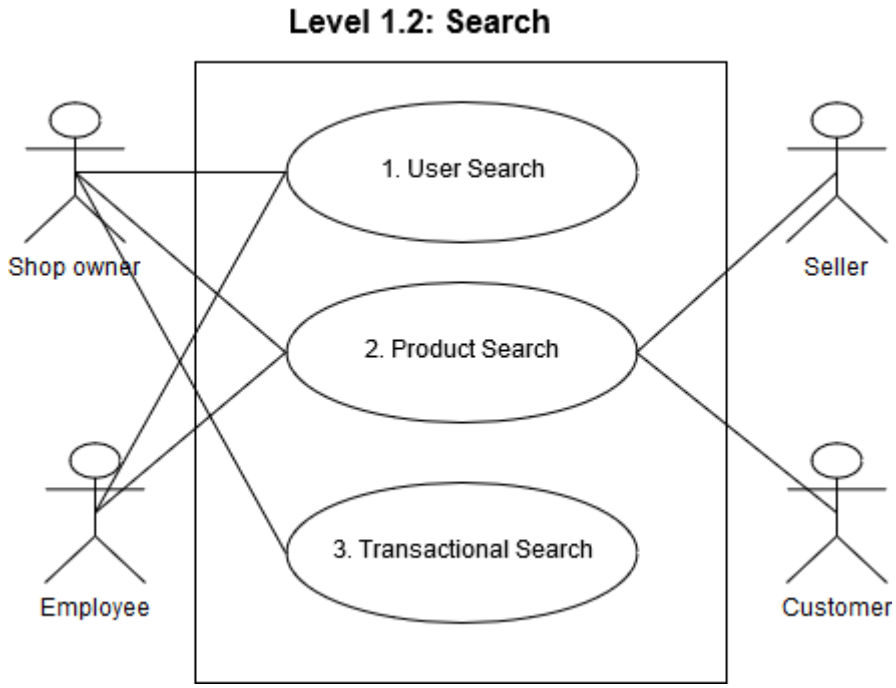


Figure 4: Level 1.2 - Search

Actor

1. Shop Owner
2. Employee
3. Seller
4. Customer

Description

There are three kinds of search: user search, product search and transactional search.

User search

Shop owner can search for users (employee, customer) with the **name** to see the information. All the information of that specific user is shown then. If there are more users with the same name, a second choice must need to make someone specific. Employee can search his/her information and customer information.

Product search

All the actors can search product. In order to search product, type of product has to be selected. Type must be car, motorbike or equipment. To search specific product, product name has to be provided. If product exists, all information of entered product name will be showed.

Transactional search

Shop owner and employee can see all the transactional information. To search transaction, transaction type(sell or buy) has to be specified. Then using transaction id, all information of that transaction can be showed.

Action – Reply

User search

Action: User enters another user name.

Reply: System checks the availability of the user. If exist, show all information of entered product.

Product search

Action: User enters the product type

Reply: System checks the availability of the product type.

Action: User enters product name.

Reply: System show all information of entered product name if product name is available in the system.

Transactional search

Action: User enters type of transactional search

Reply: System shows all transaction of given type.

Action: User enters specific transaction id of given type.

Reply: System shows all information of entered transaction id.

4.3.5 Level 1.3 - Order and Maintenance

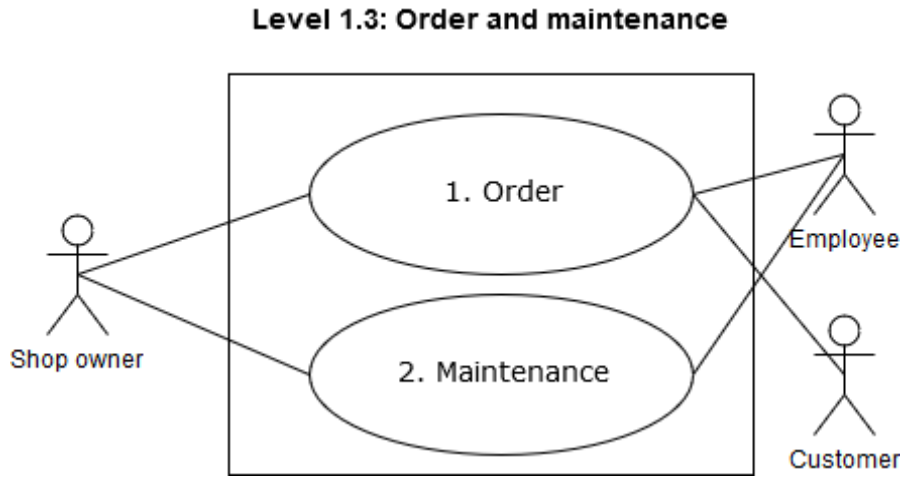


Figure 5: Level 1.3 - Order and Maintenance

Actors

1. Shop Owner
2. Employee
3. Customer

Description

There are two parts of this subsystem. One is Order and another one is Maintenance.

Order: Order is divided into two part.

Maintenance: System confirms a feature to the user to report their products, user can report to the system to get repairmen of their product before expiring warranty date. User has to provide her/his product id. System checks it's expiring date. If this product has been bought before expiring the warranty date, user will get the interface to report her/his product. User has to describe the problem welly. Shop owner then response to user whether the product is recoverable or not. If recoverable, s/he will be requested to take the service for free. Customer submit his/her problem through an interface. Customer give buyername, dateOfPurchase, product type, model, brand, problem description in the interface.

Action - Reply

Action: fill up a form

Reply: send notification

4.3.5.1 Level 1.3.1 – Order

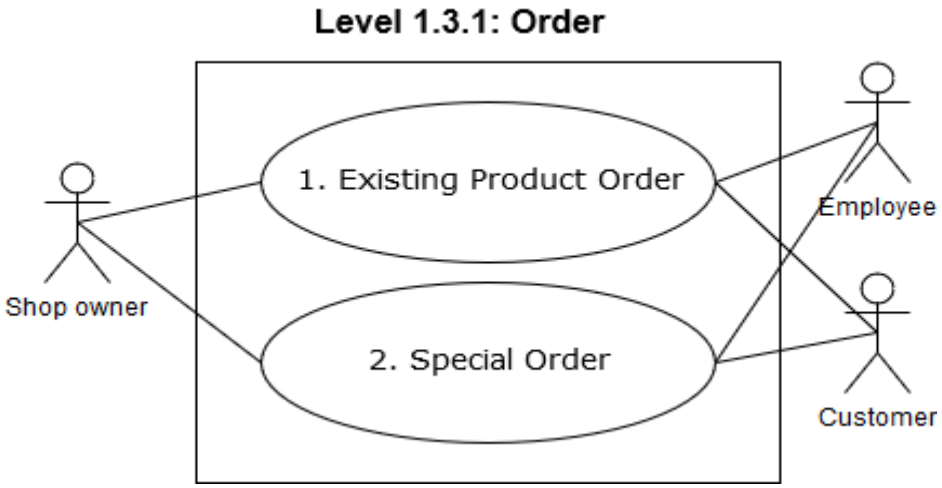


Figure 6: Level 1.3.1 - Order

Description

Existing Product Order

After logged in to the system, user can order to any products through the website interface. When s/he orders, all the information of that product will be added to the ordered list. System must want to know from the customer the date by when s/he wants to buy that product. To order any product certain percentage of amount have to pay. Paid amount information will be stored in the transaction table.

Special Order

User can desire to have a special configured car/bike that is not available in the shop on that time. So, system has an option for the user to order of any car/bike. To achieve this, user has to describe her/his desired product information, that is brand name, model, original buying year, fuel, engine, edition of the model, condition of the product, speed, price condition, color, features (exterior, interior, equipment) and image of the product. If any time product is available on the shop for sale, that user will be notified.

Action - Reply

Existing Product Order

Action:Customer chose the product and give informaion through an interface.
Reply: A confirmation message will be sent to customer

Special Order

Action: Customer give some information of his/her desire product.
Reply: Customer will notify, if the product is available.

4.3.6 Level 1.4 – Transaction

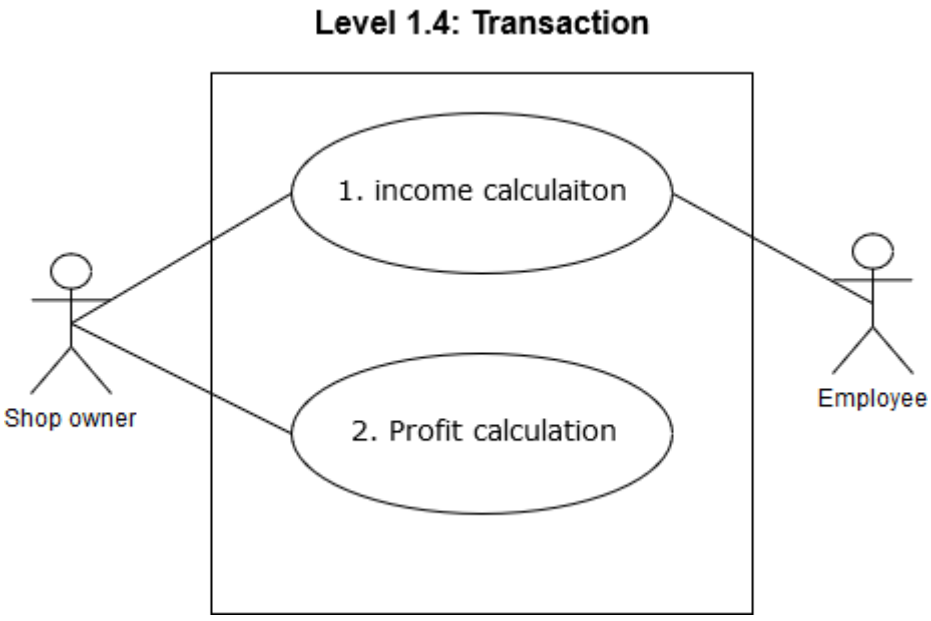


Figure 7: Level 1.4 - Transaction

Actors

1. Shop Owner
2. Employee

Description

Income Calculation: To calculate income, data from the transaction table will be used. All the data from income type will be added to calculate income

of the shop. For any transaction some information has to be stored. They are: transaction_id, transaction_type, customer_id, customer_type, product_type(car, motor-bike or equipment), product_id, date, time, paid_amount, rest_amount, seller_id.

Profit Calculation: Here, system calculate profit only for the input data. To calculate profit, system needs income and expenditure amount which comes from transaction data.

Action-Reply

Income Calculation

Action: User enters information of product sell.

Reply: Entered product sell information will be stored.

Profit calculation

Action: user wants to calculate total profit.

Reply: System calculate total revenue and total expenditure from transaction data. From these information, total profit will be calculated.

4.3.7 Level 1.5 – Vehicle Management

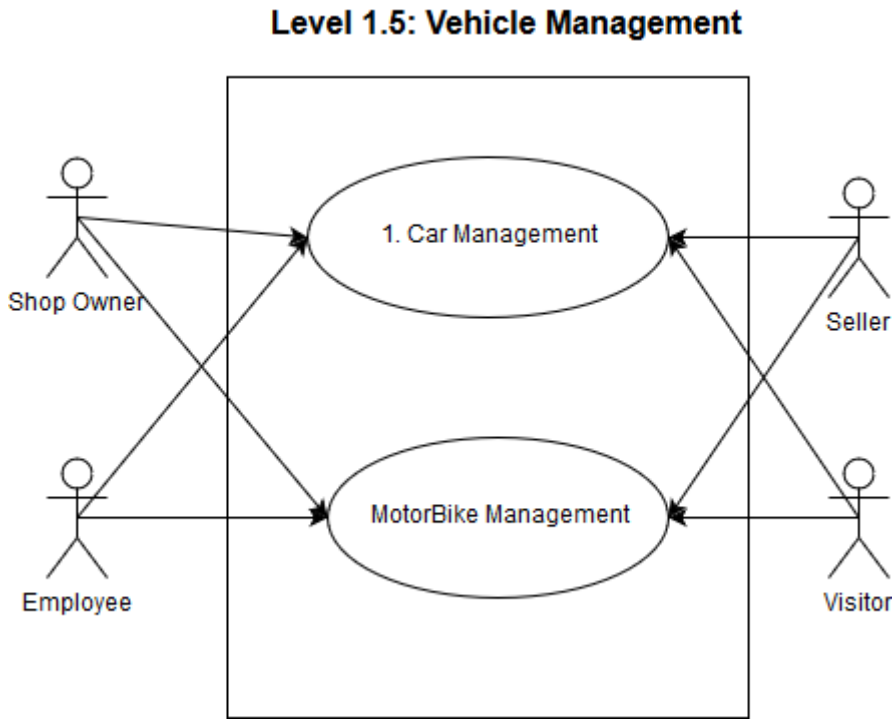


Figure 8: Level 1.5 - Vehicle Management

Actors

1. Shop Owner
2. Employee
3. Seller
4. Visitor

Description

Vehicle Management consists of car management and motorbike management subsystem.

4.3.7.1 Level 1.5.1 – Car Management

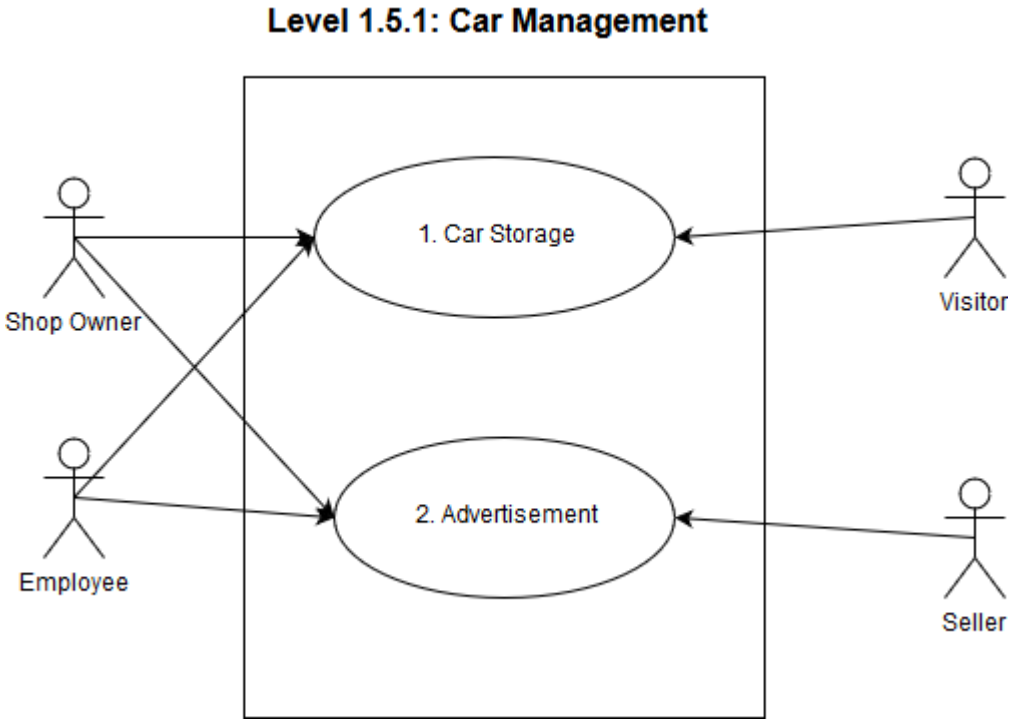


Figure 9: Level 1.5.1 - Car Management

Actors

1. Shop owner
2. Employee
3. Seller
4. Visitor

Description

Car Management consists of Car Storage and Advertisement subsystem.

4.3.7.1.1 Level 1.5.1.1 – Car Storage

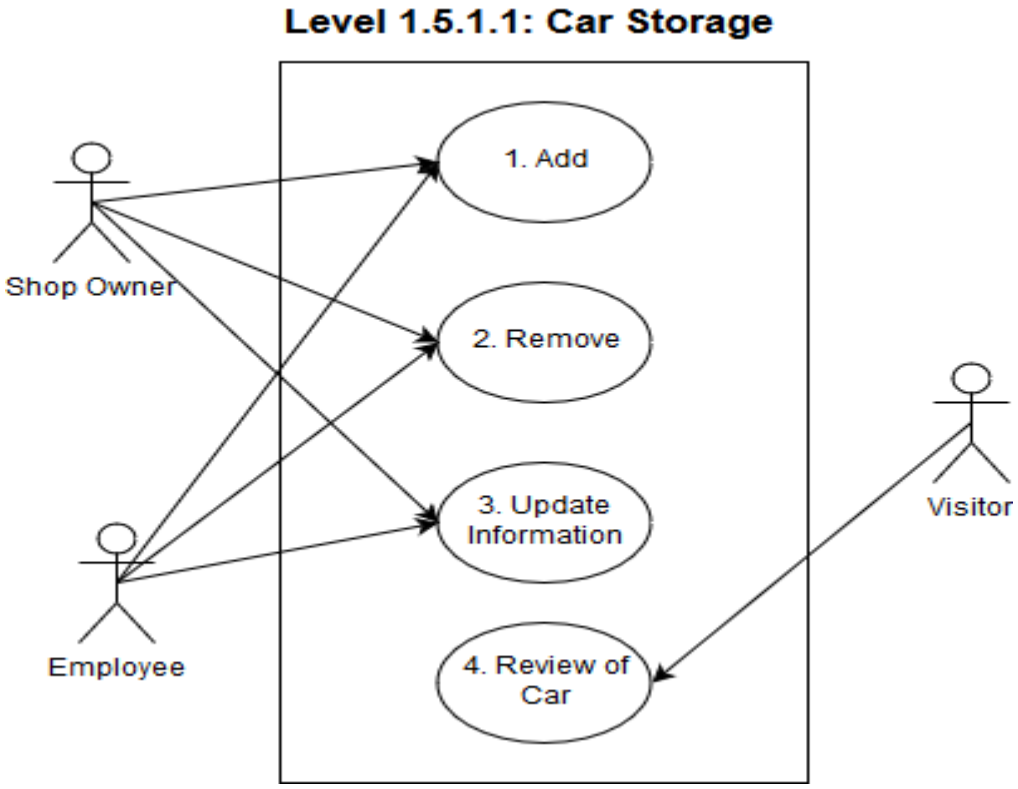


Figure 10: Level 1.5.1.1 - Car Storage

Actors

1. Shop Owner
2. Employee
3. Visitor

Description

Car Storage: Shop owner and employee can add, remove and update information of any car. Visitor can like and comment about any car that will count as a review of the car.

Action - Reply

Action: want to add car

Reply: allow to add car

Action: want to remove car

Reply: allow to remove car

Action: want to update information

Reply: allow to update information

Action: want to review

Reply: allow to review

4.3.7.1.2 Level 1.5.1.2 – Advertisement

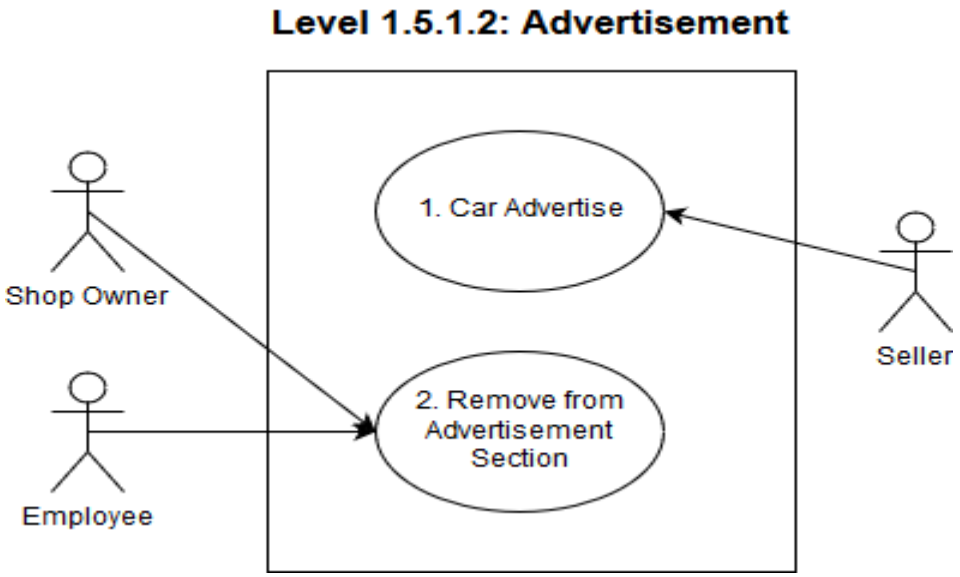


Figure 11: Level 1.5.1.2 - Advertisement

Actors

1. Shop Owner
2. Seller
3. Employee

Description

Advertisement: Shop owner and Employee can manage advertisement product. Seller can advertise his/her car through an acceptance of shop owner.

Action - Reply

Action: Want to accept car

Reply: Allow to accept car

Action: Want to advertise car

Reply: Allow to advertise car

4.3.7.2 Level 1.5.2 – MotorBike Management

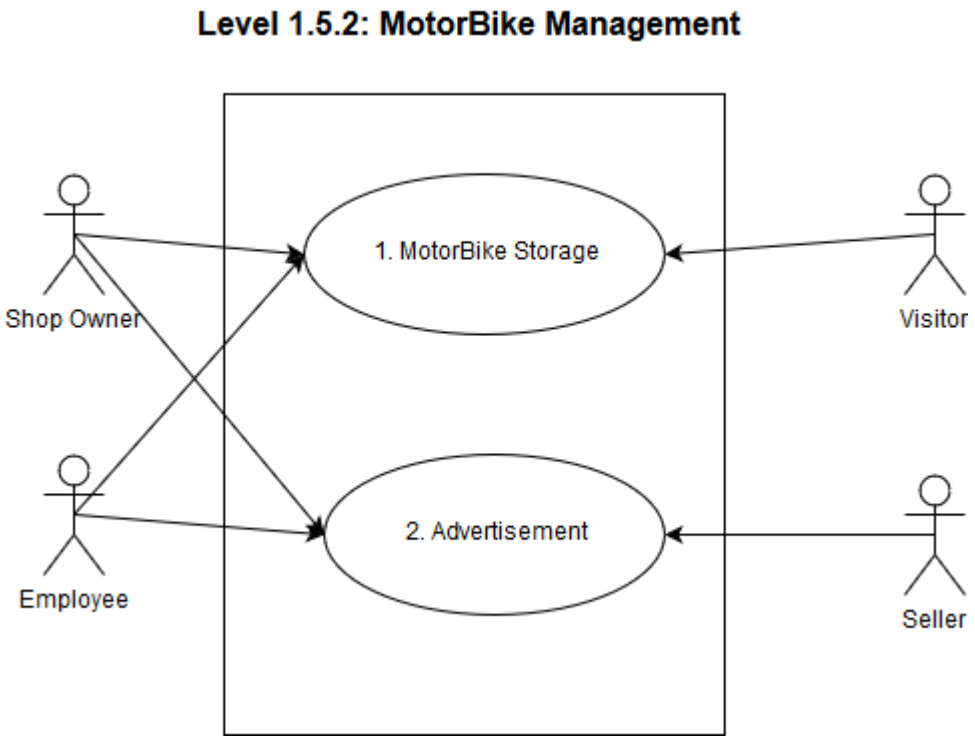


Figure 12: Level 1.5.2 - MotorBike Management

Actors

1. Shop Owner
2. Employee
3. Visitor
4. Seller

Description

MotorBike Management consists of MotorBike Storage and Advertisement subsystem.

4.3.7.2.1 Level 1.5.2.1 – MotorBike Storage

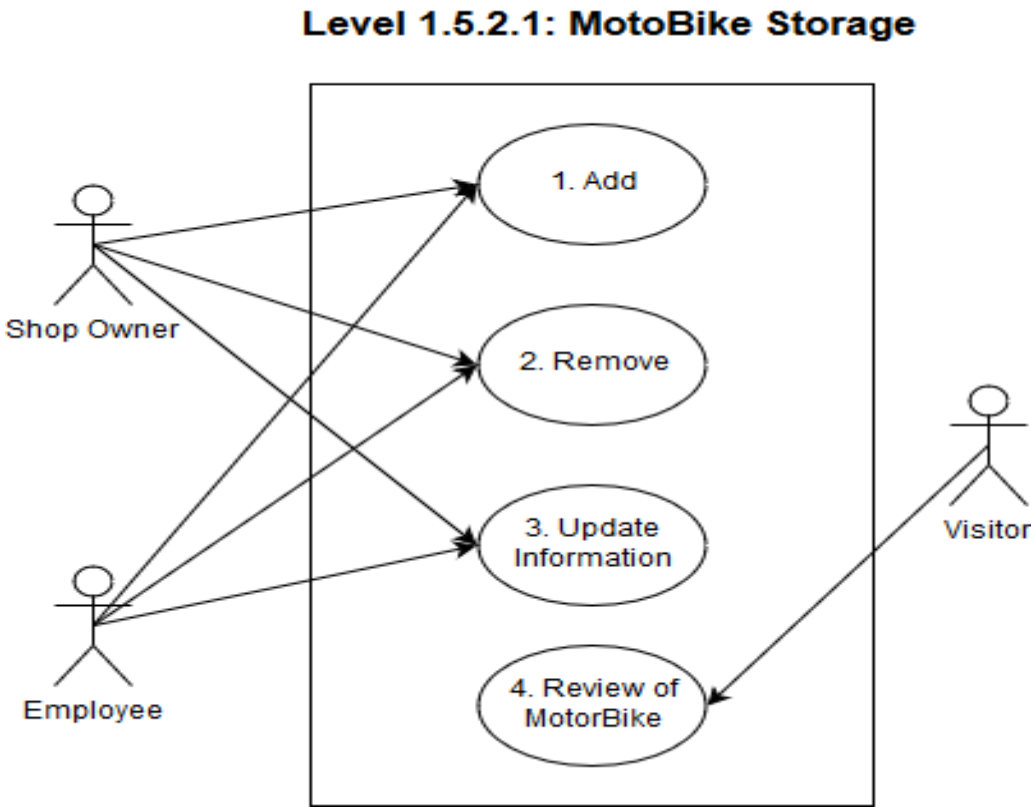


Figure 13: Level 1.5.2.1 - MotorBike Storage

Actors

1. Shop Owner
2. Employee
3. Visitor

Description

MotorBike Storage: Shop owner and employee can add, remove and update information of any bike. Visitor can like and comment about any bike that will count as a review of the bike.

Action - Reply

Action: want to add bike

Reply: allow to add bike

Action: want to remove bike

Reply: allow to remove bike

Action: want to update information

Reply: allow to update information

Action: want to review

Reply: allow to review

4.3.7.2.2 Level 1.5.2.2 – Advertisement

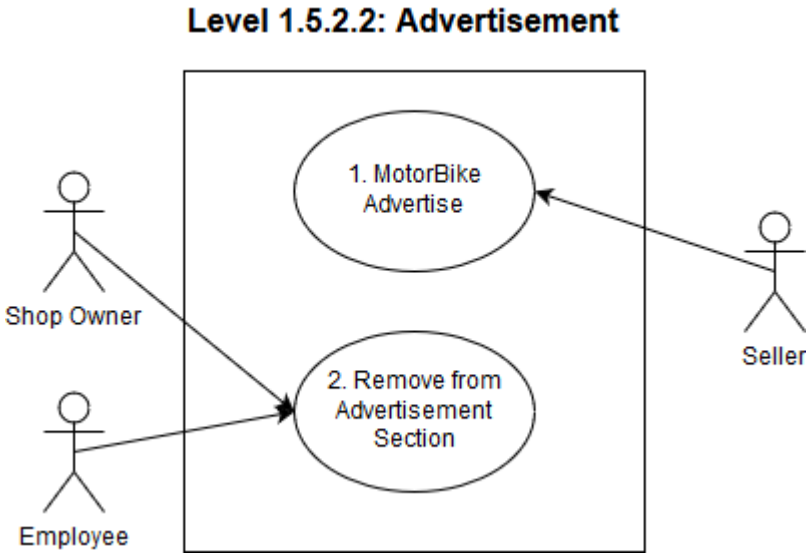


Figure 14: Level 1.5.2.2 - Advertisement

Actors

1. Shop Owner
2. Employee
3. Seller

Description

Advertisement: Shop owner and Employee can manage advertisement product. Seller can advertise his/her bike through an acceptance of shop owner.

Action - Reply

Action: Want to accept bike

Reply: Allow to accept bike

Action: Want to advertise bike

Reply: Allow to advertise bike

4.3.8 Level 1.6 – Equipment Management

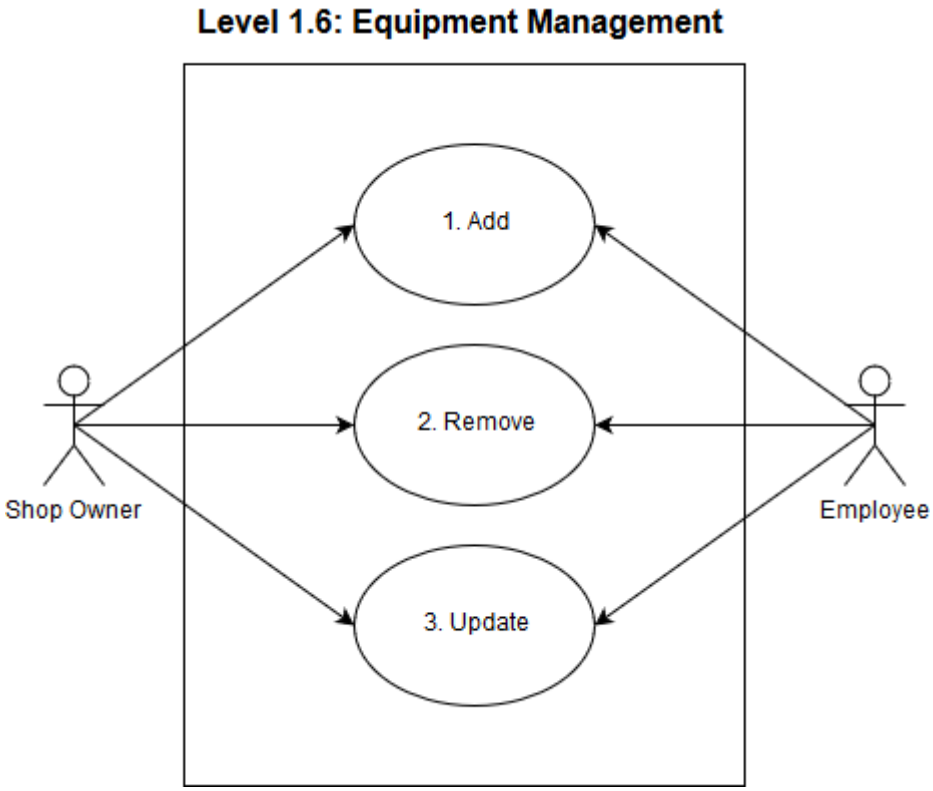


Figure 15: Level 1.6 - Equipment Management

Actors

1. Shop Owner
2. Employee

Description

Shop Owner and Employee can add, remove and update equipment.

Action - Reply

Action: Want to add equipment

Reply: Allow to add

Action: Want to remove equipment

Reply: Allow to remove

Action: Want to update equipment information

Reply: Allow to update information

4.3.9 Level 1.7 – Notification

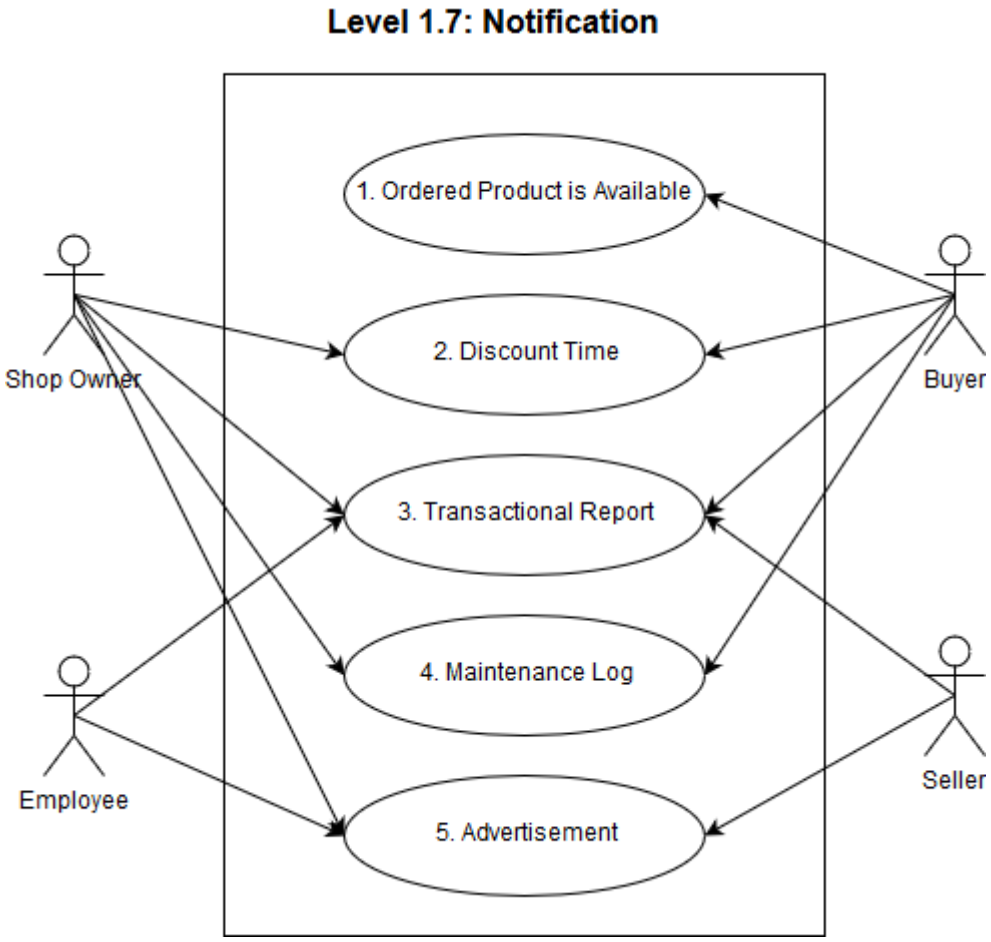


Figure 16: Level 1.6 - Notification

Actors

1. Shop Owner
2. Employee
3. Buyer/Customer
4. Seller/ Dealer

Description

Ordered product is available

User can make a pre-order of products that is not available in the shop on that time. So, when the time his/her ordered product is available, s/he will be notified instantly.

Discount Time

Shop Owner can offer a discount to all the users occasionally or for the purpose of business. This discount message is sent to all the users through notification on their account.

Transactional Report

After completing a transaction, user will be notified the transactional information on their account.

Maintenance Log

System provides an opportunity to the user to report to the system if they face with any problem of using product before expiring their warranty date. When customer is eligible to get the free servicing, s/he will be requested by the owner through a notification. After recovering they problem, system again sends a notification to their account.

Advertisement

User can advertise their product on the website. When s/he upload all information about her/his product, a notification is sent to the shop owner. Besides, if shop owner chooses her/his product to buy, a notification will be sent to his account.

Action - Reply

Action: Want to get notification

Reply: Send notification

4.4 Activity Diagram

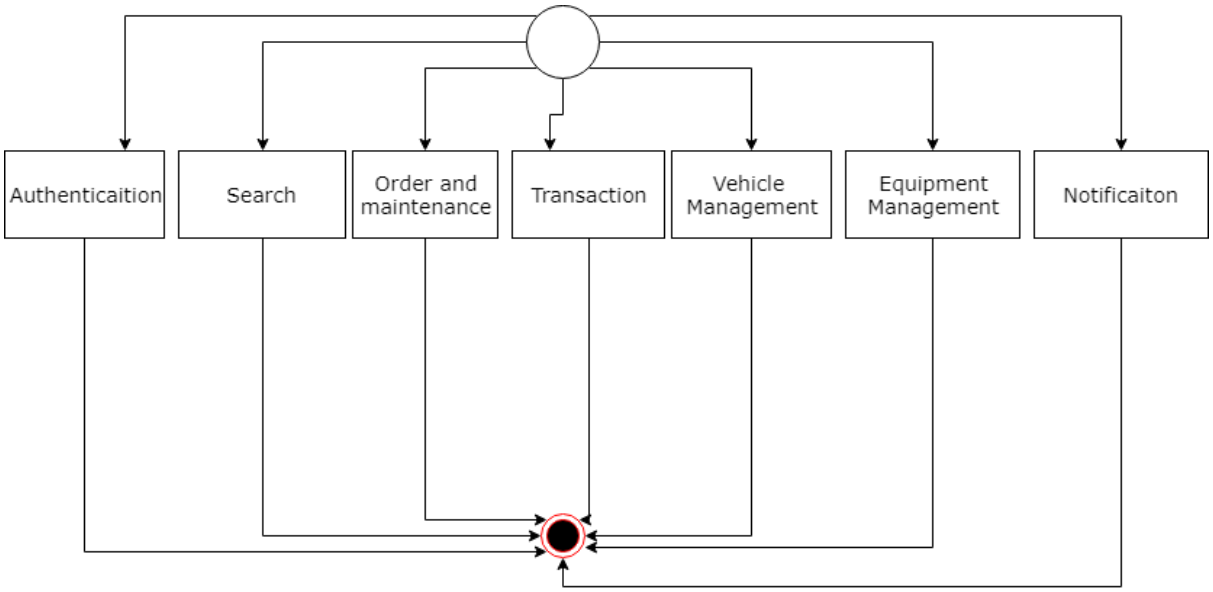


Figure 16: Activity Diagram 1 - Trading of Old Motor Vehicle System

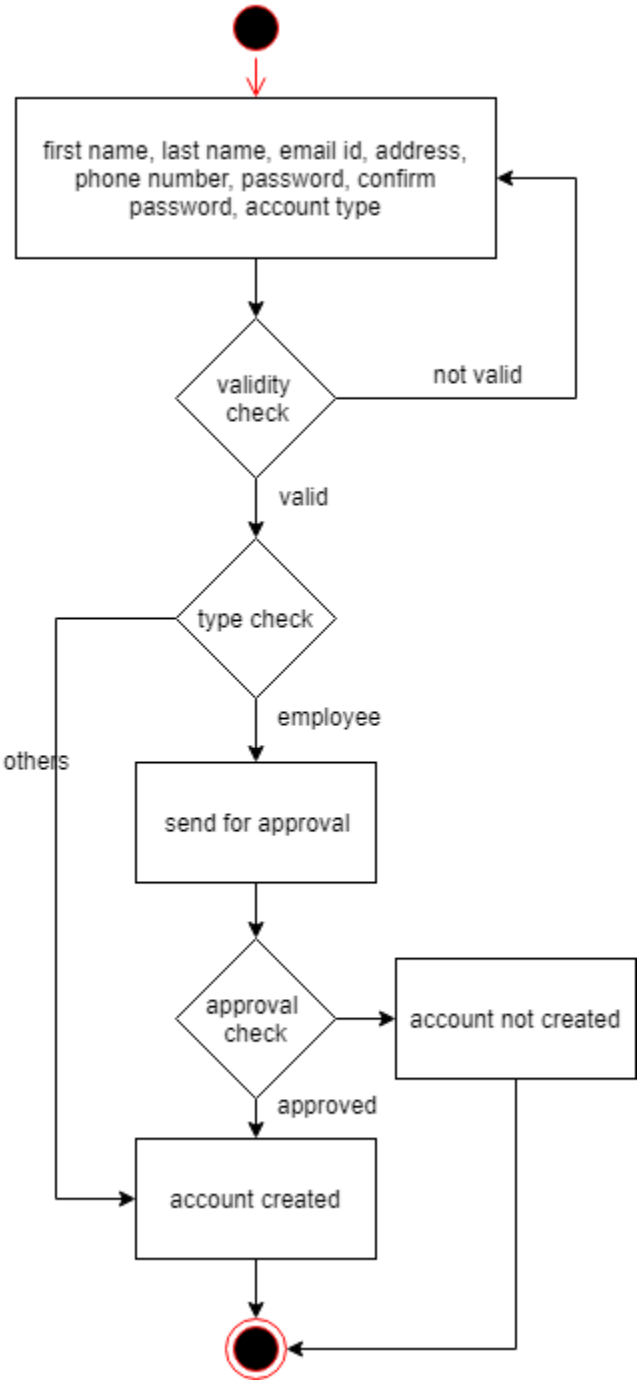


Figure 17: Activity Diagram 1.1.1 - Sign Up

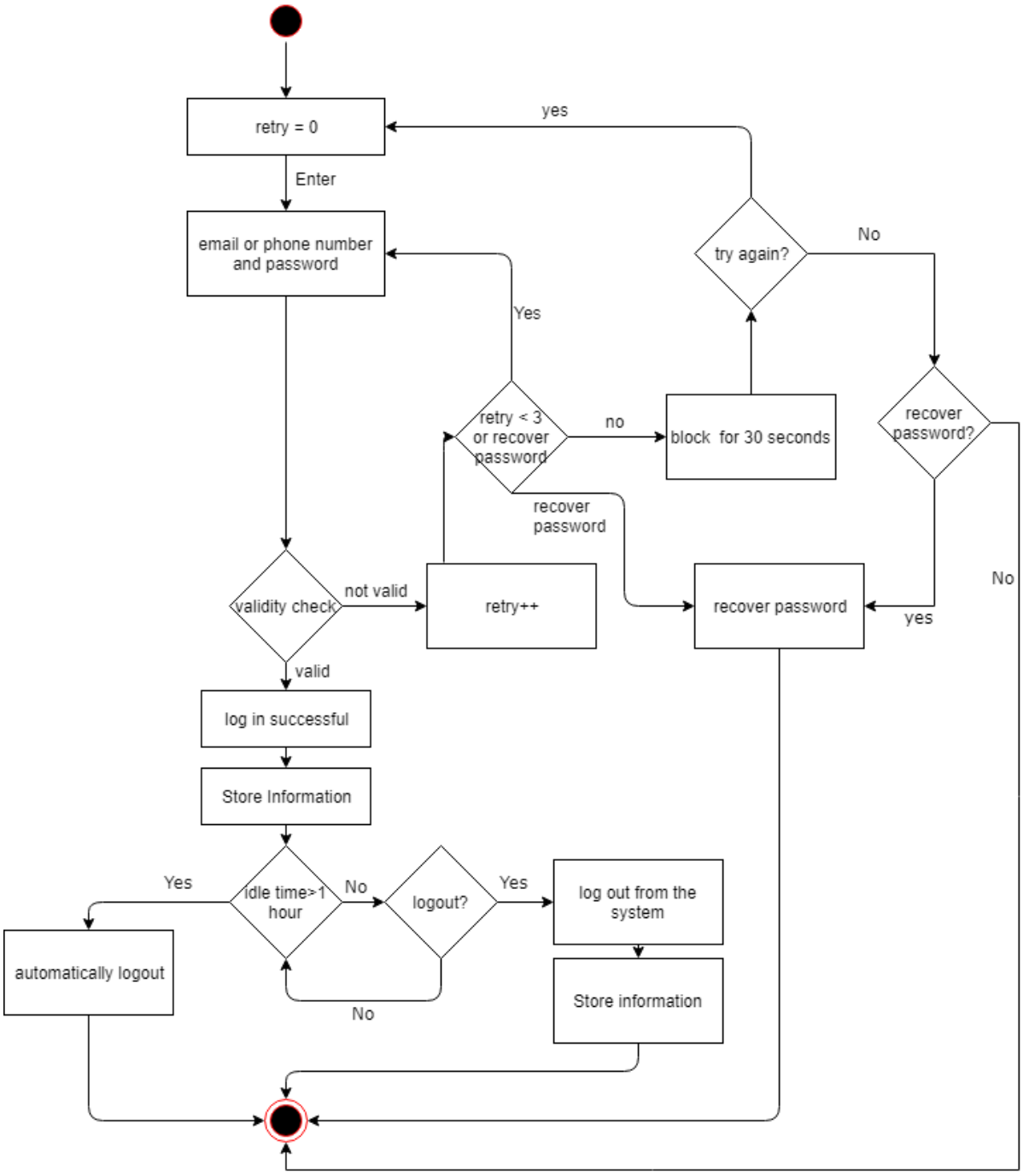


Figure 18: Activity Diagram 1.1.2 - Sign In

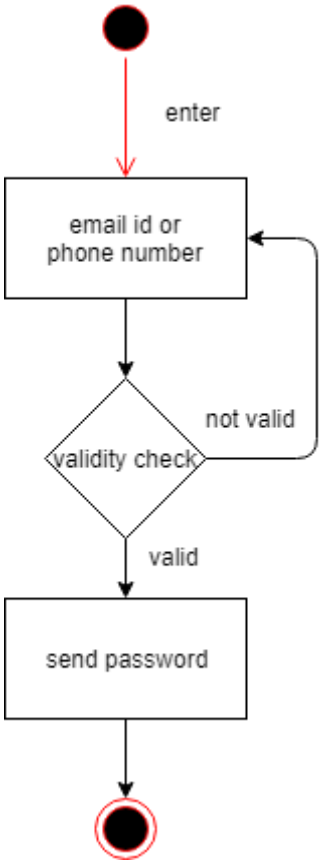


Figure 19: Activity Diagram 1.1.3 - Recover Password

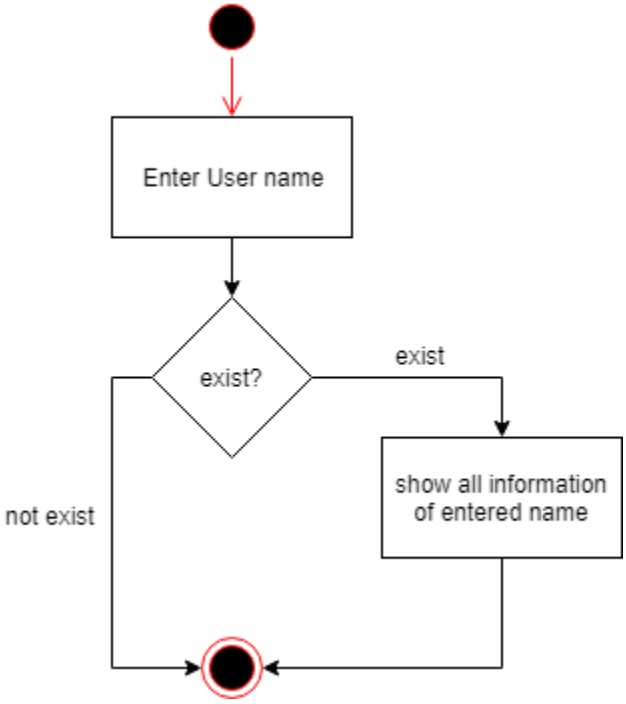


Figure 20: Activity Diagram 1.2.1 - User Search

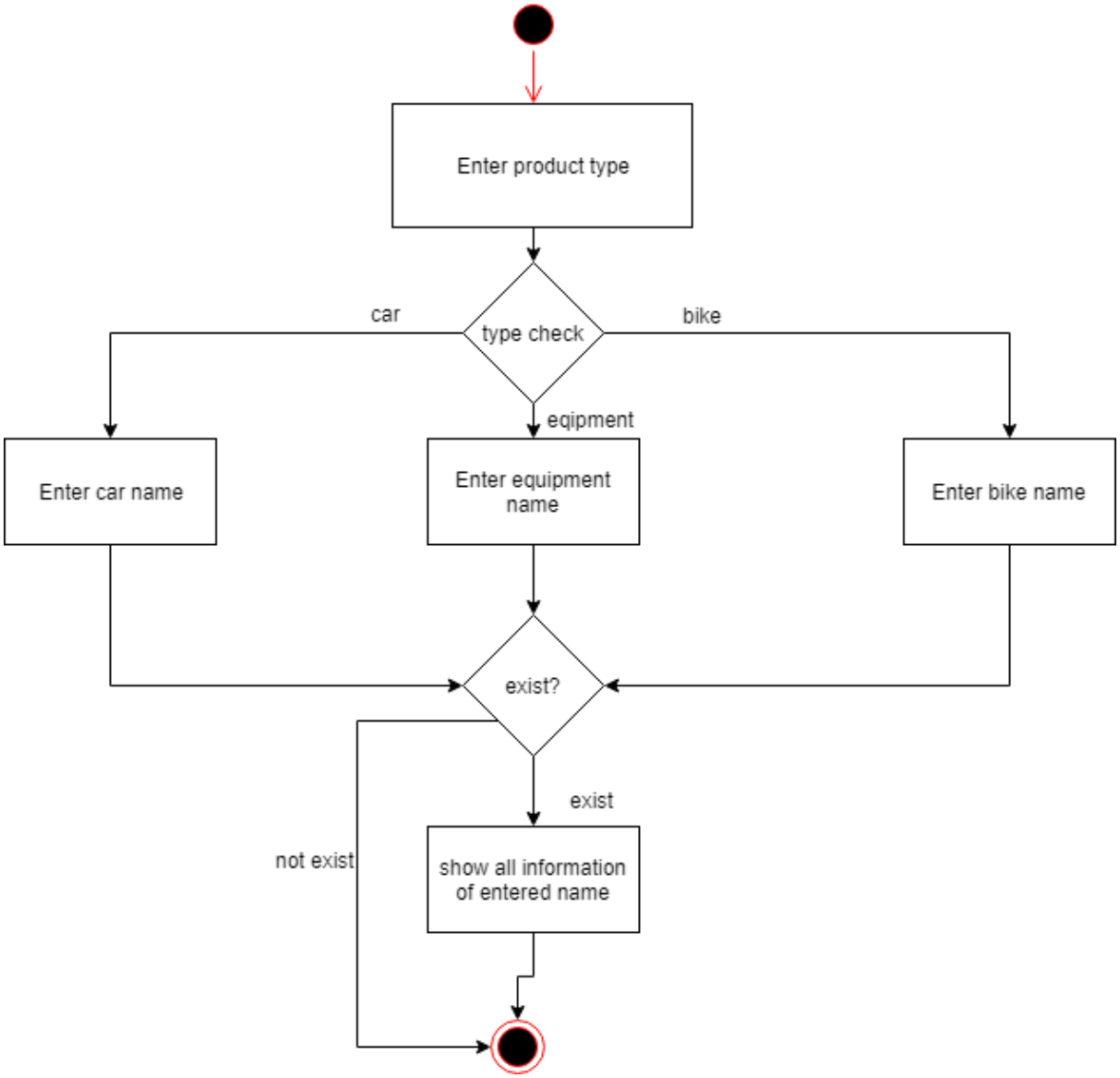


Figure 21: Activity Diagram 1.2.2 - Product Search

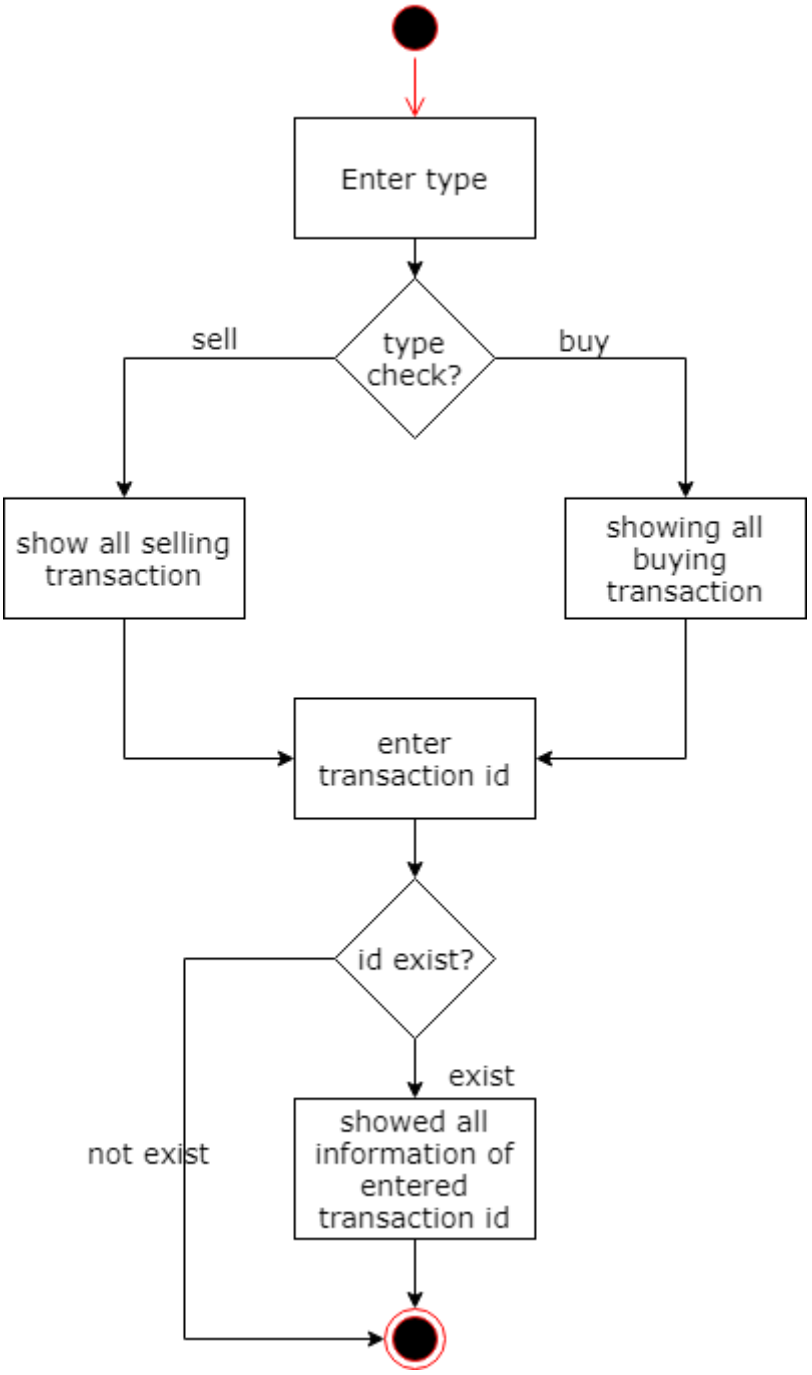


Figure 22: Activity Diagram 1.2.3 - Transaction Search

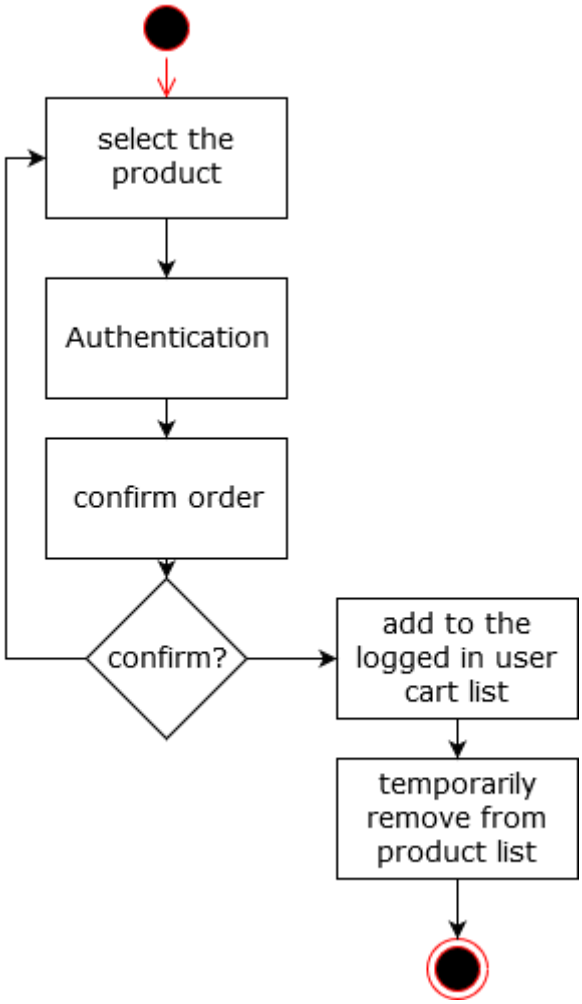


Figure 23: Activity Diagram 1.3.1.1 - Existing Product Order

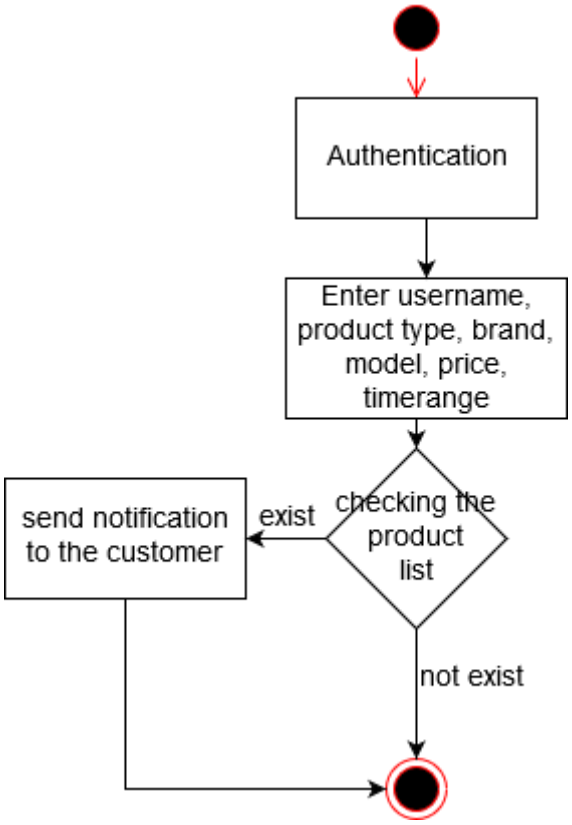


Figure 24: Activity Diagram 1.3.1.2 - Special Order

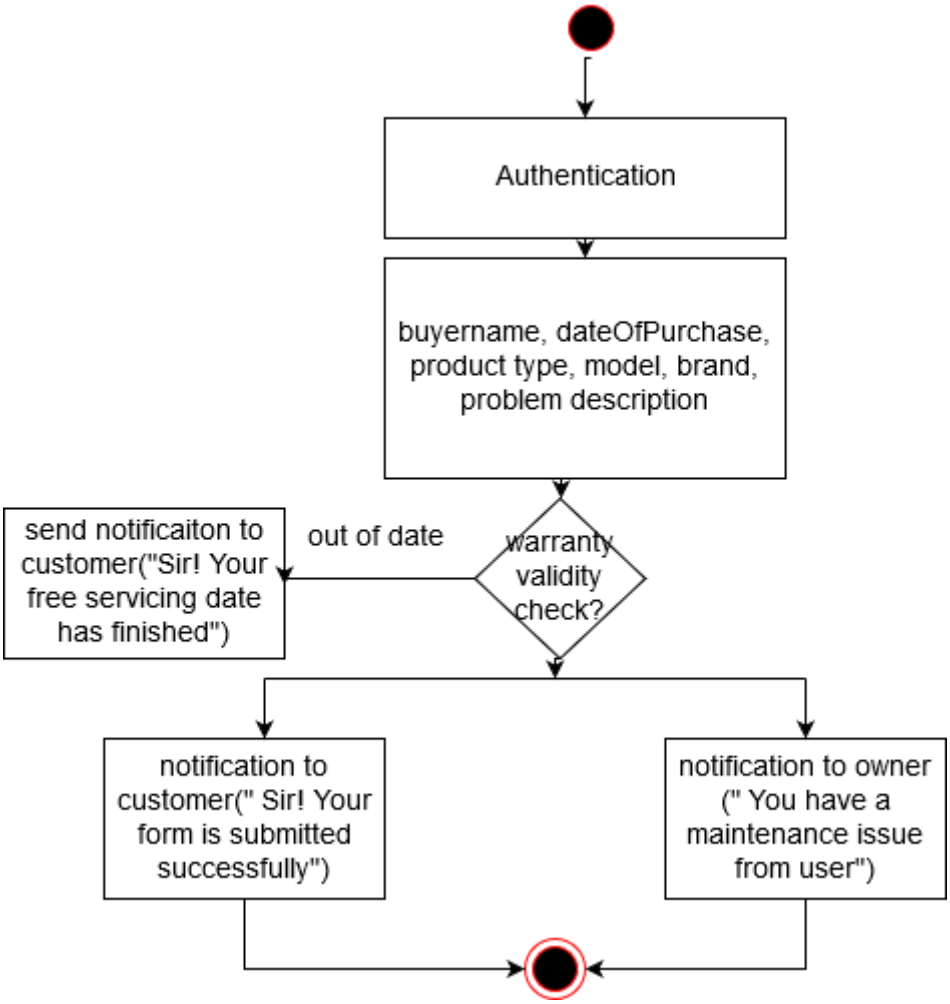


Figure 25: Activity Diagram 1.3.2 - Maintenance

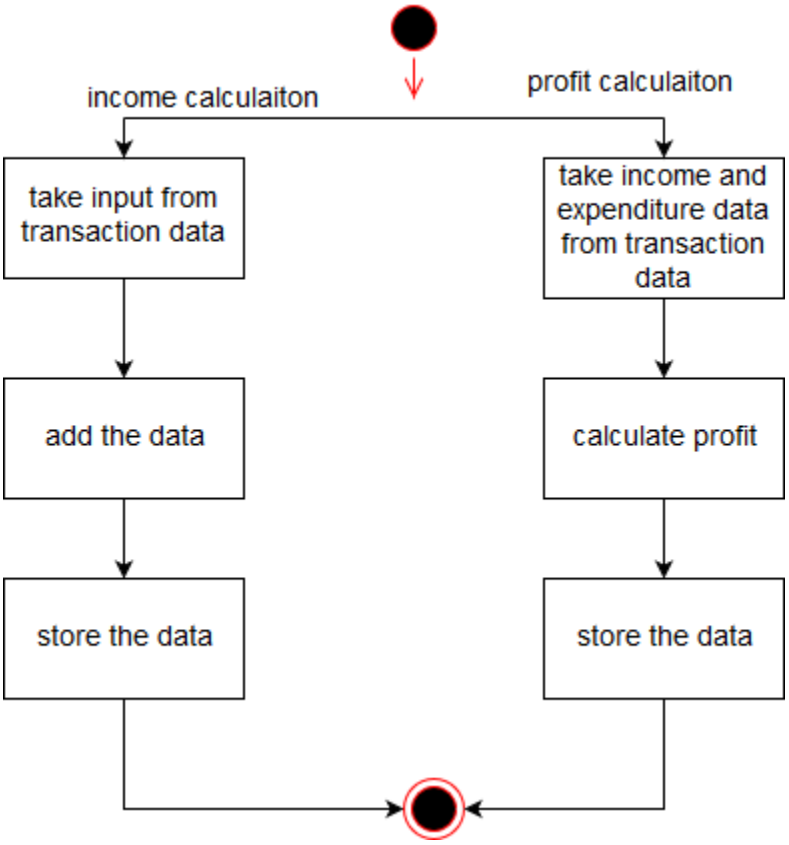


Figure 26: Activity Diagram 1..4 - Transaction

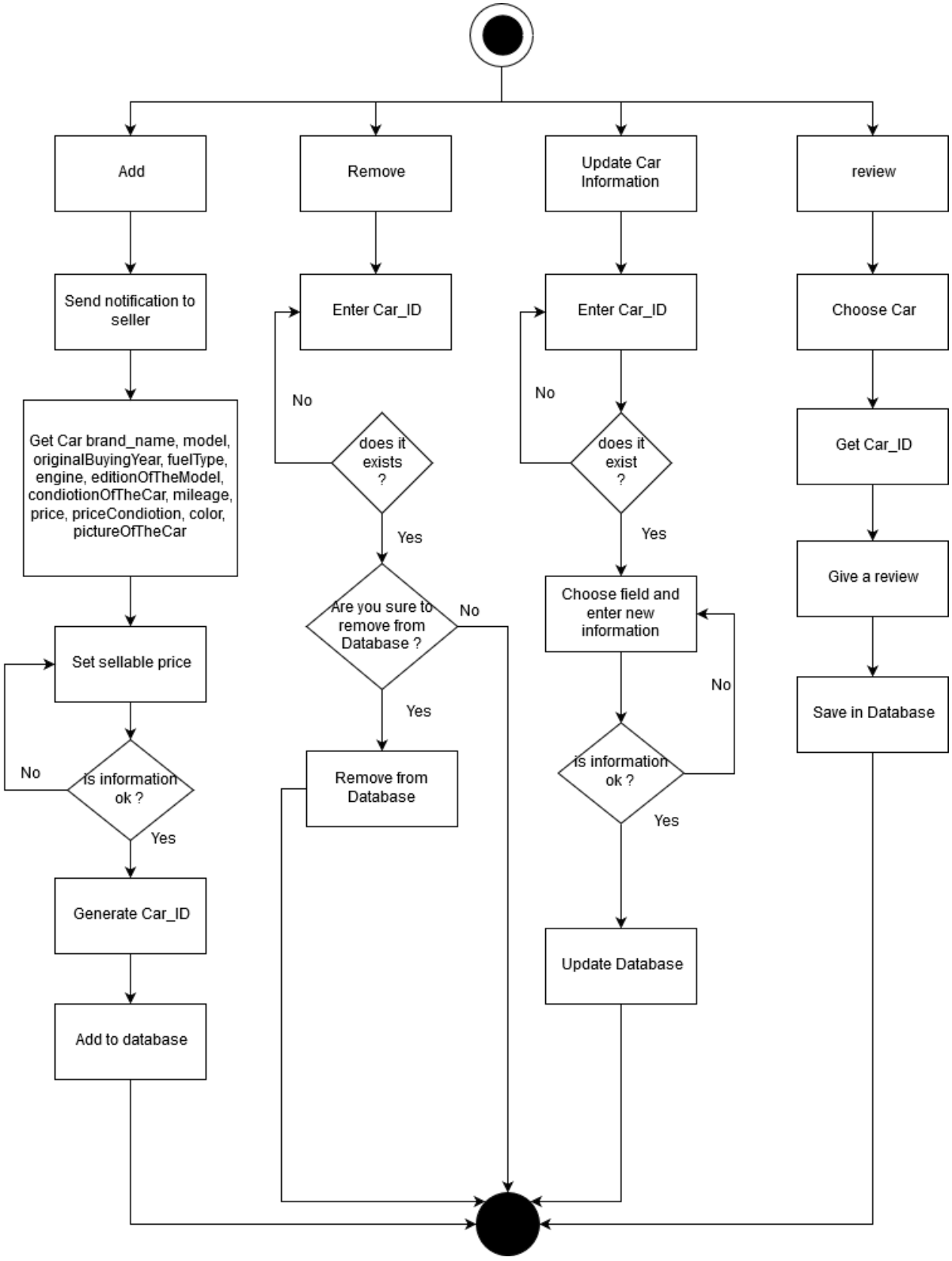


Figure 27: Activity Diagram 1.5.1.1 - Car Storage

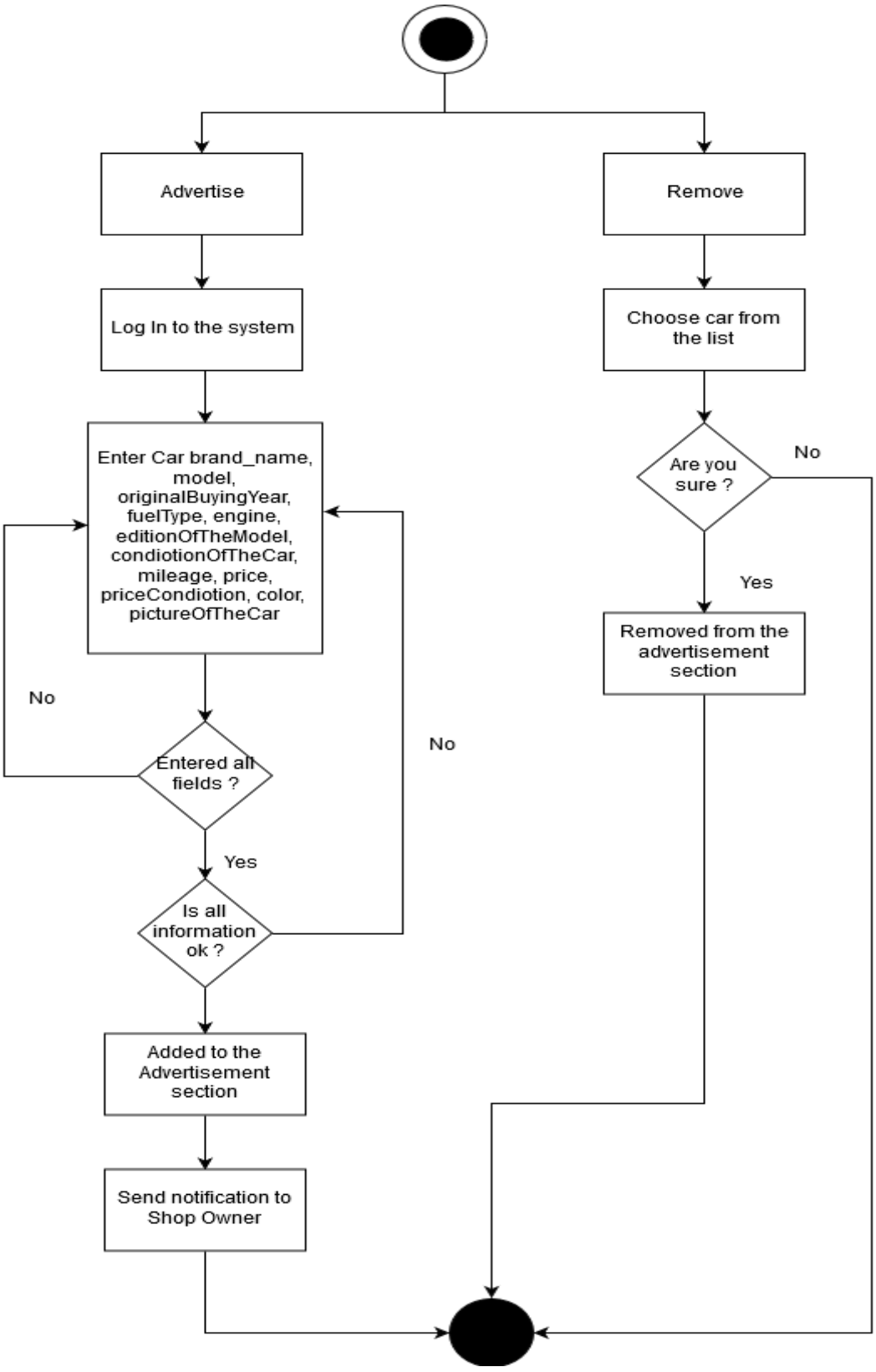


Figure 28: Activity Diagram 1.5.1.2 - Advertisement

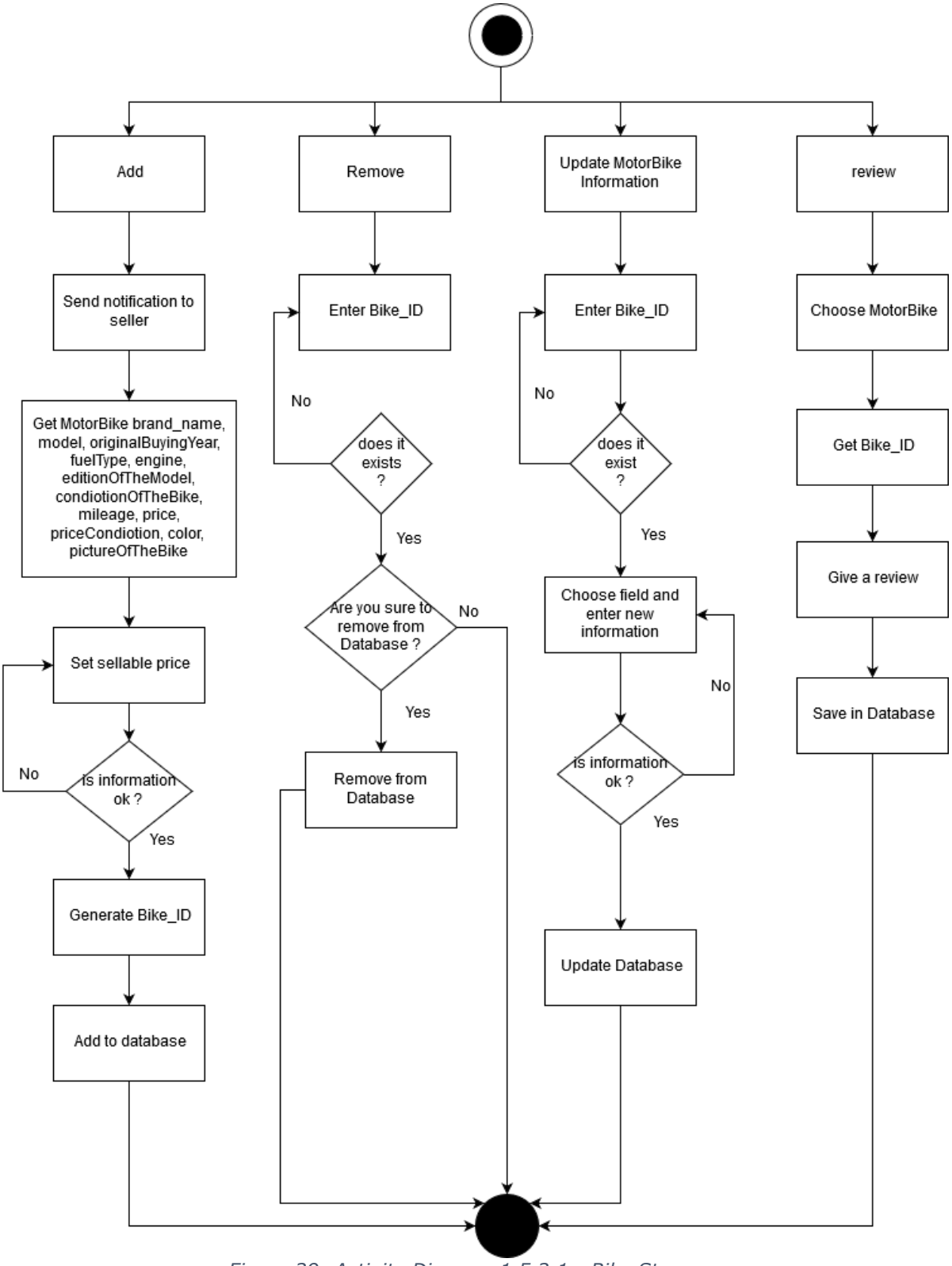


Figure 29: Activity Diagram 1.5.2.1 - Bike Storage

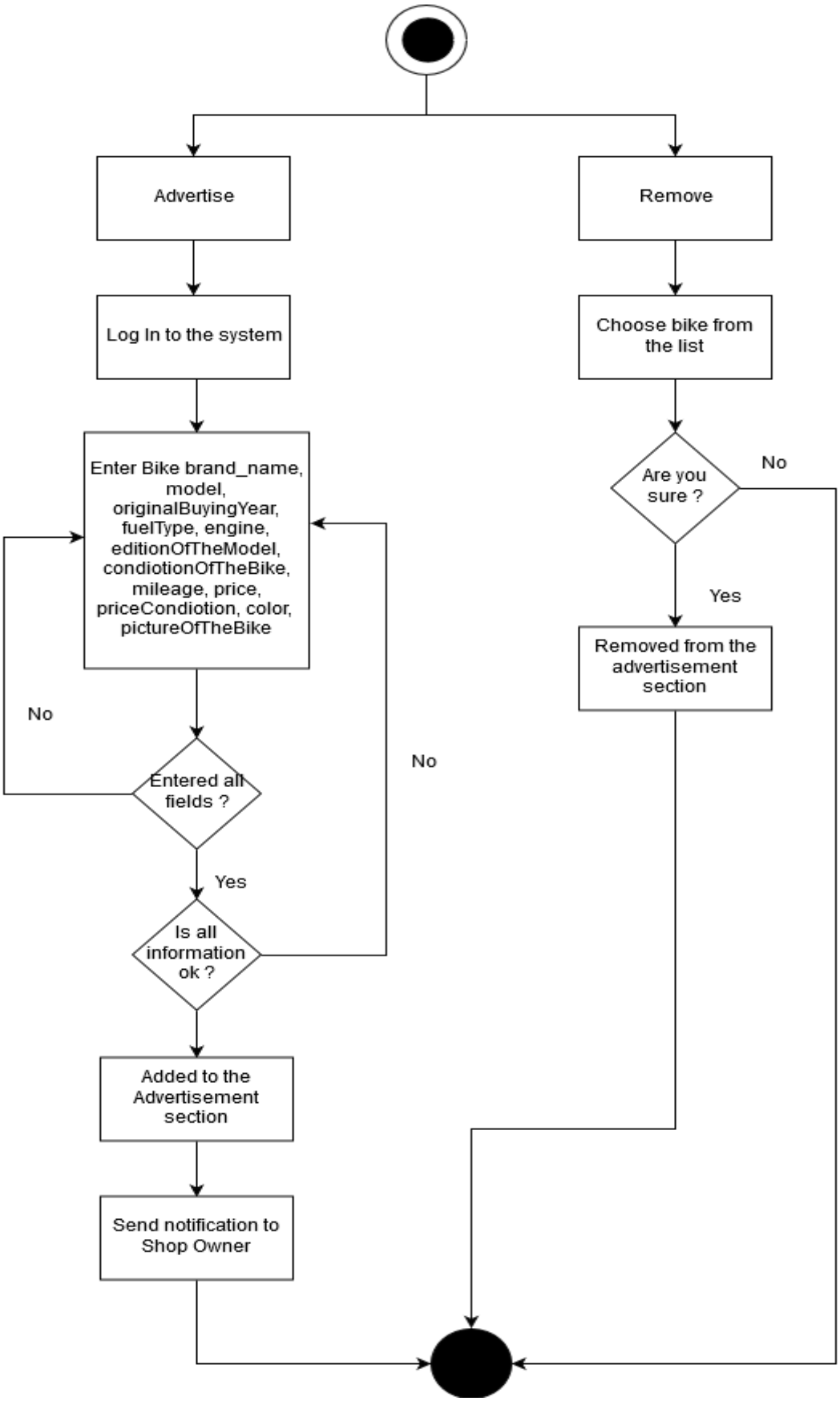


Figure 30: Activity Diagram 1.5.2.2 - Advertisement

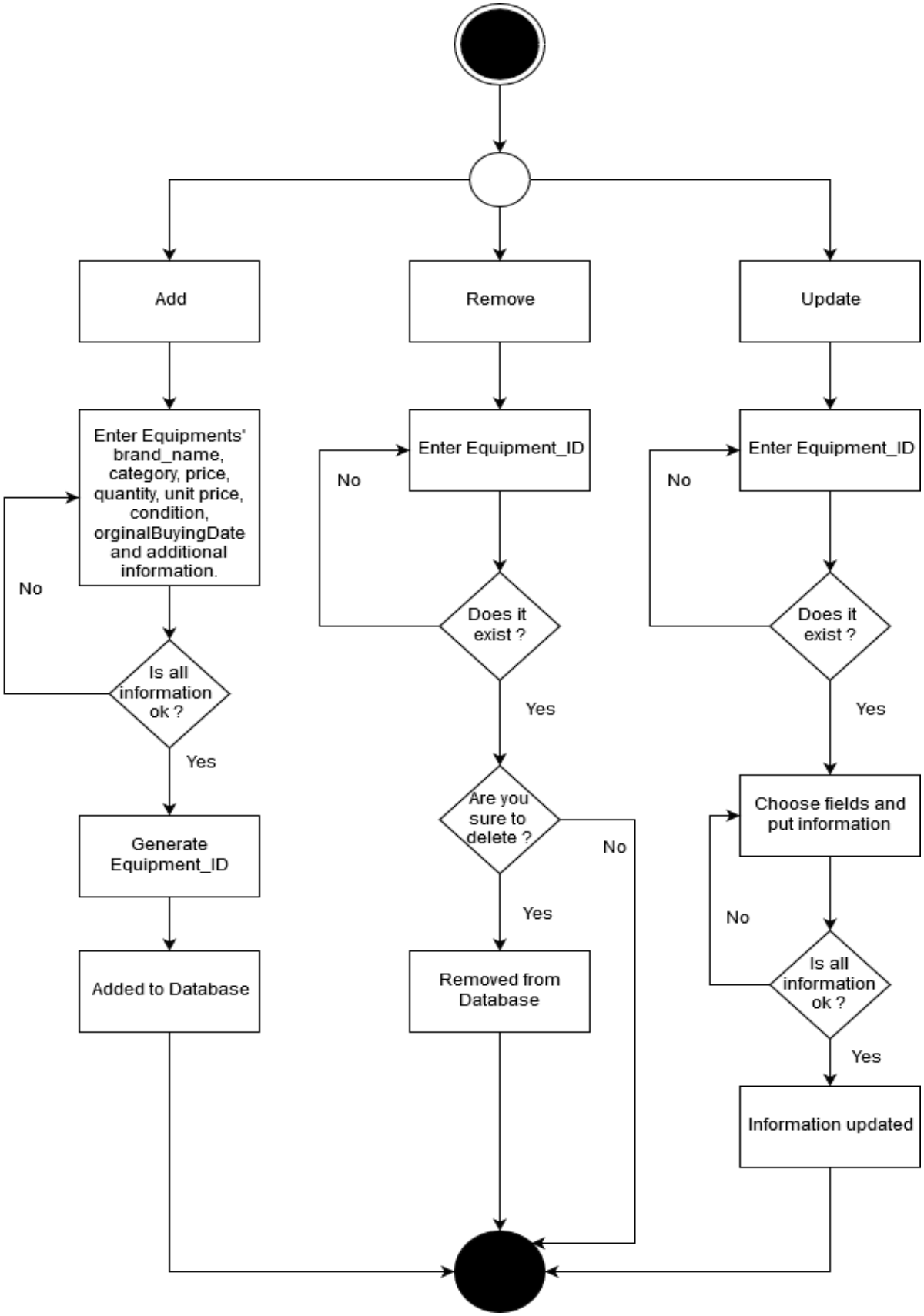


Figure 31: Activity Diagram 1.6 - Equipment Management

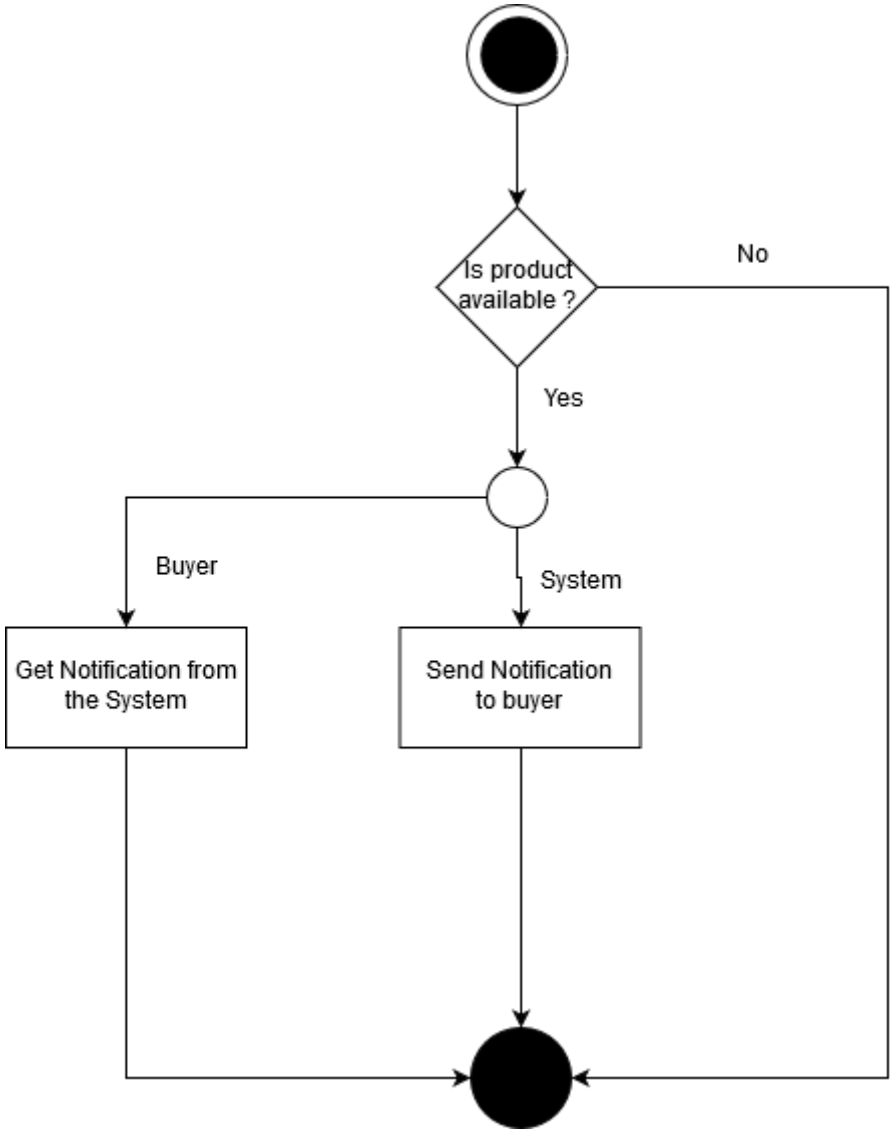


Figure 32: Activity Diagram 1.7.1 - Order Product Available

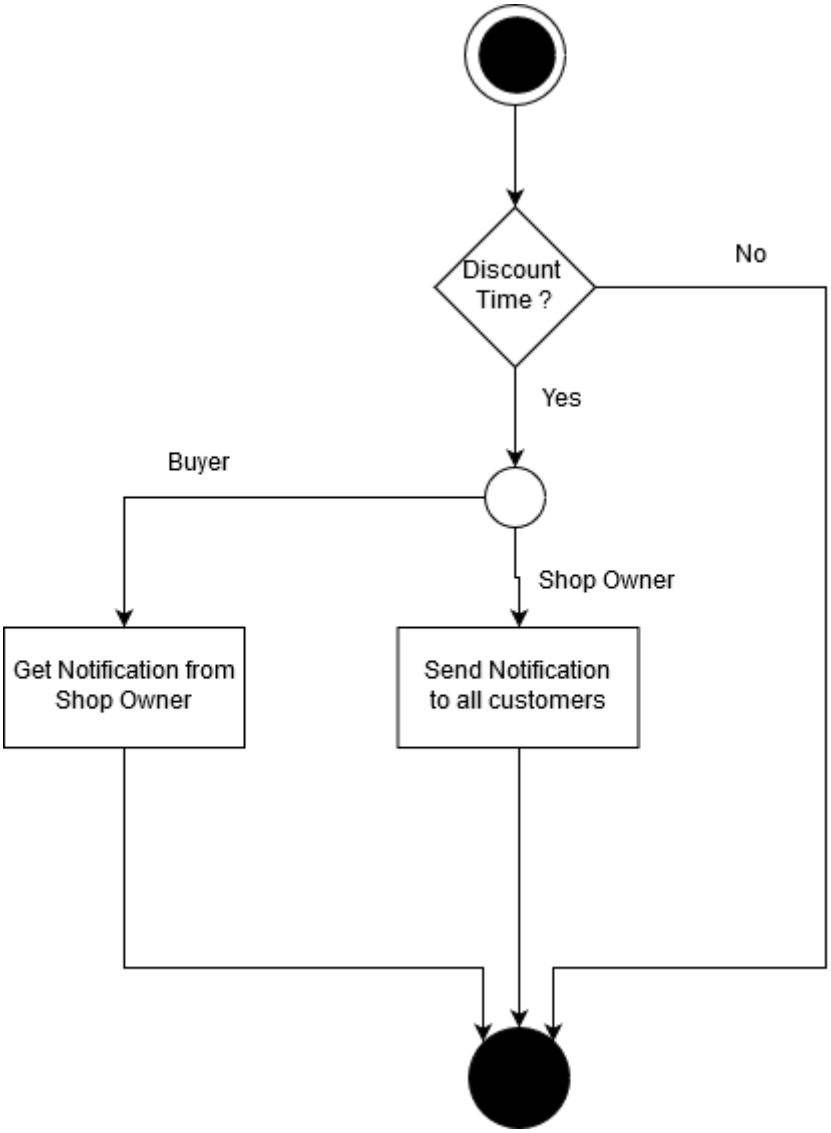


Figure 33: Activity Diagram 1.7.2 - Discount time

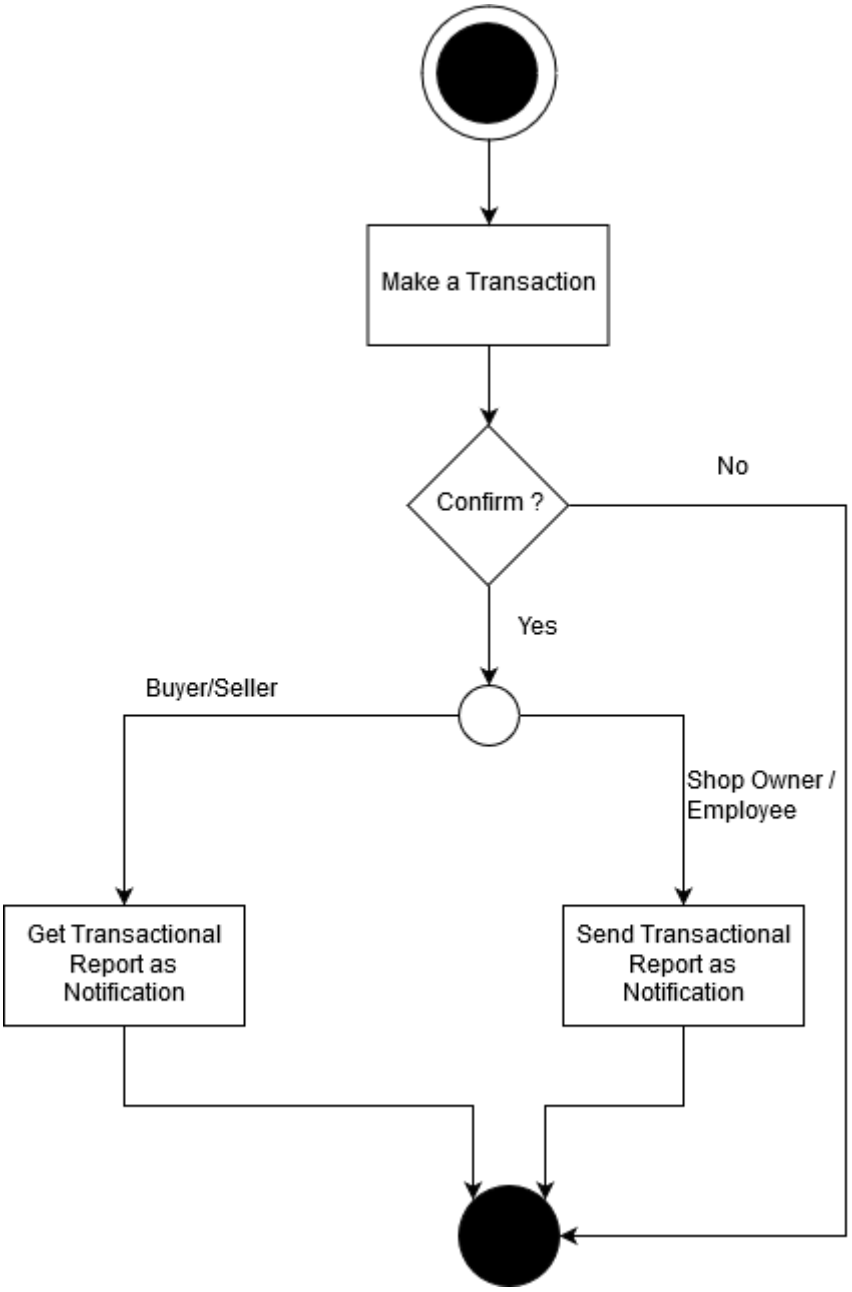


Figure 34: Activity Diagram 1.7.3 - Transactional Report

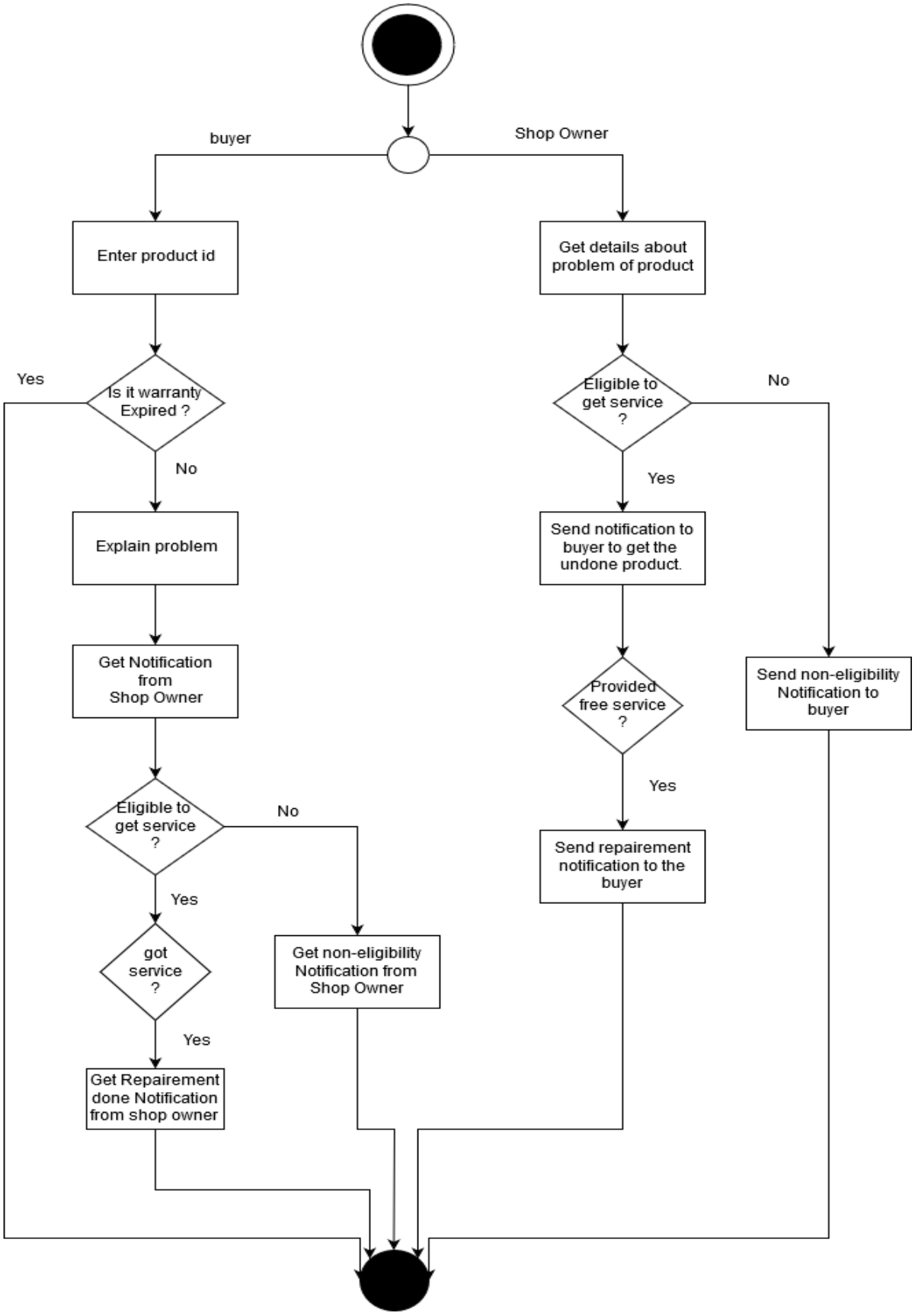


Figure 35: Activity Diagram 1.7.4 – Maintenance log

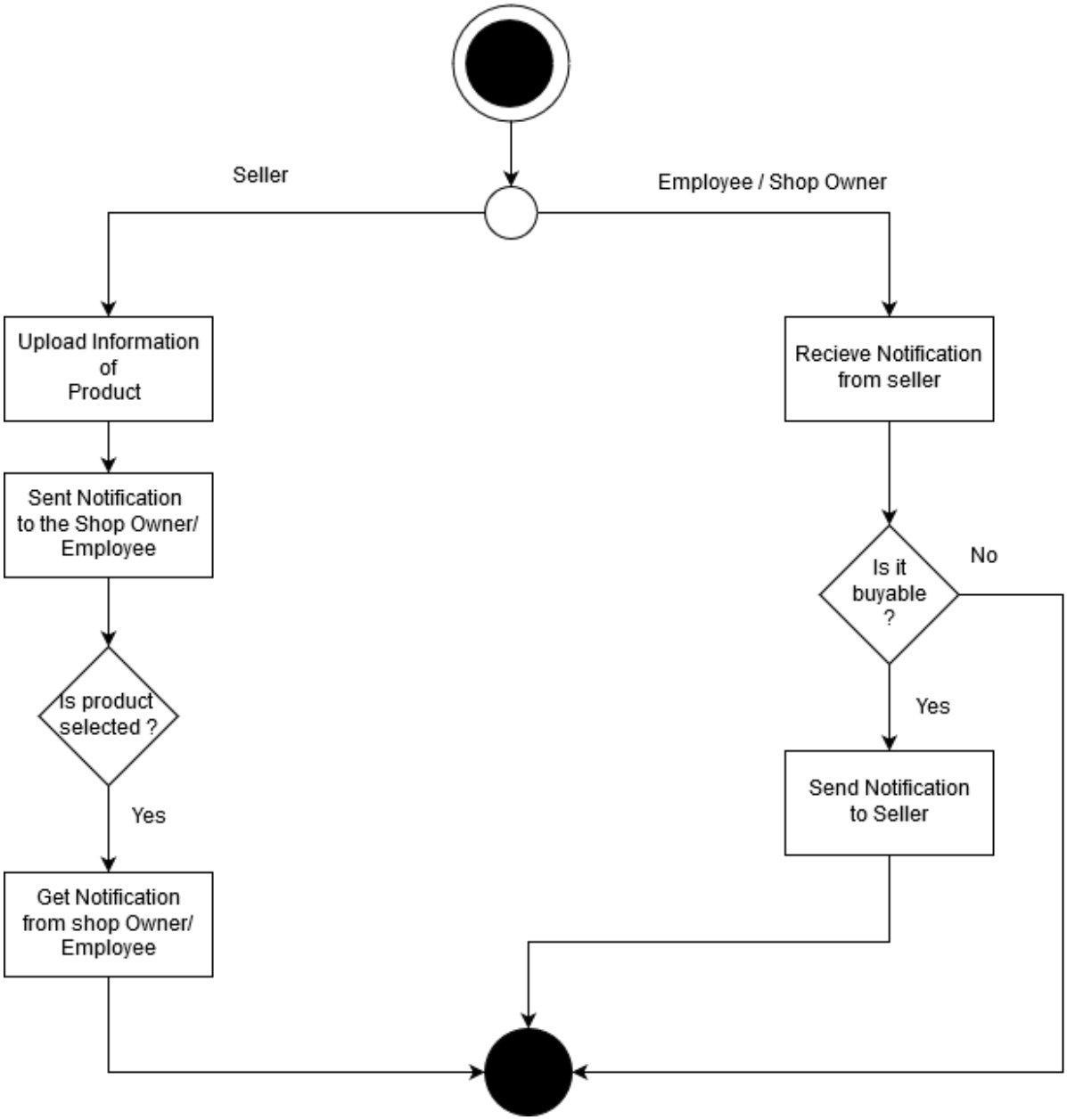


Figure 36: Activity Diagram 1.7.5 - Advertisement

4.5 Swim lane Diagram

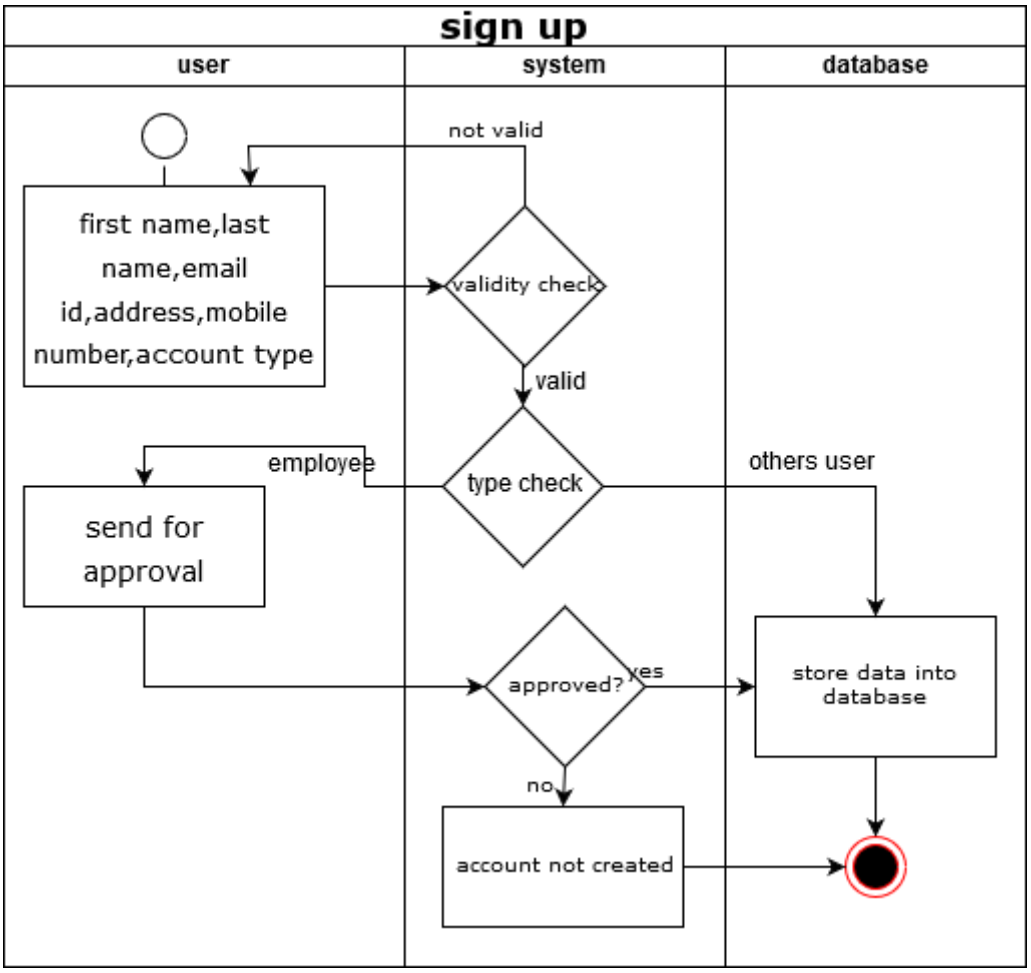


Figure 37: Swim Lane Diagram 1.1.1 - Sign Up

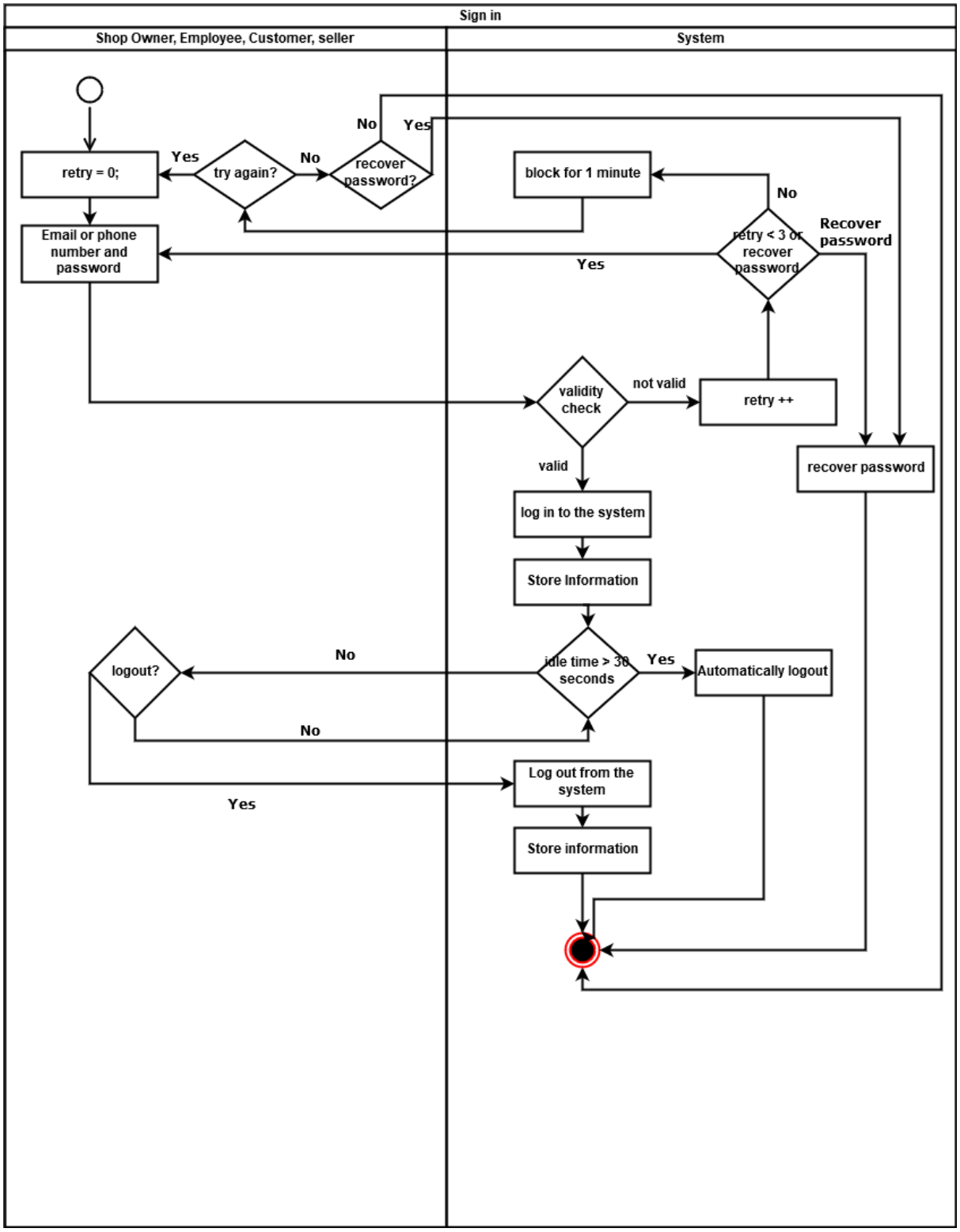


Figure 38: Swim Lane Diagram 1.1.2 - Sign In

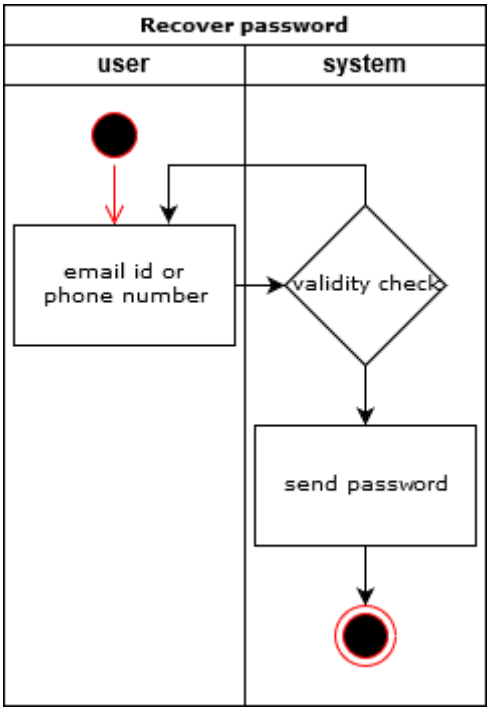


Figure 39: Swim Lane Diagram 1.1.3 - Recover Password

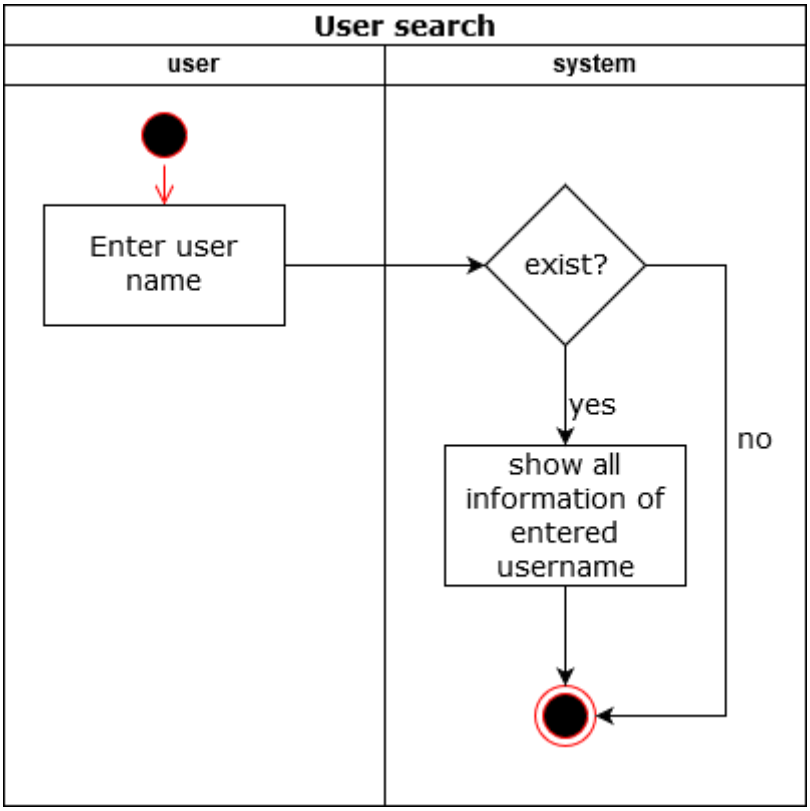


Figure 40: Swim Lane Diagram 1.2.1 - User Search

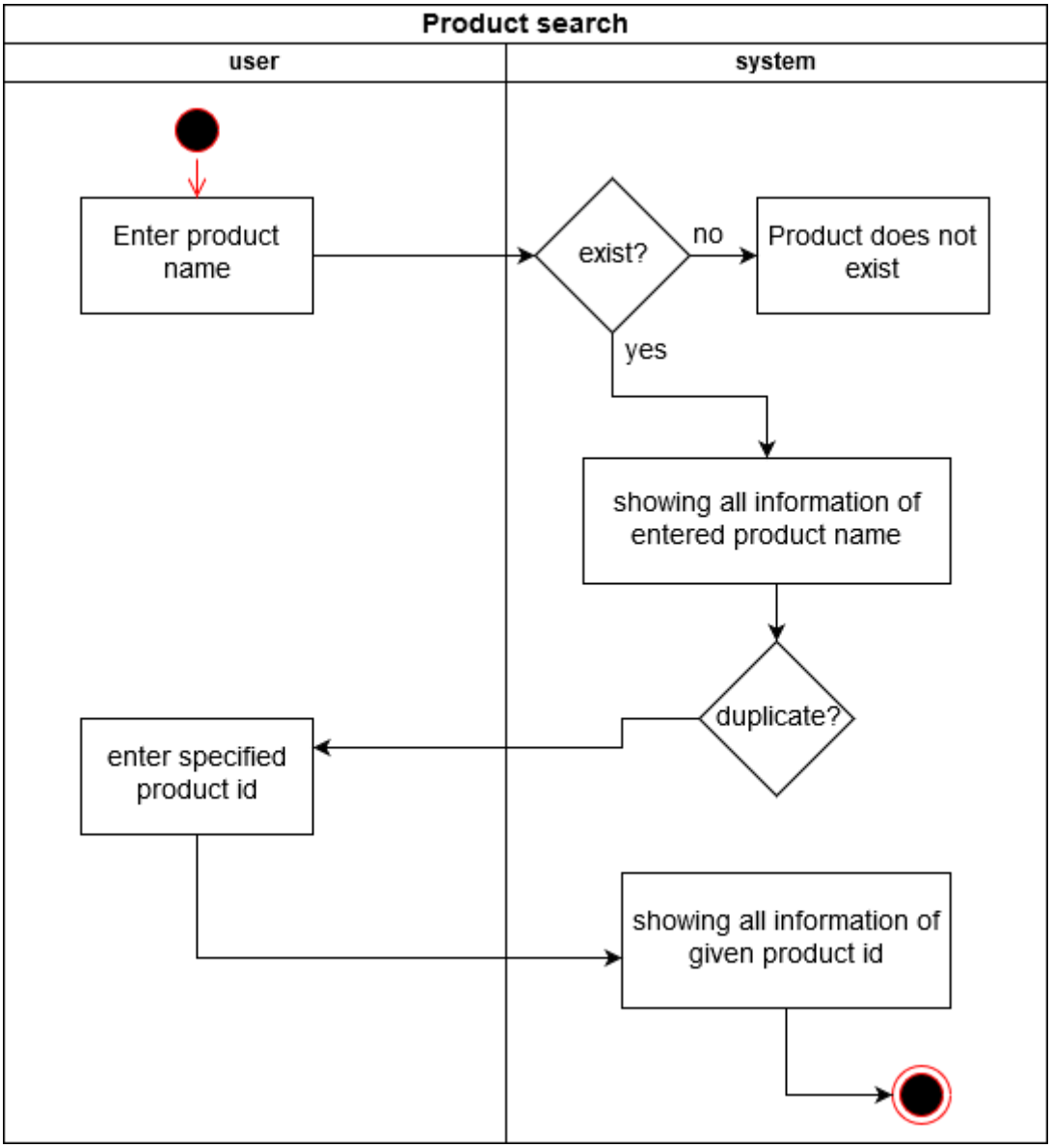


Figure 41: Swim Lane Diagram 1.2.2 - Product Search

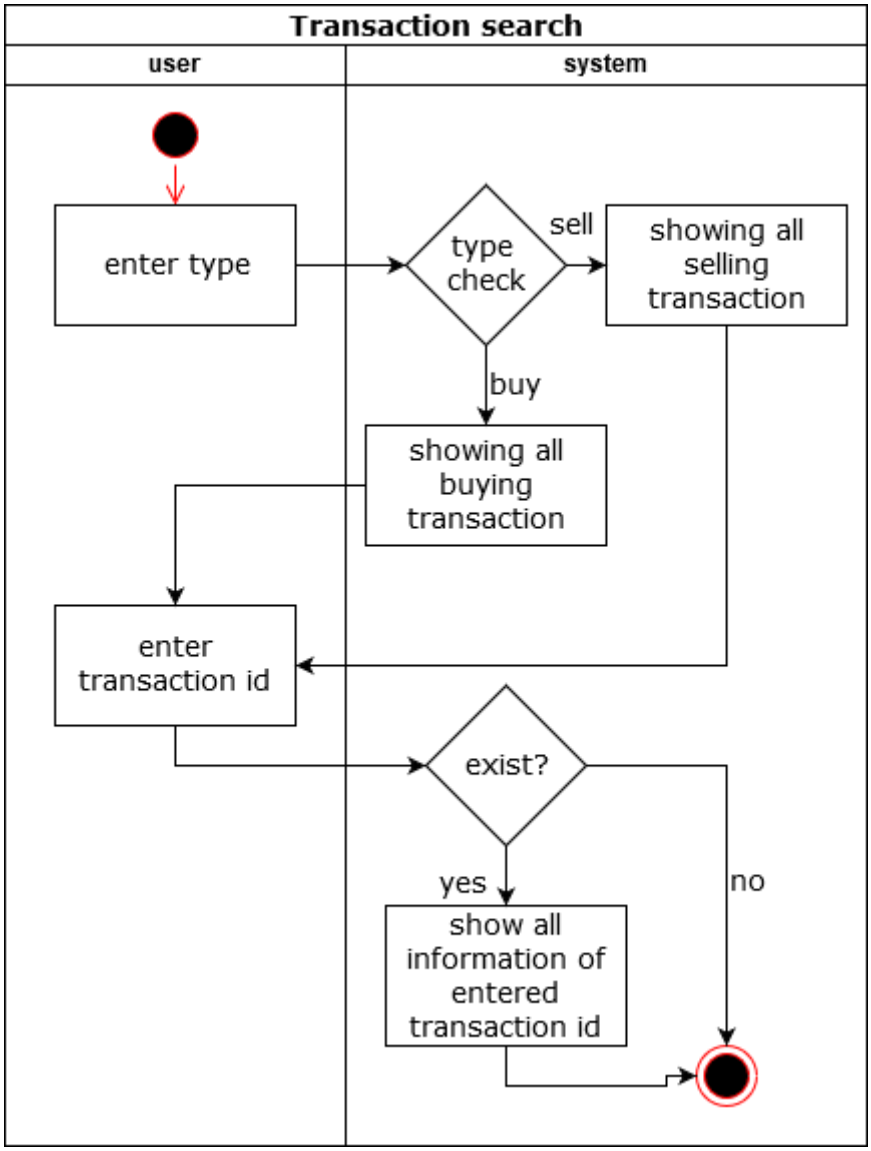


Figure 42: Swim Lane Diagram 1.2.3 - Transactional Search

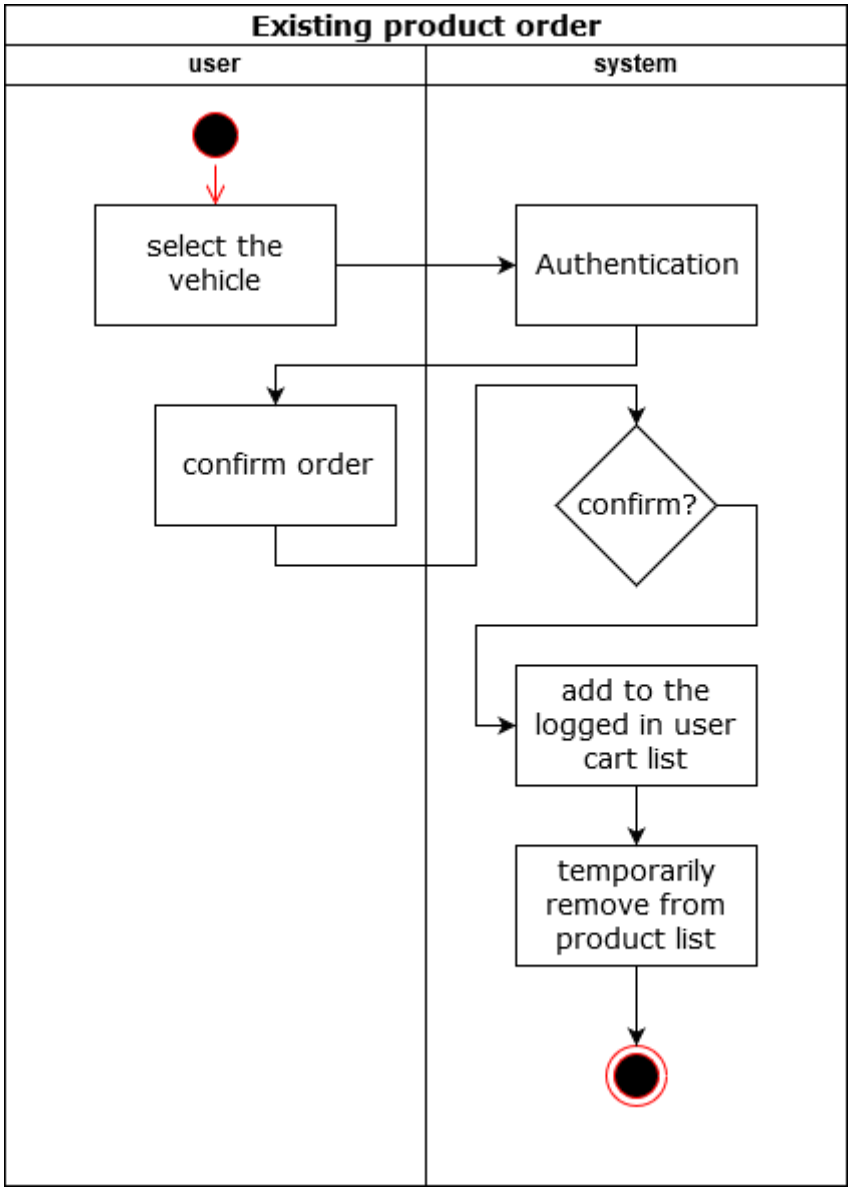


Figure 43: Swim Lane Diagram 1.3.1.1 - Existing Product Order

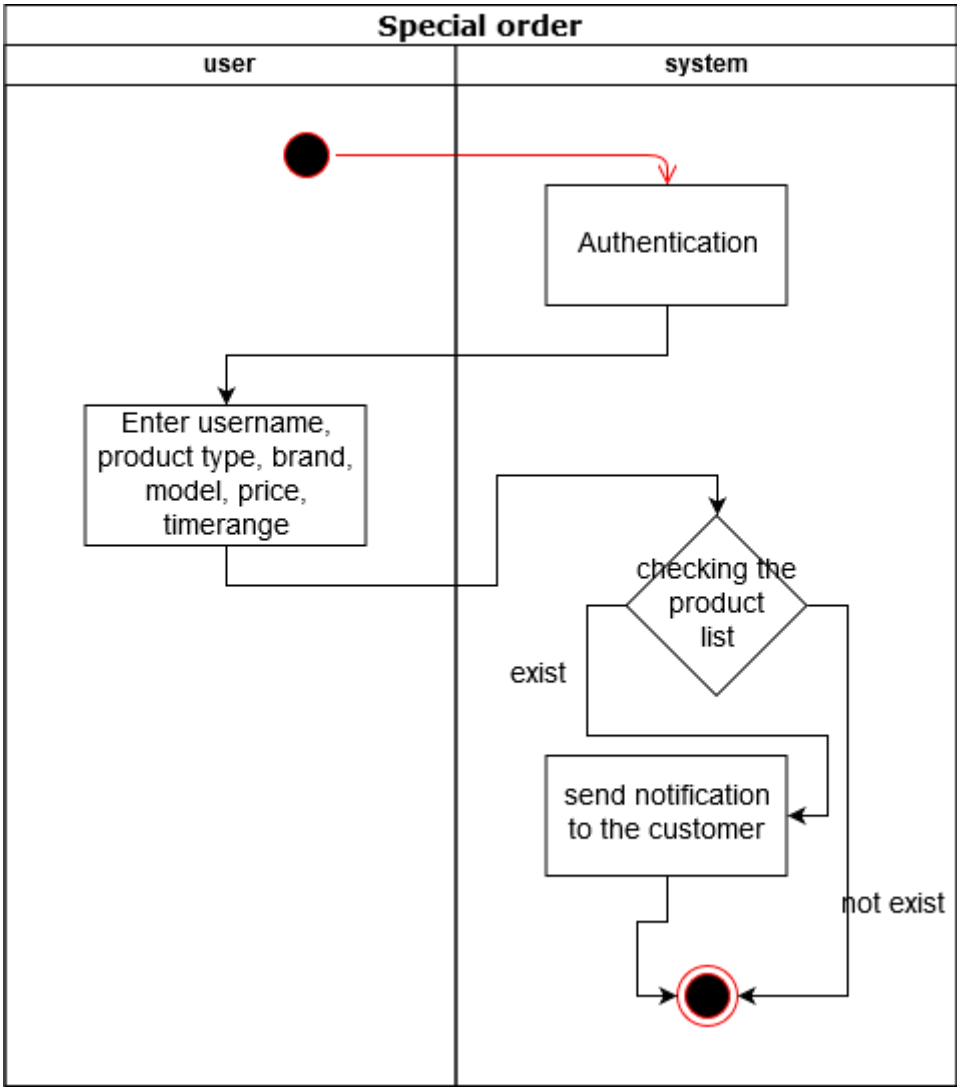


Figure 44: Swim Lane Diagram 1.3.1.2 - Special Order

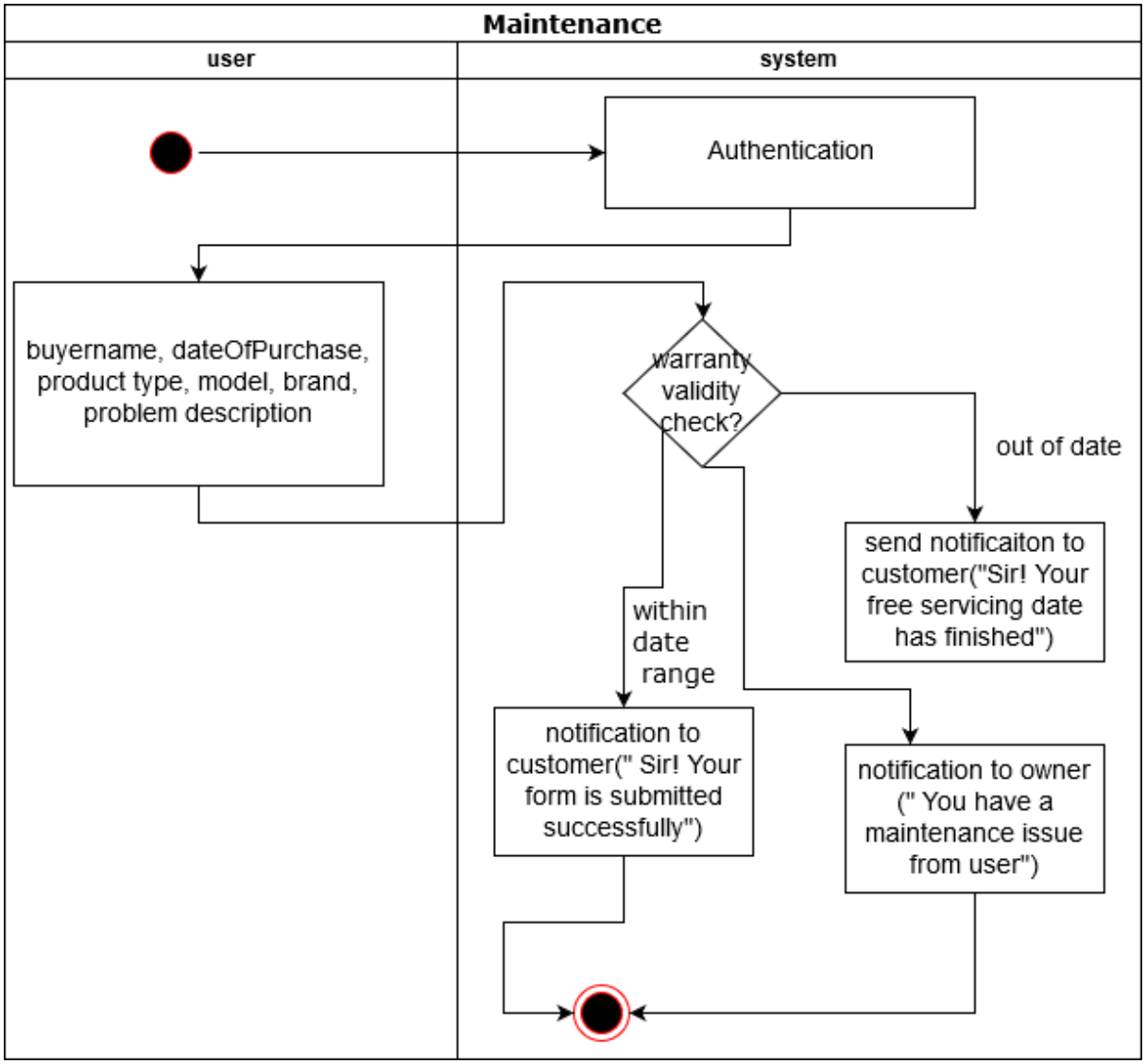


Figure 45: Swim Lane Diagram 1.3.2 - Maintenance

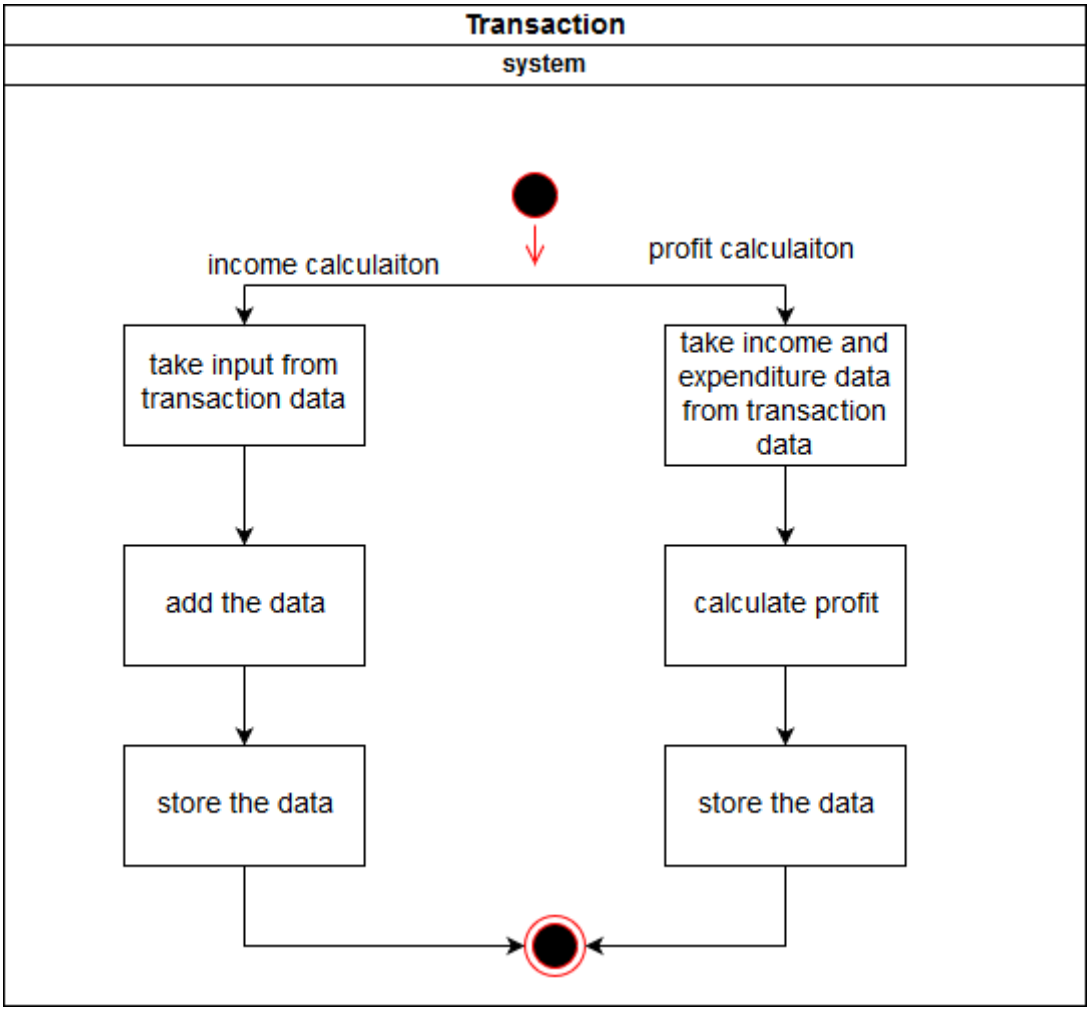


Figure 46: Swim Lane Diagram 1.4 - Transaction

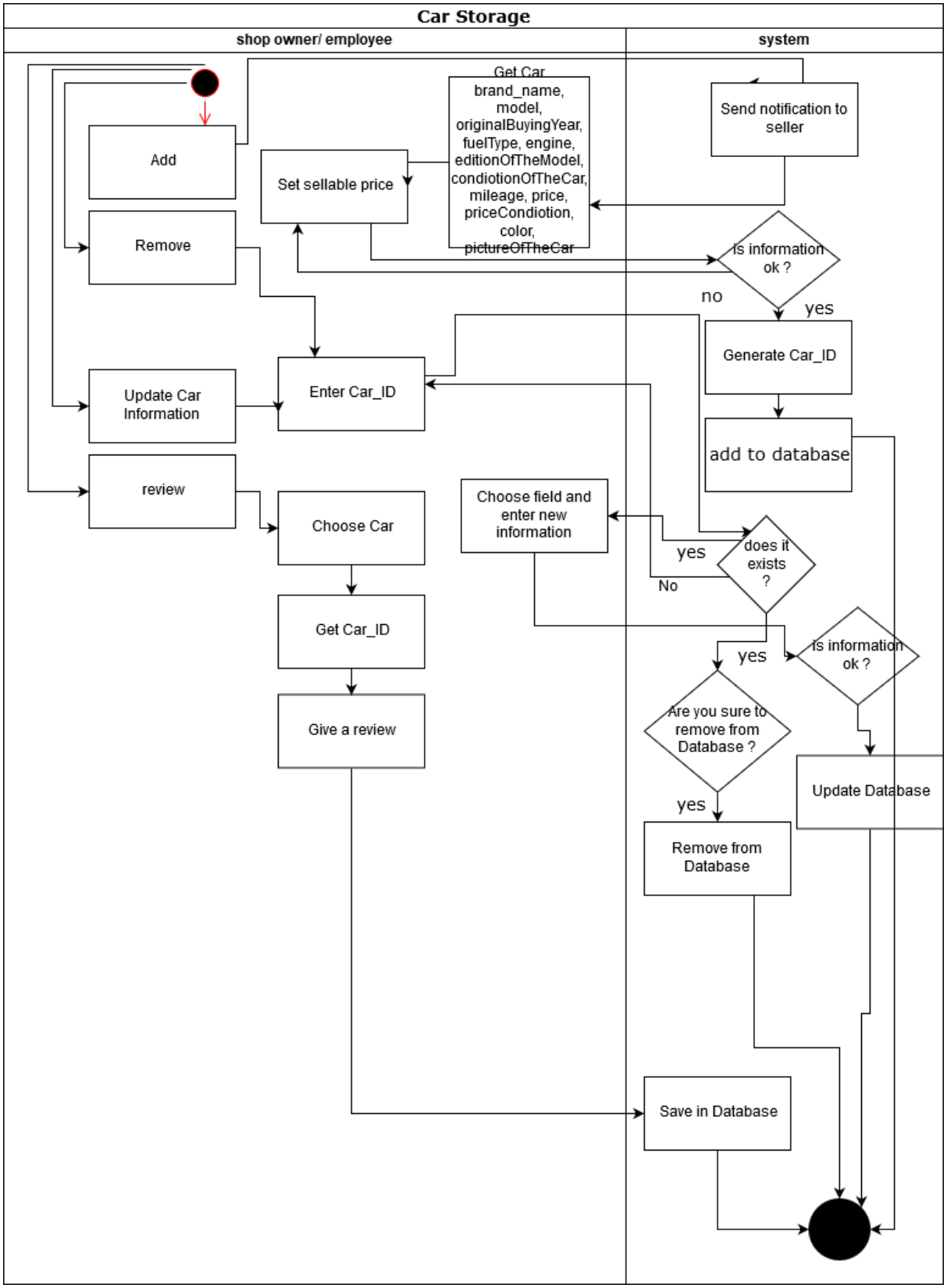


Figure 47: Swim Lane Diagram 1.5.1.1 - Car Storage

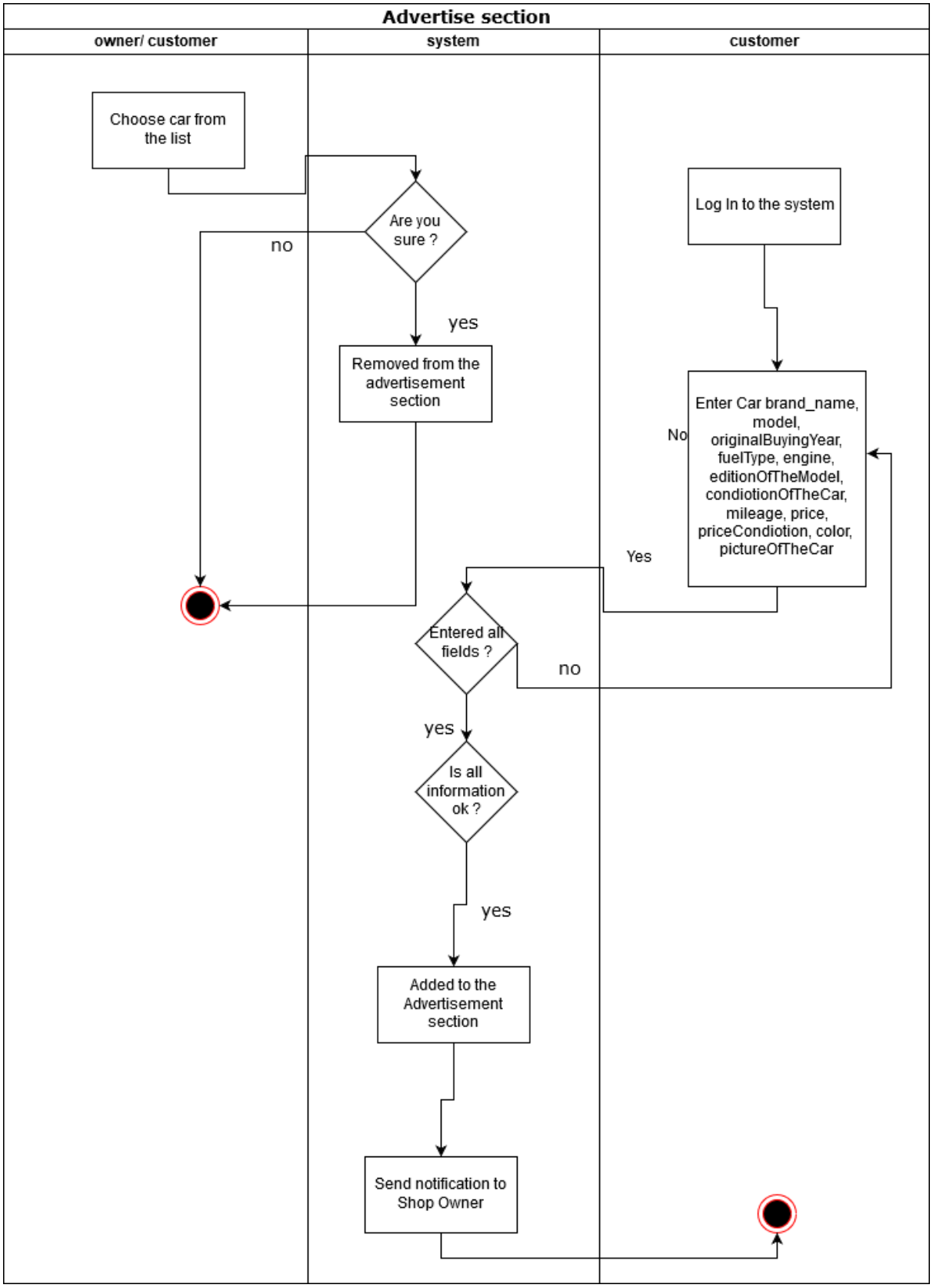


Figure 48: Swim Lane Diagram 1.5.1.2 – Advertisement

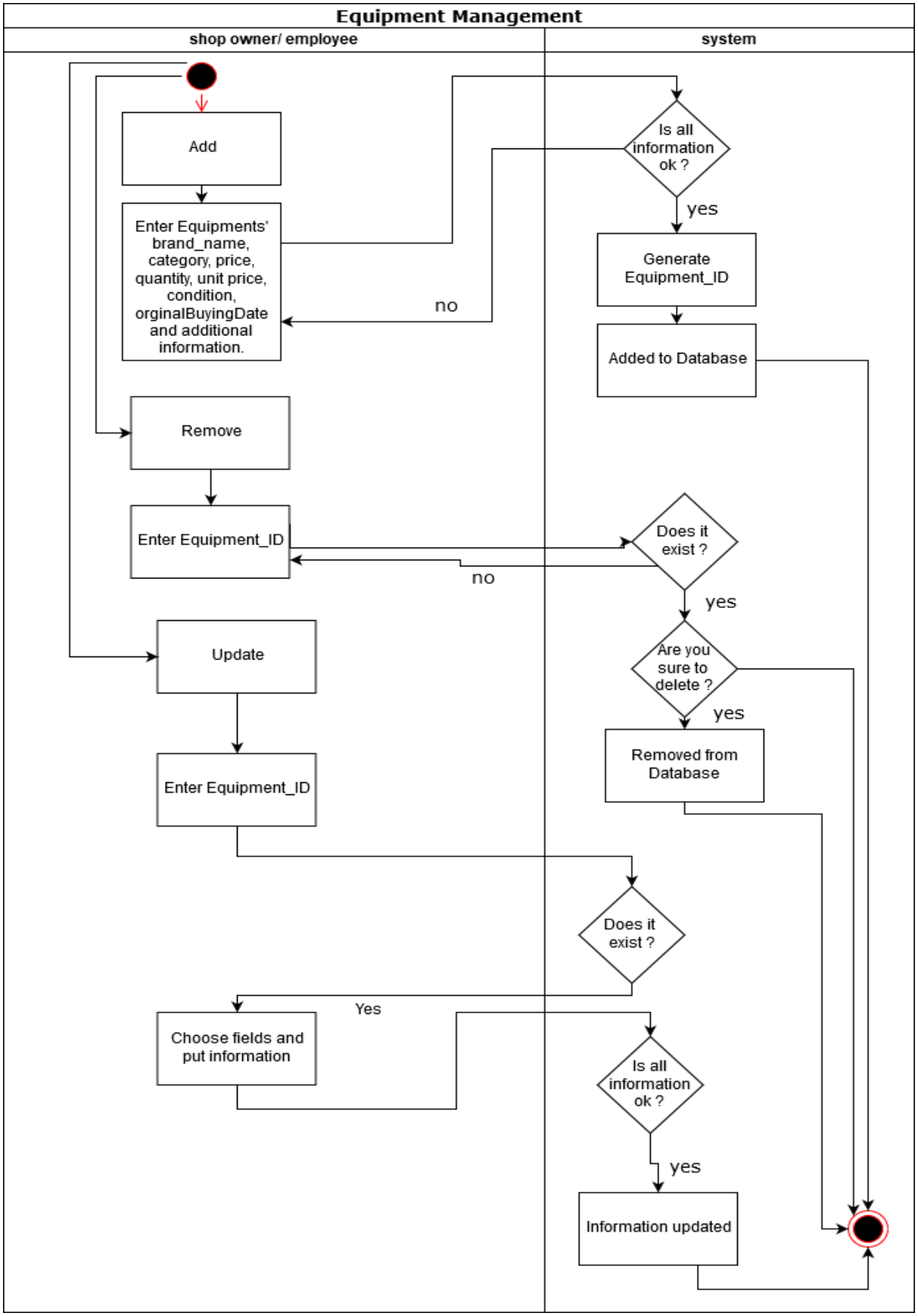


Figure 49: Swim Lane Diagram 1.6 - Equipment Management

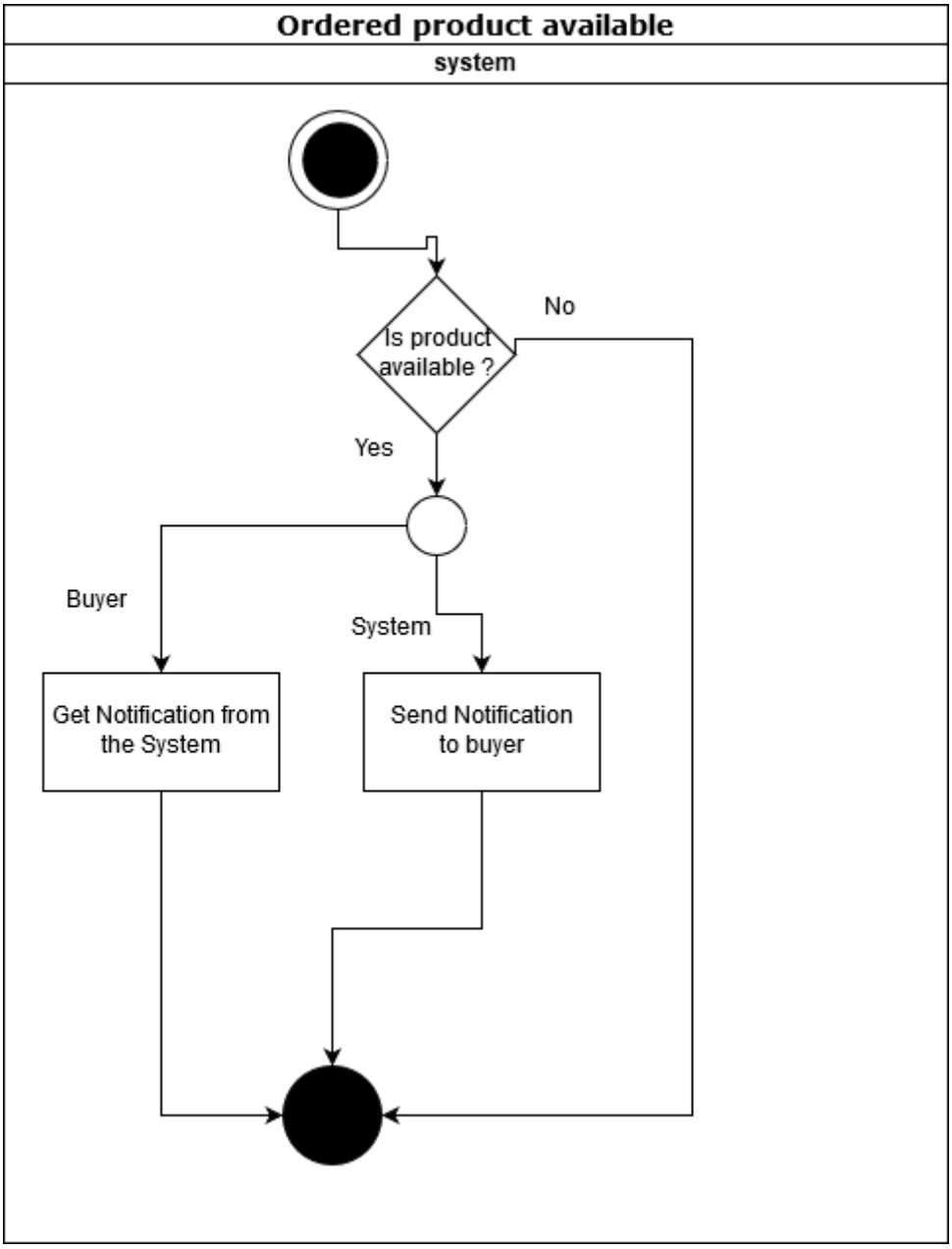


Figure 50: Swim Lane Diagram 1.7.1 - Order Product Available

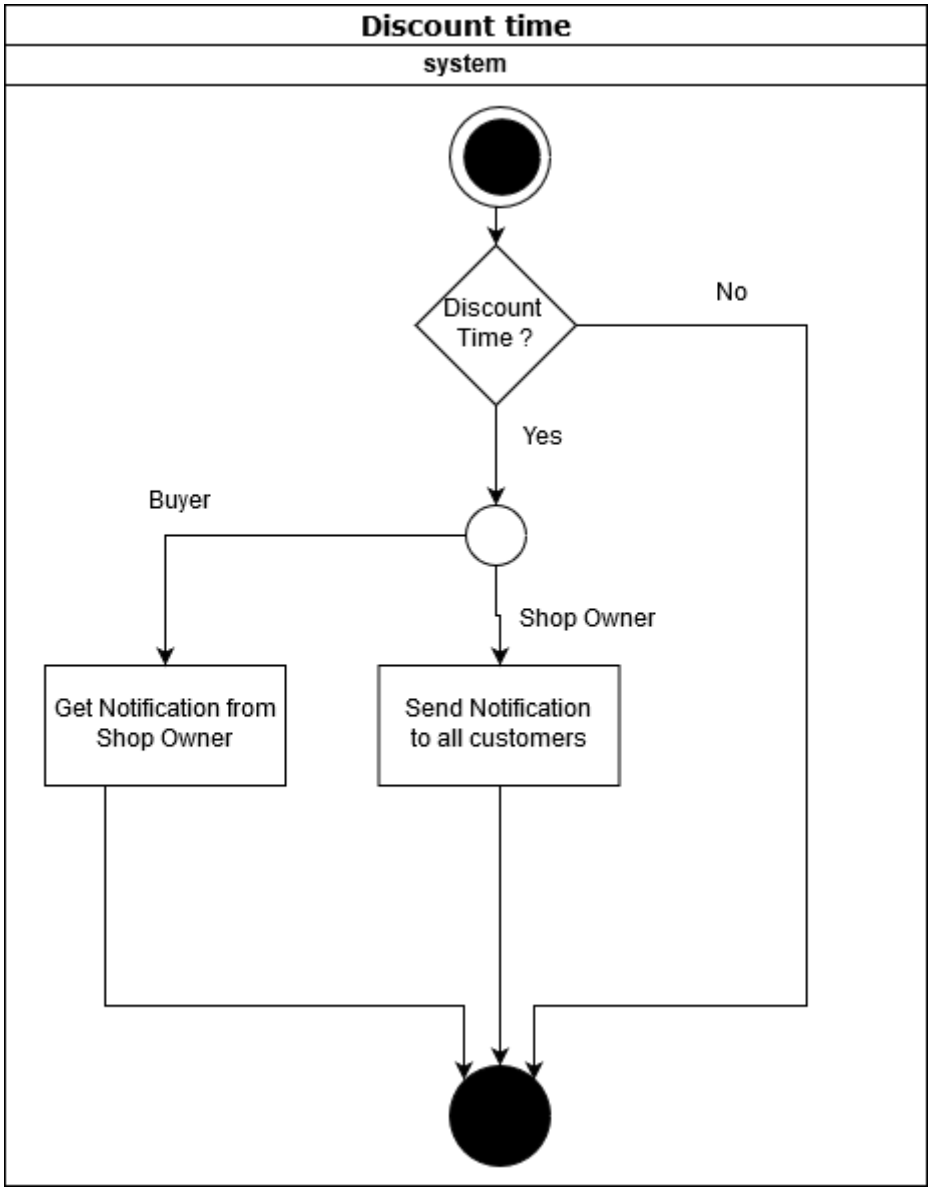


Figure 51: Swim Lane Diagram 1.7.2 - Discount Time

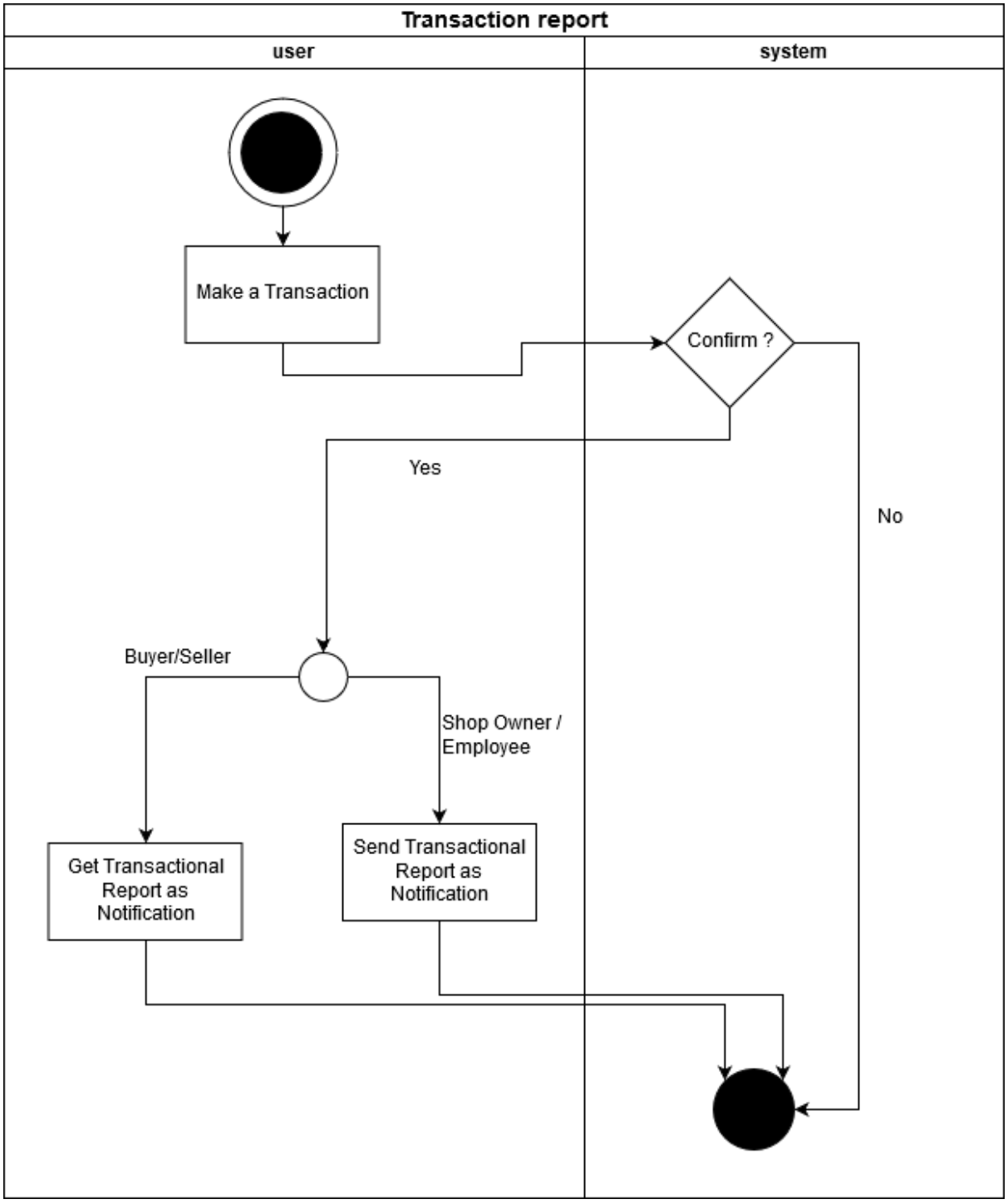


Figure 52: Swim Lane Diagram 1.7.3 - Transaction Report

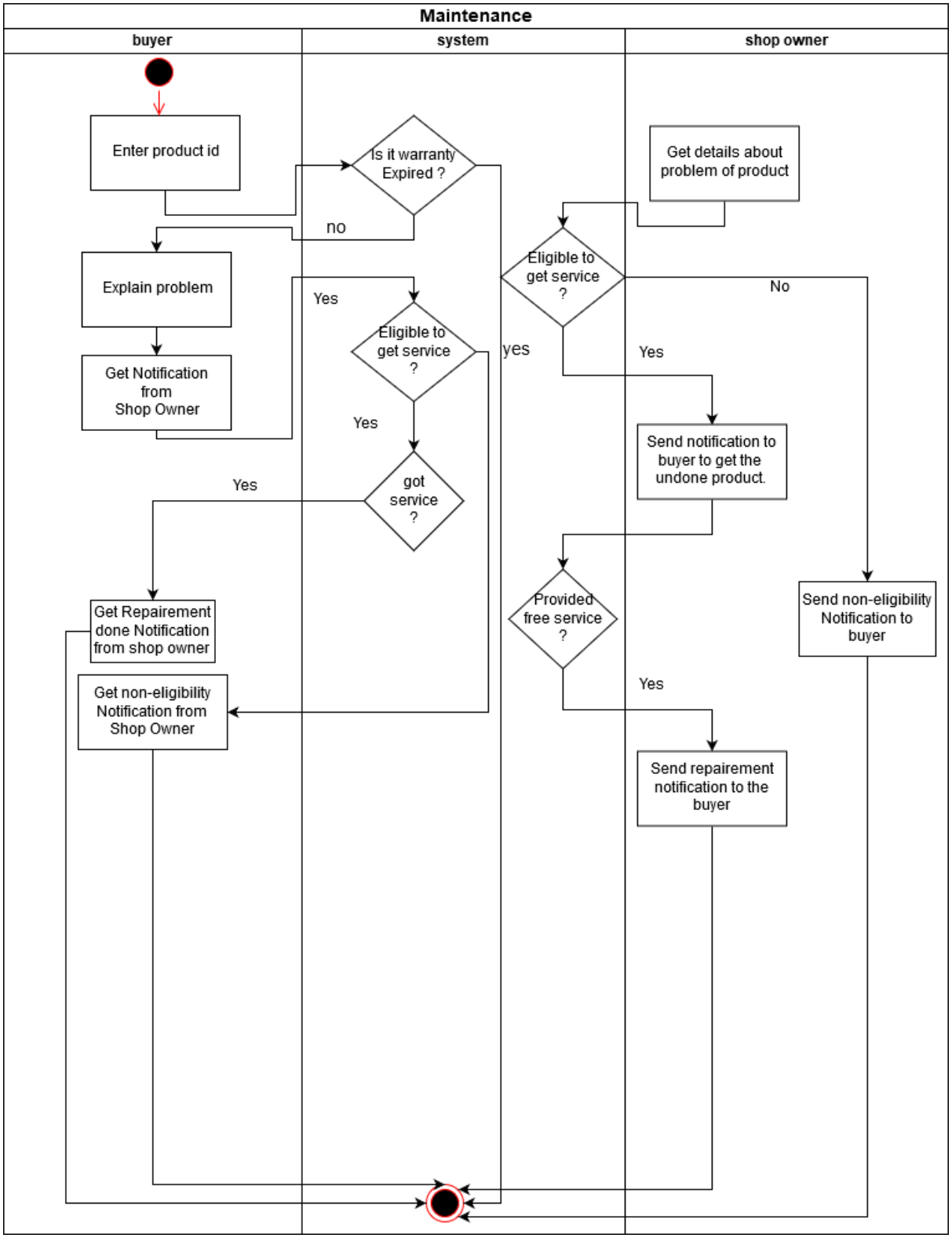


Figure 53: Swim Lane Diagram 1.7.4 - Maintenance Log

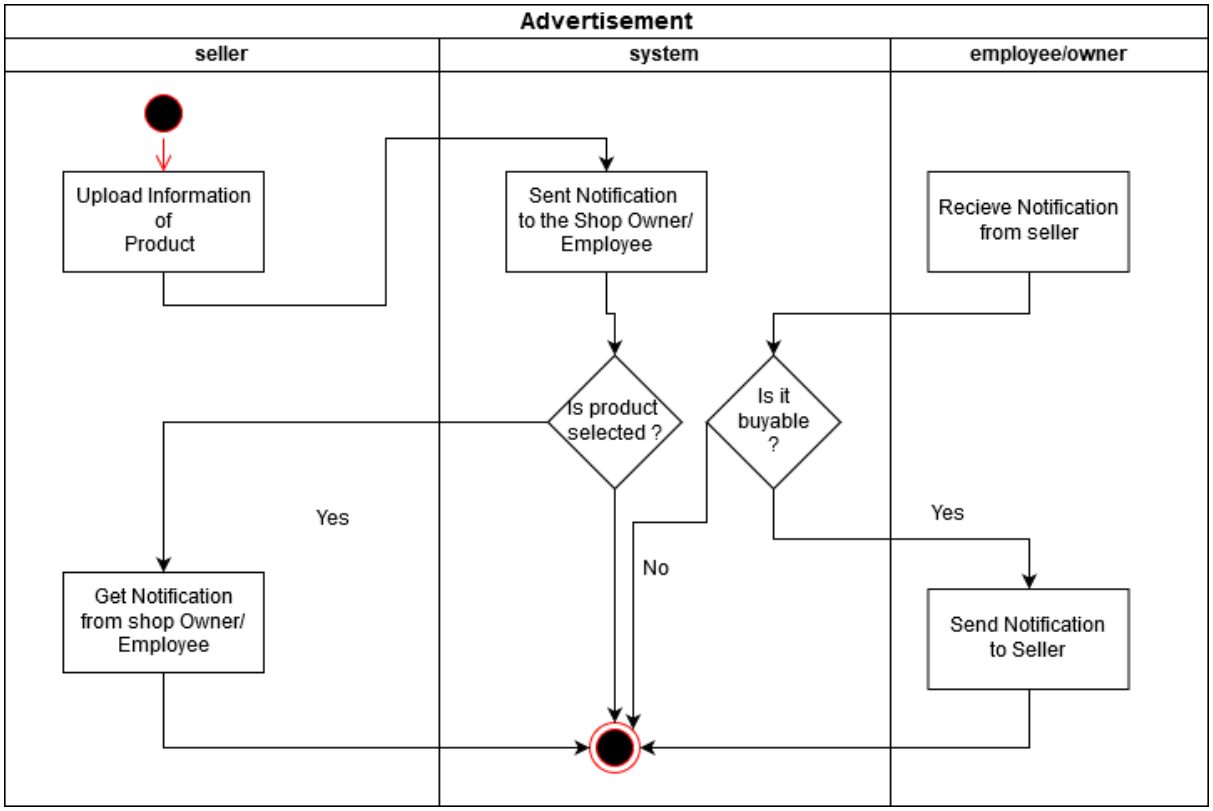


Figure 54: Swim Lane Diagram 1.7.5 - Advertisement

Chapter 5

Data Modeling

5.1 Data modeling concept

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

5.2 Data objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means an information that has a number of different properties or attributes. 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

All the nouns in the scenario were identified.

Table 1: Noun identification for data modeling

No	Noun	S/P	Attributes of
1.	Authentication	S	
2.	Vehicle	S	
3.	Bike	S	2,20
4.	System	S	
5.	User	S	
6.	Car	S	2,20
7.	Account	S	
8.	Interface	S	
9.	First name	S	5,22,23,24,25,96
10.	Last name	S	5,22,23,24,25,96
11.	Email id	S	5,22,23,24,25,96
12.	Address	S	5,22,23,24,25,96
13.	Mobile number	S	5,22,23,24,25,96
14.	Password	S	5,22,23,24,25,96
15.	Account type	S	5,22,23,24,25,96
16.	Option	P	
17.	Block time	S	1
18.	Chance	P	

19.	Information	P	
20.	Product	S	
21.	Transaction	S	
22.	Shop owner	S	
23.	Employee	S	
24.	seller	s	
25.	Customer	S	
26.	Name	S	
27.	Second	P	
28.	Choice	P	
29.	Actor	P	
30.	Equipment	S	
31.	Date range	S	
32.	Transaction_id	S	21
33.	Transaction_type	S	21
34.	Customer_id	S	21
35.	Customer_type	S	21
36.	Product_type	S	21
37.	Product_id	S	21
38.	Date	S	21
39.	Time	S	21

40.	Paid_amount	S	21
41.	Rest_amount	S	21
42.	Seller_id	S	21
43.	Seller_type	S	21
44.	Website	P	
45.	Order_list	S	104
46.	percentage	S	3,6,2
47.	Amount	S	
48.	shop	P	
49.	Brand name	S	3,6,2
50.	Model	S	3,6,2
51.	Original_buying_year	S	3,6,2
52.	Fuel	S	3,6,2
53.	Engine	S	3,6,2
54.	Edition_of_the_model	S	3,6,2
55.	Speed	S	3,6,2
56.	Price_condition	S	3,6,2
57.	Color	S	3,6,2
58.	Feature	S	3,6,2
59.	Exterior	P	
60.	Interior	P	

61.	Image	S	3,6,2
62.	Notification	S	
63.	Report	S	
64.	Repairmen	P	
65.	Warranty_date	S	103
66.	Problem_description	S	103
67.	Buyer_name	S	103
68.	Several_reason	P	
69.	Ordered_product	S	
70.	Preorder	P	
71.	Discount_time	S	3,6,2
72.	Offer	P	
73.	Discount	P	
74.	Business	P	
75.	Message	P	
76.	Transactional_information	S	62
77.	Transactional_report	S	62
78.	Maintenance_log	S	62
79.	Free_servicing	S	
80.	Equipment_list	S	30
81.	Category	S	30

82.	Quantity	S	30
83.	Unit_price	S	30
84.	Equipment_id	S	30
85.	equipment_name	S	30
86.	Management	P	
87.	Description	S	
88.	Condition_of_the_car	S	6
89.	Price	S	3,6,2
90.	Image_of_the_car	S	6
91.	Fields	P	
92.	Store	P	
93.	Database	S	
94.	Advertisement	S	
95.	Section	P	
96.	Dealer	S	87
97.	Opportunity	P	
98.	Review	S	
99.	Popularity	S	
100.	Condition_of_the_bike	S	3
101.	Image_of_the_bike	S	3
102.	Bike_id	S	3

103.	maintenance	s	
104.	order	s	
105	sign in	s	1
106	sign up	s	1
107	income	s	21
108	profit calculation	s	21
109	user_id	s	5,22,23,24,25,96
110	car_id	s	3
111	notification_id	s	62

5.2.2 Potential data objects

- User: 9,10,11,12,13,14,15
- Shop owner: 9,10,11,12,13,14,15
- Employee: 9,10,11,12,13,14,15
- Seller: 9,10,11,12,13,14,15
- Customer: 9,10,11,12,13,14,15
- Dealer: 9,10,11,12,13,14,15
- Transaction: 32,33,34,35,36,37,38,39,40,41,42,43
- Vehicle:46,49,50,51,52,53,54,55,56,57,58,61,71,89
- Car: 46,49,50,51,52,53,54,55,56,57,58,61,71,89,88,90
- Bike: 46,49,50,51,52,53,54,55,56,57,58,61,71,89,100,101,102
- Notification: 76,77,78
- Maintenance:65,66,67

- Order: 45
- income:21
- profit calculation:21
- equipment:80,81,82,83,84,85

5.2.3 Analysis for finalizing Data objects

- shop owner, employee, dealer and customer are users of trading of old motor vehicles system. so, all four kinds of users can be merged into a user data object.
- income and profit calculation can be merged into transaction
- all other data objects can be used as data object as they have enough importance in the system.

5.2.4 Final Data objects

Table 2: Final Data Objects

1	User: user_id, first name, last name, email, phone, password, address, account type
2	Shop owner: user_id
3	Employee: user_id
4	customer: type, user_id
5	dealer: type, user_id,description

6	vehicle: brand_name, model, originalBuyingYear, fuel, engine, editionOfTheModel, speed, price_condition, color, feature, image, discount_percentage, discount_time, price, product_id
7	car: car_id, brand_name, model, originalBuyingYear, fuel, engine,editionOfTheModel, speed, price_condition, color, feature, imageOfTheCar, discount_percentage, discount_time, price
8	bike: bike_id, brand_name, model, originalBuyingYear , fuel, engine, editionOfTheModel, speed, price_condition, color, feature, imageOfTheBike, discount_percentage, discount_time, price
9	notification: notification_id
10	equipment: equipment_id, quantity, unit_price, equipment_name
11	transaction: transaction_id, transaction_type, vehicle_id, vehicle_type, date, time, paid_amount, rest_amount, seller_id, seller_type

5.2.5 Data Object Relations

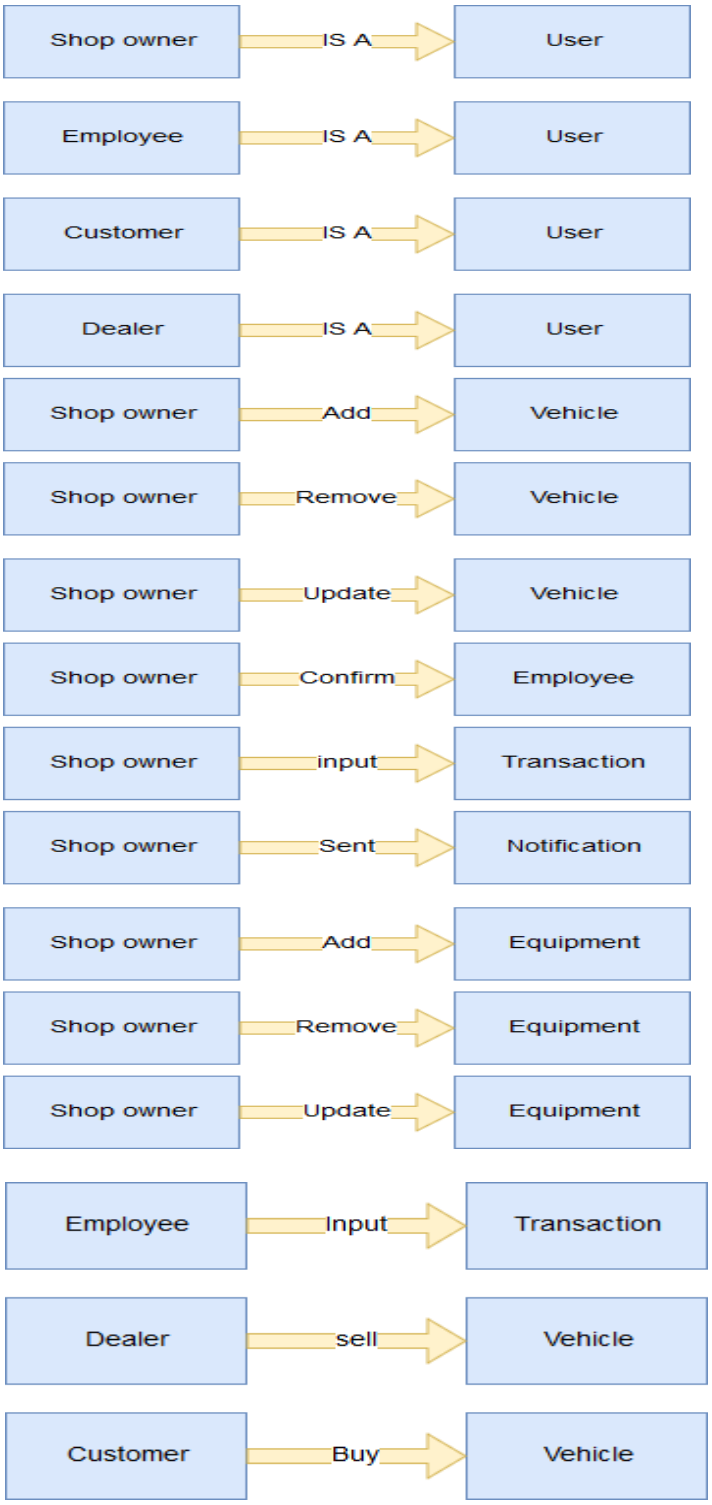


Figure 55: Relation Between Data Objects

5.3 Entity Relationship Diagram

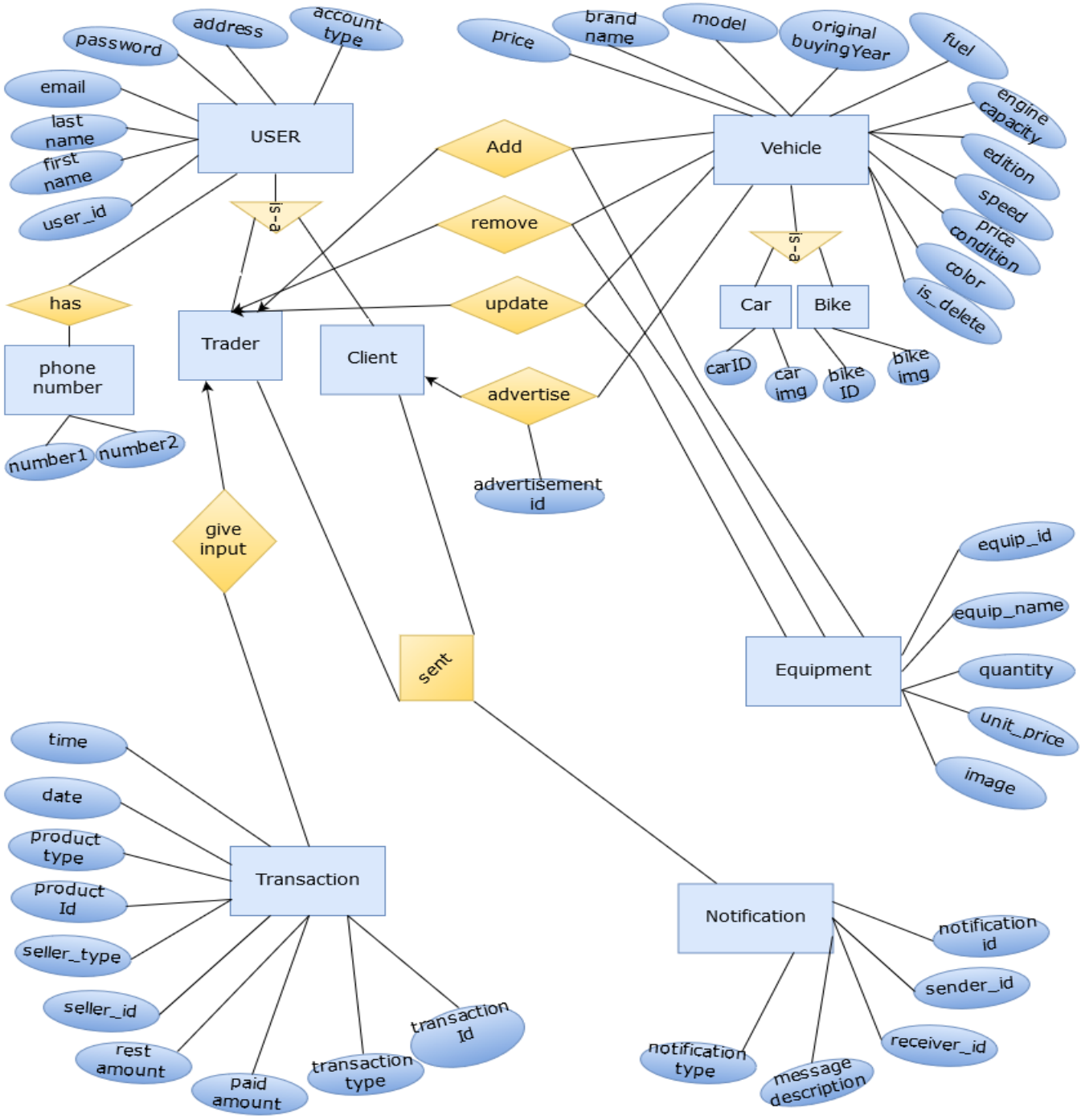


Figure 56: ER diagram of TOMVS

5.4 Schema Diagram

Table 3: Schema for User

User		
Attributes	Type	Size
User_id	NUMBER	30
FirstName	VARCHAR2	15
LastName	VARCHAR2	15
Email	VARCHAR2	30
Password	VARCHAR2	30
Address	VARCHAR2	50
account_type	VARCHAR2	10

Table 4: Schema for User-Phoner

User-Phone

Attributes	Type	Size
User Id	NUMBER	30
Phone_number	VARCHAR2	11

Table 5: Schema for Shop Owner

Shop Owner		
Attributes	Type	Size
User Id	NUMBER	30

Table 6: Schema for Employee

Employee		
Attributes	Type	Size
User Id	NUMBER	30

Table 7: Schema for Customer

Customer		
Attributes	Type	Size

User Id	NUMBER	30

Table 8: Schema for Seller/Dealer

Seller/Dealer		
Attributes	Type	Size
User Id	NUMBER	30

Table 9: Schema for Vehicle

Vehicle		
Attributes	Type	Size
Brand_name	VARCHAR2	30
Model	VARCHAR2	30
Original_buying_year	NUMBER	30
fuel	VARCHAR2	10
engine_capacity	NUMBER	10
edition	VARCHAR2	15
speed	NUMBER	15
price_condition	VARCHAR2	15

color	VARCHAR2	15
is_deleted	VARCHAR2	10
image	BLOB	<256 bytes

Table 10: Schema for Car

Car		
Attributes	Type	Size
car_id	NUMBER	30
car_image	BLOB	<256 bytes

Table 11: Schema for Bike

Bike		
Attributes	Type	Size
bike_id	NUMBER	30
bike_image	BLOB	<256 bytes

Table 12: Schema for Equipment

Equipment

Attributes	Type	Size
equip_id	NUMBER	30
equip_name	VARCHAR2	30
quantity	NUMBER	20
unit_price	NUMBER	20
image	BLOB	<256 BYTES

Table 13: Schema for Customer - Notification

Notification		
Attributes	Type	Size
notification_id	VARCHAR2	10
sender_id	VARCHAR2	10
receiver_id	NUMBER	10
message_description	VARCHAR2	80
notification_type	VARCHAR2	10

Table 14: Schema for Transaction

Transaction		
Attributes	Type	Size
transaction_id	NUMBER	10

transaction_type	VARCHAR2	10
paid_amount	NUMBER	20
rest_amount	NUMBER	20
seller_id	NUMBER	10
seller_type	NUMBER	20
product_id	NUMBER	10
product_type	VARCHAR2	20
date	DATE	30
time	TIME	30

Table 15: Schema for Car Addition

Car addition		
Attributes	Type	Size
User_id	NUMBER	10
car_id	NUMBER	10

Table 16: Schema for Car Remove

Car Remove		
Attributes	Type	Size

User_id	NUMBER	10
car_id	NUMBER	10

Table 17: Schema for Update

Car Update		
Attributes	Type	Size
User_id	NUMBER	10
car_id	NUMBER	10

Table 18: Schema for Bike addition

Bike addition		
Attributes	Type	Size
User_id	NUMBER	10
bike_id	NUMBER	10

Table 19: Schema for Bike Remove

Bike Remove		
Attributes	Type	Size
User_id	NUMBER	10
bike_id	NUMBER	10

Table 20: Schema for Bike Update

Bike Update		
Attributes	Type	Size

User_id	NUMBER	10
bike_id	NUMBER	10

Table 21: Schema for Advertisement

Advertise Vehicle		
Attributes	Type	Size
vehicle_id	NUMBER	10
Brand_name	VARCHAR2	30
Model	VARCHAR2	30
Original_buying_year	NUMBER	30
fuel	VARCHAR2	10
engine_capacity	NUMBER	10
edition	VARCHAR2	15
speed	NUMBER	15
price_condition	VARCHAR2	15
color	VARCHAR2	15
is_deleted	VARCHAR2	10
Dealer_id	NUMBER	10
image	BLOB	<256 bytes

Table 22: Schema for Equipment Addition

Equipment addition		
Attributes	Type	Size
User_id	NUMBER	10
equipment_id	NUMBER	10

Table 23: Schema for Equipment Remove

Equipment remove		
Attributes	Type	Size
User_id	NUMBER	10
equipment_id	NUMBER	10

Table 24: Schema for Equipment Update

Equipment Update		
Attributes	Type	Size
User_id	NUMBER	10
equipment_id	NUMBER	10

Chapter 6

Class Based Modeling

This chapter is intended to describe class based modeling of the Trading of Old Motor Vehicle 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 Classification

To identify the potential classes, nouns were selected 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.

Table 25: General Classification of nouns

No	Noun	GC
1.	Authentication	3,5
2.	Vehicle	2,5,7
3.	Bike	2,5,7
4.	System	3,5,6
5.	User	4,5,7
6.	Car	2,5,7
7.	Account	5
8.	Interface	5,6
9.	First name	

10.	Last name	
11.	Email id	
12.	Address	
13.	Mobile number	
14.	Password	
15.	Account type	
16.	Block time	
17.	Product	2,5,7
18.	Transaction	3,5
19.	Shop owner	4,5,7
20.	Employee	4,5,7
21.	seller	4,5,7
22.	Customer	4,5,7
23.	Name	
24.	Equipment	2,5,7
25.	Date range	
26.	Transaction_id	
27.	Transaction_type	
28.	Customer_id	
29.	Customer_type	
30.	Product_type	

31.	Product_id	
32.	Date	
33.	Time	
34.	Paid_amount	
35.	Rest_amount	
36.	Seller_id	
37.	Seller_type	
38.	Order_list	2,5,7
39.	percentage	
40.	Amount	
41.	Brand name	
42.	Model	
43.	Original_buying_year	
44.	Fuel	
45.	Engine	
46.	Edition_of_the_model	
47.	Speed	
48.	Price_condition	
49.	Color	
50.	Feature	
51.	Image	

52.	Notification	2,3,5,6
53.	Report	3
54.	Warranty_date	
55.	Problem_description	
56.	Buyer_name	
57.	Ordered_product	
58.	Discount_time	
59.	Transactional_information	
60.	Transactional_report	
61.	Maintenance_log	
62.	Free_servicing	1,3
63.	Equipment_list	
64.	Category	
65.	Quantity	
66.	Unit_price	
67.	Equipment_id	
68.	Equipment_information	
69.	Description	
70.	Condition_of_the_car	
71.	Price	
72.	Image_of_the_car	

73.	Database	1,2,5,6
74.	Advertisement	2,3,5,6
75.	Dealer	4,5,7
76.	Vehicle Management	3,5,7
76.	Review	3
77.	Popularity	
78.	Condition_of_the_bike	
79.	Image_of_the_bike	
80.	Bike_id	
81.	maintenance	1,3
82.	order	3
83	sign in	3,5
84	sign up	3,5
85	income	
86	profit calculation	3
87	user_id	
88	car_id	
89	notification_id	
90	search	3,5,7

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

Table 26: Selection Criteria of Potential Classes

No	Noun	SC
1.	Authentication	1,2,3
2.	Vehicle	2,3,4,5
3.	Bike	2,3,4,5
4.	System	1
5.	User	1,2,3,4,5
6.	Car	2,3,4,5
7.	Account	1
8.	Interface	
9.	Product	2,3,4,5

10.	Vehicle Management	1,3
11.	Transaction	1,3,4
12.	Shop Owner	1,2,3,4,5
13.	Employee	1,2,3,4,5
14.	Seller	1,2,3,4,5
15.	Customer	1,2,3,4,5
16.	Equipment	2,3,4,5
17.	Notification	1,2,3
18.	Report	2,3
19.	Free servicing	
20.	Database	1,6
21	Search	1,2,3
22	Advertisement	2,3
23.	Dealer	1,2,3,4,5
24.	Review	2
24.	Maintenance	6

25.	Order	1,2
26.	Sign in	
27.	Sign up	
28.	Profit calculation	
29.	Ordered list	2,3

6.4 Attribute Selection

Table 28: Attribute Selection of Classes

No	Class	Attribute
1	Authentication	userId RecoveryPin Locktime stayIdleTime

2	Trader	firstName lastName emailID Address Phone Password Type logInstatus traderID
3	Shop Owner	
4	Employee	
5	Client	firstName lastName emailID Address Phone Password Type logInstatus clientID
6	Dealer	
7	Customer	

8	Search	found
9	Notification	notificationId traderID clientID
10.	Transaction	transactionID traderID clientID Date Time productID amountPaid amountRest
11	Product	productID productType brandName Model Price priceCondition imageOfProduct isDeleted isPreOrdered advertisedID

12	Car	originalBuyingYear Fuel Engine editionOfTheModel Speed color
13	Bike	originalBuyingYear Fuel Engine editionOfTheModel Speed color
14	Equipment	Quantity Category addionalIformation

6.5 Methods Identification:

Table 29: Methods of Classes

No	Class	Methods
1	Authentication	signUp() isValid() isVerified() generateUserID()

		signIn() signOut() automaticallySignOut() deleteAccount() changePassword() accountRecovery() sendRecoveryPin() accountLocked()
2	Trader	addProduct() removeProduct() updateProductInformation() searchProduct() searchUser() searchTransaction() getAdvertisedProduct() removeFromAdvertisementSection(). sendNotification() getNotification()
3	Shop Owner	approveUser() deleteUser() profitCalculation() getProductProblemToBeRepaired() sendDoneProductService() sendDiscountMessege()

4	Employee	
5	Client	reviewProduct() searchProduct() searchTransaction()
6	Dealer	advertiseProduct() deleteFromAdvertisement() sellProduct()
7	Customer	orderProduct() deleteOrder() buyProduct() preOrderOfProduct() submitProductProblem() getService()
8	Search	searchProduct() searchUser() searchTransaction()

9	Notification	isOrderedProductAvailable() getDiscountMessege() sendDiscountMessege() sendOrderMemo() getOrderMemo() sendMessegeAboutCompleteTransaction() getMessegeAboutCompleteTransaction() fixProblem() getProblemDescription() getServiceDone() sendServiceDone() sendAdvertisemens() getAdvertisement() getNewProductInAdvertisementList() getDeleteFromAdvertisementSection() getProductIsChosen()
10.	Transaction	buyProduct() sellProduct() orderProduct() createMemo() profitCalculation()

11	Product	addProduct() addProductAsAPreOrder() removeProduct() updateProductReview() getProductReview()
12	Car	addToCarAdvertisementList() deleteFromCarAdvertisementList()
13	Bike	addToBikeAdvertisementList() deleteFromBikeAdvertisementList()
14	Equipment	

6.6 Class Cards

Table 30: Class Card of Authentication

Authentication	
Attributes	Methods
userId RecoveryPin	signUp() isValid()

Locktime stayIdleTime	isVerified() generateUserID() signIn() signOut() automaticallySignOut() deleteAccount() changePassword() accountRecovery() sendRecoveryPin() accountLocked()
Responsibilities	Collaborator
Create Account	Trader, Client
Authenticate to the System	Trader, Client
Recover Account	Trader, Client

Table 31: Class Card of Trader

Trader	
Attributes	Methods
firstName lastName emailID Address Phone Password Type logInstatus traderID	addProduct() removeProduct() updateProductInformation() searchProduct() searchUser() searchTransaction() getAdvertisedProduct() removeFromAdvertisementSection(). sendNotification() getNotification()
Responsibilities	Collaborator
Product add/remove to storage	Product, Dealer Notification, Transaction
Manage Advertisement Section	Product
Update Product Information	Product
Search	Search
Exchange Notification	Notification, Client
Update Own Information	

Table 32: Class Card of Shop Owner

Shop Owner	
Attributes	Methods
	approveUser() deleteUser() profitCalculation() getProductProblemToBeRepaired() sendDoneProductService() sendDiscountMessege()
Responsibilities	Collaborator
Confirming Account Creation	Authentication, Employee, Client
Delete User	Authentication, Employee, Client
Getting the profit	Transaction
Maintenance Product	Customer, Product, Notification

Table 33: Class Card of Client

Client

Attributes	Methods
firstName lastName emailID Address Phone Password Type logInstatus clientID	reviewProduct() searchProduct() searchTransaction()
Responsibilities	Collaborator
Giving a feedback of product	Product
Product Search	Search
Transaction Search	Search

Table 34: Class Card of Dealer

Dealer	
Attributes	Methods

	advertiseProduct() deleteFromAdvertisement() sellProduct()
Responsibilities	Collaborator
Advertising of any Product	Product, Notification, Trader
Cancel of Advertisement	Product
Product sell	Trader, Transaction, Product, Notification

Table 35: Class Card of Customer

Customer	
Attributes	Methods
	orderProduct() deleteOrder() buyProduct() preOrderOfProduct() submitProductProblem() getService()
Responsibilities	Collaborator
Order Maintenance	

	Product, Notification, Trader, Transaction
Purchasing product	Product, Notification, Trader, Transaction
Report to System for product defeat.	Product, Notification, Shop Owner

Table 36: Class Card of Search

Search	
Attributes	Methods
found	searchProduct() searchUser() searchTransaction()
Responsibilities	Collaborator
Searching for Product, User and transaction	

Table 37: Class Card of Notification

Notification	
Attributes	Methods

notificationId traderID clientID	isOrderedProductAvailable() getDiscountMessege() sendDiscountMessege() sendOrderMemo() getOrderMemo() sendMessegeAboutCompleteTransaction() getMessegeAboutCompleteTransaction() fixProblem() getProblemDescription() getServiceDone() sendServiceDone() sendAdvertisemens() getAdvertisement() getNewProductInAdvertisementList() getDeleteFromAdvertisementSection() getProductIsChosen()
Responsibilities	Collaborator
Exchanges Notification between Traders and Clients after each transaction made.	Trader, Clients, Product, Transaction
To make a report about defeat product before expiring warranty date.	Product, Customer, Shop Owner

Table 38: Class Card of Transaction

Transaction	
Attributes	Methods

transactionID traderID clientID Date Time productID amountPaid amountRest	buyProduct() sellProduct() orderProduct() createMemo() profitCalculation()
Responsibilities	Collaborator
Storing Every Transactional Record.	Product, Trader, Client, Notification
Create a cash memo after every Transaction	Notification, Trader, Client, Product

Table 39: Class Card of Product

Product	
Attributes	Methods
productID productType brandName Model Price priceCondition imageOfProduct isDeleted isPreOrdered advertisedID	addProduct() addProductAsAPreOrder() removeProduct() updateProductReview() getProductReview()

Responsibilities	Collaborator
Managing Products attributes	Trader

Table 40: Class Card of Car

Car	
Attributes	Methods
originalBuyingYear Fuel Engine editionOfTheModel Speed color	addToCarAdvertisementList() deleteFromCarAdvertisementList()
Responsibilities	Collaborator
Managing advertisement section	Dealer, Car, Notification, Trader
Storing Information	Trader,Car

Table 41: Class Card of Bike

Bike	
Attributes	Methods
originalBuyingYear Fuel Engine editionOfTheModel Speed Color	addToBikeAdvertisementList() deleteFromBikeAdvertisementList()
Responsibilities	Collaborator
Managing advertisement section	Dealer, Bike, Notification, Trader
Storing Information	Trader,Bike

Table 42: Class Card of Equipment

Equipment

Attributes	Methods
Quantity Category addionalIformation	addToBikeAdvertisementList() deleteFromBikeAdvertisementList()
Responsibilities	Collaborator
Storing Information	Trader, Equipment

Chapter 7

Behavioral Modeling

7.1 State Transition Diagram

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

Identifying Events

Table 43: Event Identification

No	Event	Initiator	Collaborator
1	Sign up	User	Authentication
2	Sign up and validation check	Authentication	Verification. Message, Database
3	Request approval	Owner	Message, Database
4	Verify sign up	Verification	Database
5	Log In and lock account	Authentication	Verification, Activity Log
6	Match	Verification	Database
7	Recover password	User	Authentication
8	Password Recovery	Authentication	
9	Send password	Message	
10	Automatic Log Out	Authentication	Activity Log
11	Log out	Authentication	Activity Log
12	Recover password	Authentication	Verification, Message, Database
13	Log In	User	Authentication
14	Log out	User	Authentication

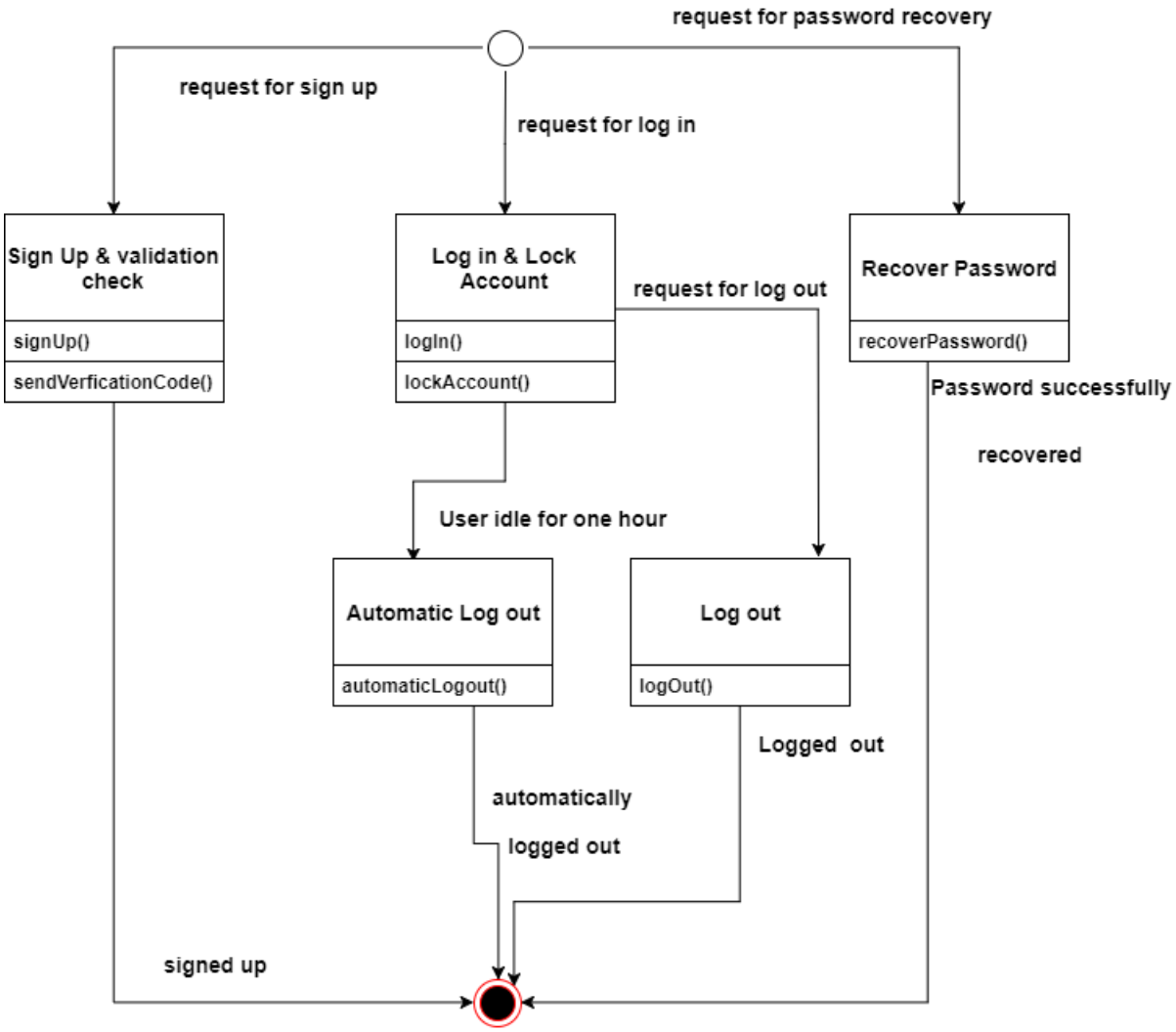


Figure 60: Authentication

7.2 Sequence Diagram of Trading of Old Motor Vehicles

Chapter 8

Data Flow Diagram

This chapter is intended to describe data flow diagram of the Trading of Old Motor Vehicles System.

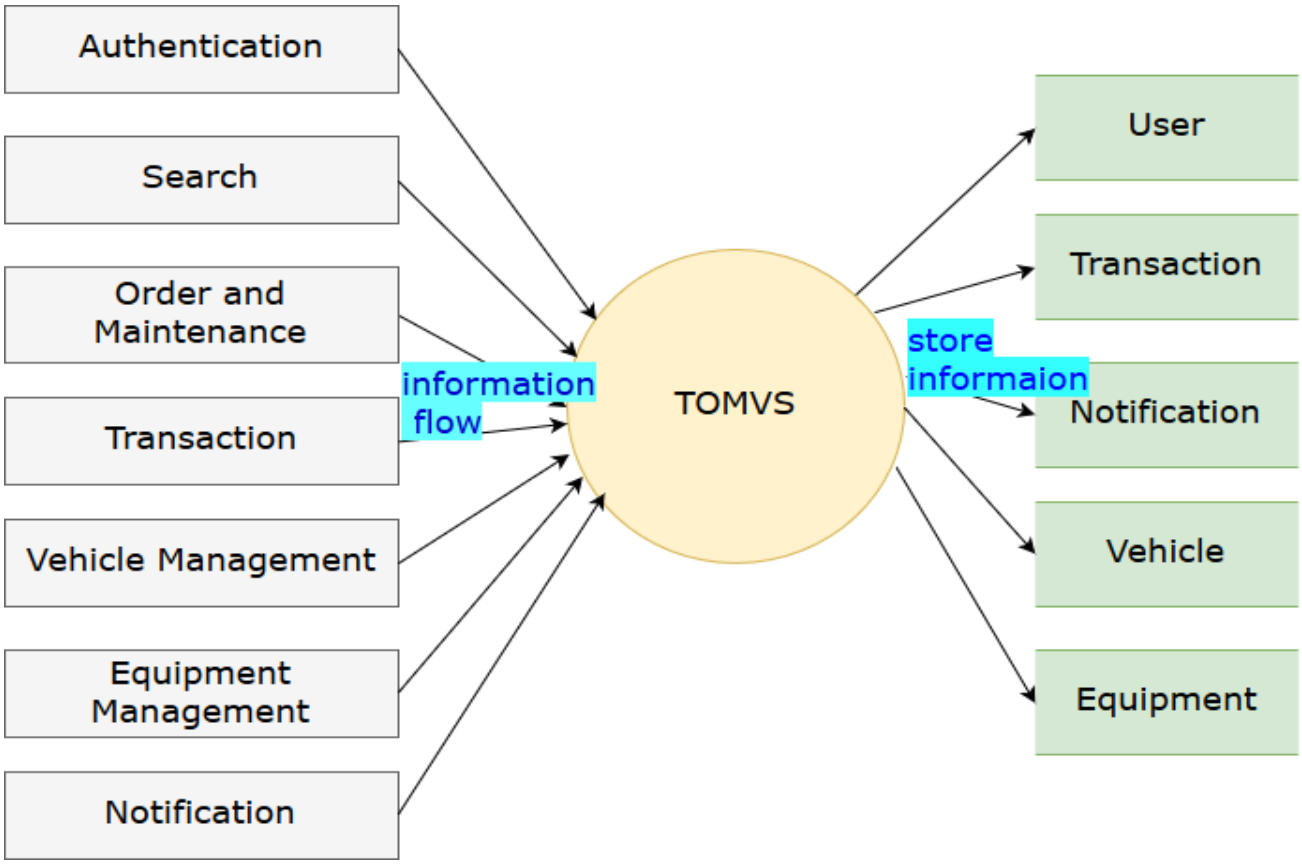
8.1 data flow diagram

A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system.

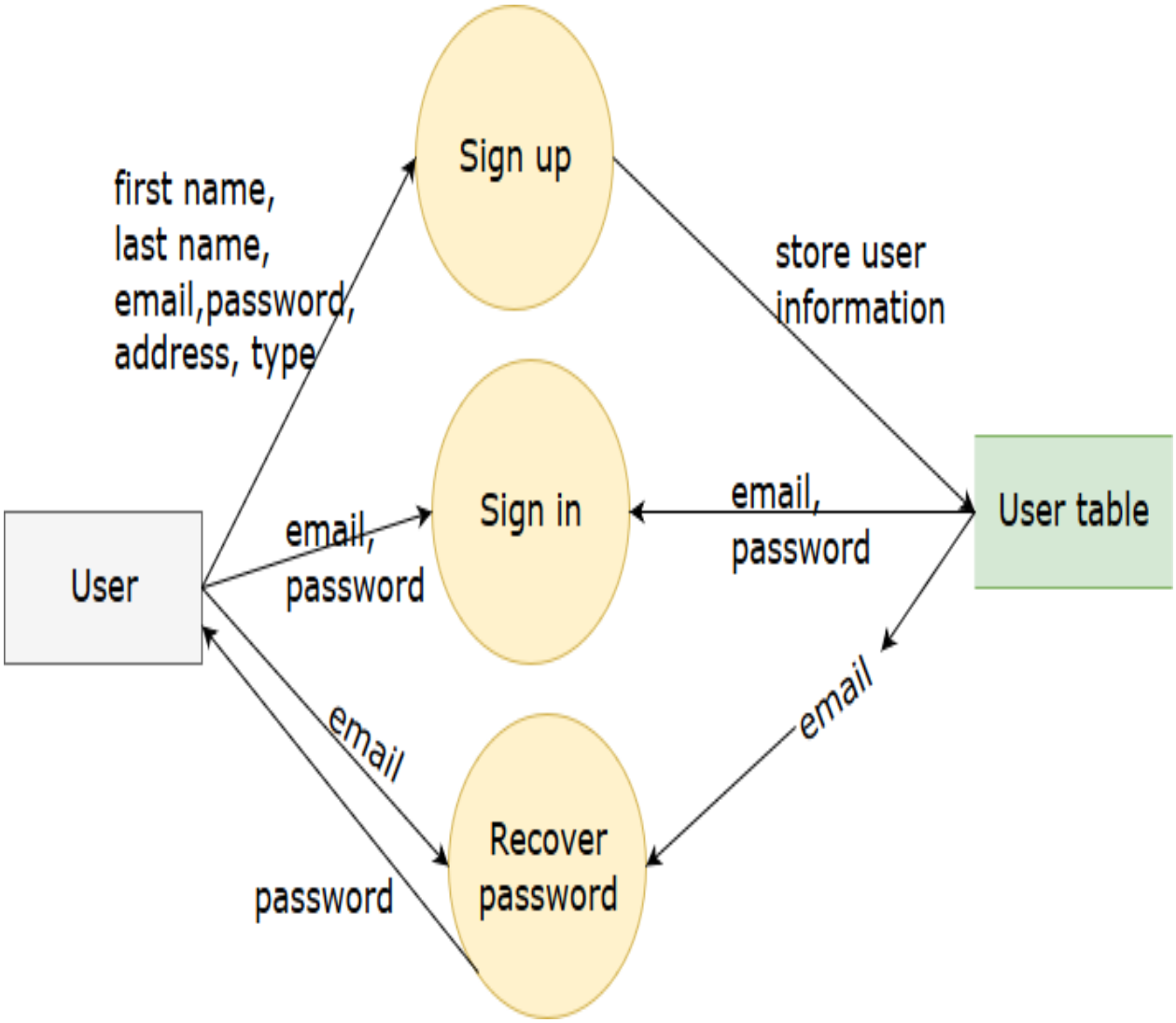
8.1 DFD of the Trading of Old Motor Vehicles (TOMVS)

There are four level data flow in the Trading of Old Motor Vehicles System. Level 0 is known as context level.

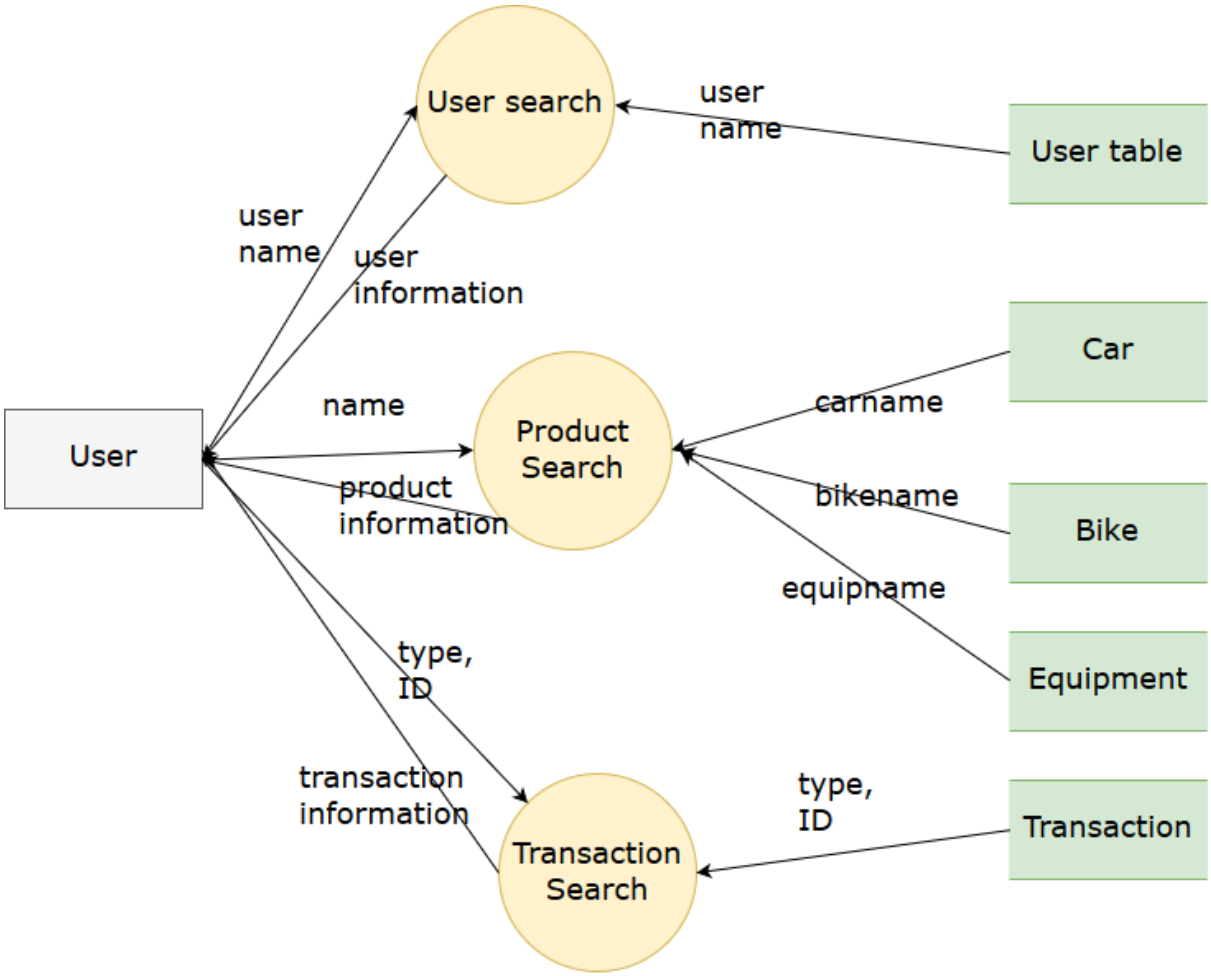
8.1.1 Level 0 (Context Level)



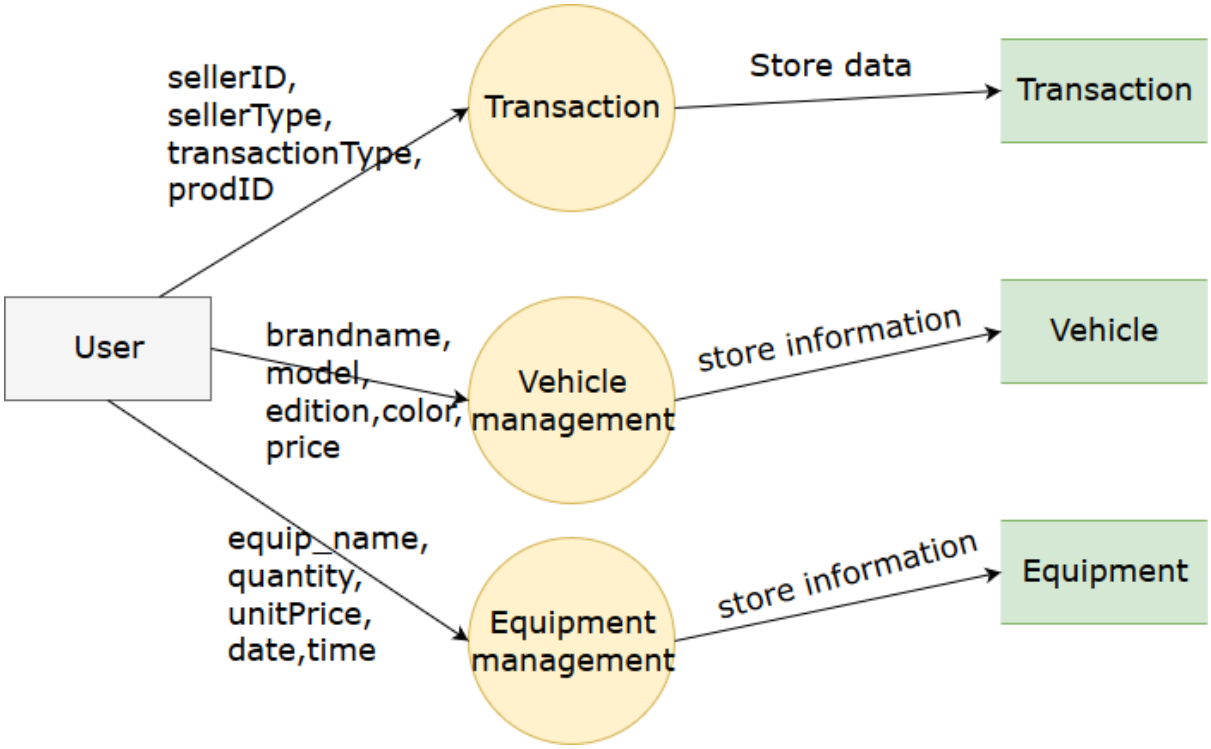
8.1.2 Level 1


Sign up
first name,
last name,
email,password,
address, type
store user
information
Sign in
email,
password
email,
password
User table
User
email
email
Recover
password
password

8.1.2 Level 2


User search
user
name
User table
user
name
user
information
Car
name
Product
Search
carname
User
product
information
bikename
Bike
equipname
type,
ID
Equipment
transaction
information
type,
ID
Transaction
Search
Transaction

8.1.3 Level 3


sellerID,
sellerType,
transactionType,
prodID
Transaction
Store data
Transaction
User
brandname,
model,
edition,color,
price
Vehicle
management
store information
Vehicle
equip_name,
quantity,
unitPrice,
date,time
Equipment
management
store information
Equipment

Chapter 9

Conclusion

From this SRS report on Trading of Old Motor Vehicles, the readers will get a clear and easy view of the overall system of management system of the online trade of motor vehicles. 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. Hopefully, this document can also help the junior BSSE students. We tried best to remove all dependencies and make an effective and fully designed SRS.

Chapter 10

References

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