The Ministry of Education of the Republic of Belarus

The institution of higher education

"THE BELARUSIAN STATE UNIVERSITY

OF INFORMATICS AND RADIO ELECTRONICS"

Faculty of Information Technology and Control

The department of information technology of automated systems

EXPLANATORY NOTE

to the course project

of the discipline "Computer Aided Software Systems Design"

on the topic of:

THE SECOND-HAND GOODS TRADING SYSTEM

Completed by: Student of gr. 920611
                                                               Li Boyi

The Course-project manager: Senior lecturer of the ITAS Dpt.
                                                               Khajynova N.V.

Minsk 2022

CONTENTS

INTRODUCTION

The development of the times, the progress of technology, the alternation of old and new items is changing day by day, every family, everyone will have updated items in the idle, how to deal with these items has also become a daily concern. Some people will get to the local second-hand market to trade, some people will be far from the second-hand market or second-hand items are not easy to carry and sell it as scrap or even throw away. This is not only bad for the environment, but also a waste of social resources.

With the popularity of the network, the traditional second-hand trading market also appeared in the network, due to the pervasive nature of network information, changing people's habits. Nowadays, from the traditional sale and purchase of second-hand goods to go to the second-hand market, to just open the computer or cell phone can be released throughout the country to sell second-hand goods or information for purchase, which is greatly convenient for the general public. The network also greatly promotes the development of second-hand transactions, the convenience of the network also makes more and more people are used to buying and selling second-hand items on the Internet, and no longer need to visit the traditional second-hand market. So where to post second-hand trading information in order to make the most of the convenience of the Internet? The company's main goal is to provide a platform for the release of information, and this is where the second-hand trading platform was born.

The birth of the second-hand trading platform makes second-hand trading no longer restricted to the second-hand marketplace. The convenience of the network allows most people to choose to publish second-hand trading information on the second-hand trading platform, without having to leave home to buy and sell second-hand items, which is very convenient for the people. It can be said that the second-hand trading network has greatly promoted the enthusiasm of the people to buy and sell second-hand goods, and also promoted the effective use of social resources.

As a used goods trading system. First of all, users can post used goods as sellers and buy goods as buyers, so user roles should have both buyer and seller roles. And the system should implement a "user communication subsystem" for communication between buyers and sellers, so that buyers and sellers can communicate in the communication system. The administrator can manage users, manage posted products, manage users' orders, etc. Users can also apply for arbitration of orders, and the administrator can arbitrate orders.

The purpose of this interdisciplinary paper is to show that an existing modeling language, Integrated Definition of Functional Modeling (IDEF0), is suitable for strategy modeling and automation of strategic plan development and implementation.

Data Flow Diagram (DFD) provide a straightforward and effective way for organizations to understand, refine, and implement new processes or systems. They are visual representations of your processes or systems, so they make it easy to understand and prune.

A database model shows the logical structure of a database, including the relationships and constraints that determine how data is stored and accessed. The design of a single database model is based on the rules and concepts of whichever broader data model the designer has adopted. Most data models can be represented by an accompanying database diagram.

The purpose of use case diagrams in UML is to show the different ways in which users may interact with the system.

Sequence diagrams are a dynamic modeling scheme in UML because they focus specifically on lifelines, or concurrent processes and objects, and the messages that are exchanged between them to perform a function before the end of the lifeline.

An activity diagram is essentially a flowchart that shows the activities performed by the system.

1. STRUCTURE DIAGRAM

1.1 Organizational Structure Diagram

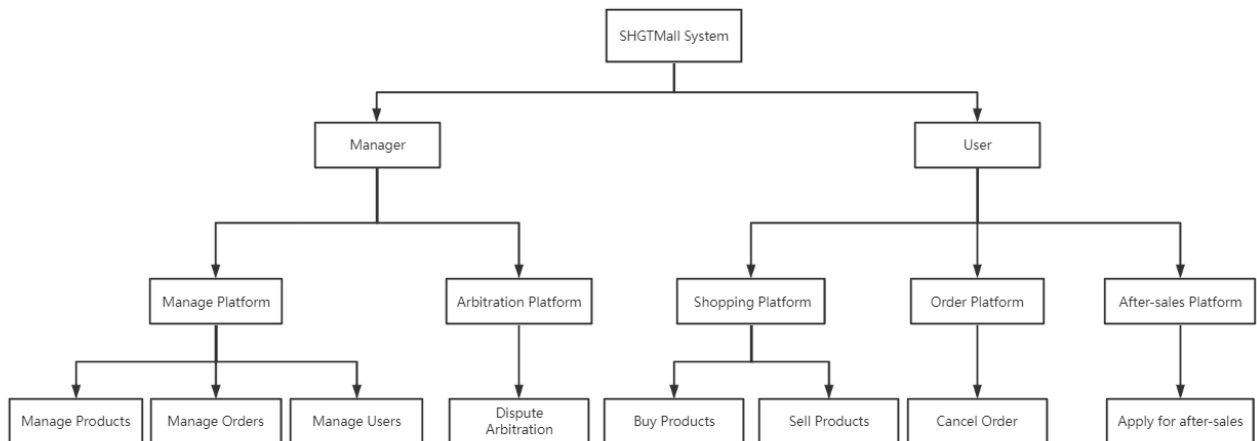The organizational structure for the second-hand goods trading system is shown as a diagram below.

Figure 1.1 - Organizational Structure diagram

This figure shows that the used goods trading system is divided into two subsystems, the first is the administrator system, the administrator in the management platform to manage products, orders, and users, and in the arbitration platform to arbitrate disputes. Users can trade products in the shopping platform, cancel orders placed in the order platform, and apply for after-sales service in the after-sales platform.

2. FUNCTIONAL DIAGRAMS

2.1 IDEF0 Diagram

Business process modeling, or IDEF0, models entire systems as a set of interrelated activities or functions so it can analyze the functions of a system independently of the objects performing those functions. IDEF0 utilizes only two graphical symbols: boxes and arrows.

IDEF0 uses activities and arrows to graphically describe and document business processes. To do this, it captures information about the business or process and displays the information and resources that are included in each step. IDEF0 activity modeling is best utilized as an analysis and logical design technique. As such, it is generally performed early in a project, and to provide analysis for the Process Flow Modeling (IDEF3) method for data collection and AS-IS process modeling [1].

IDEF0 models a system as a set of activities (functions) using only two graphic symbols: boxes and arrows.
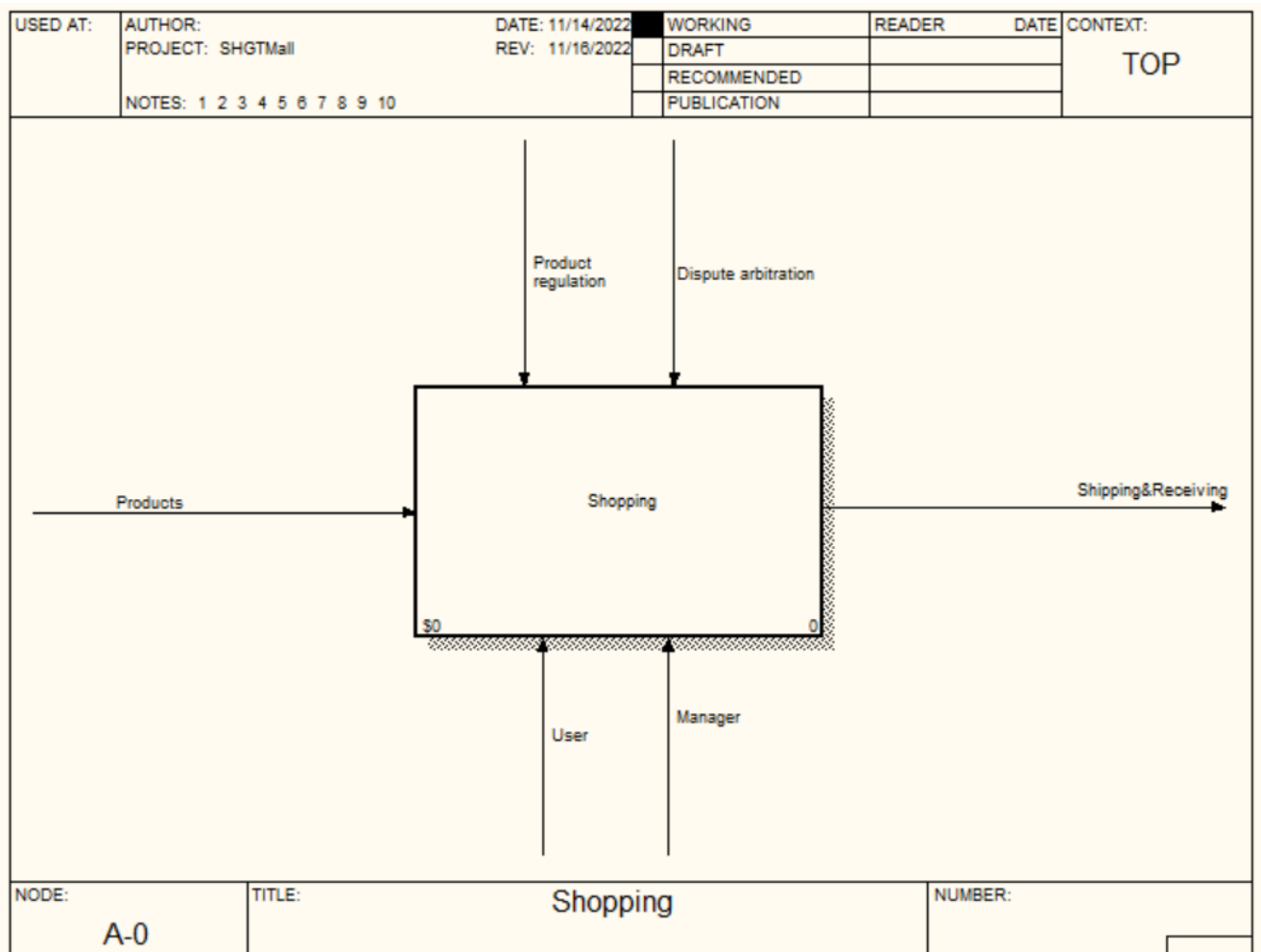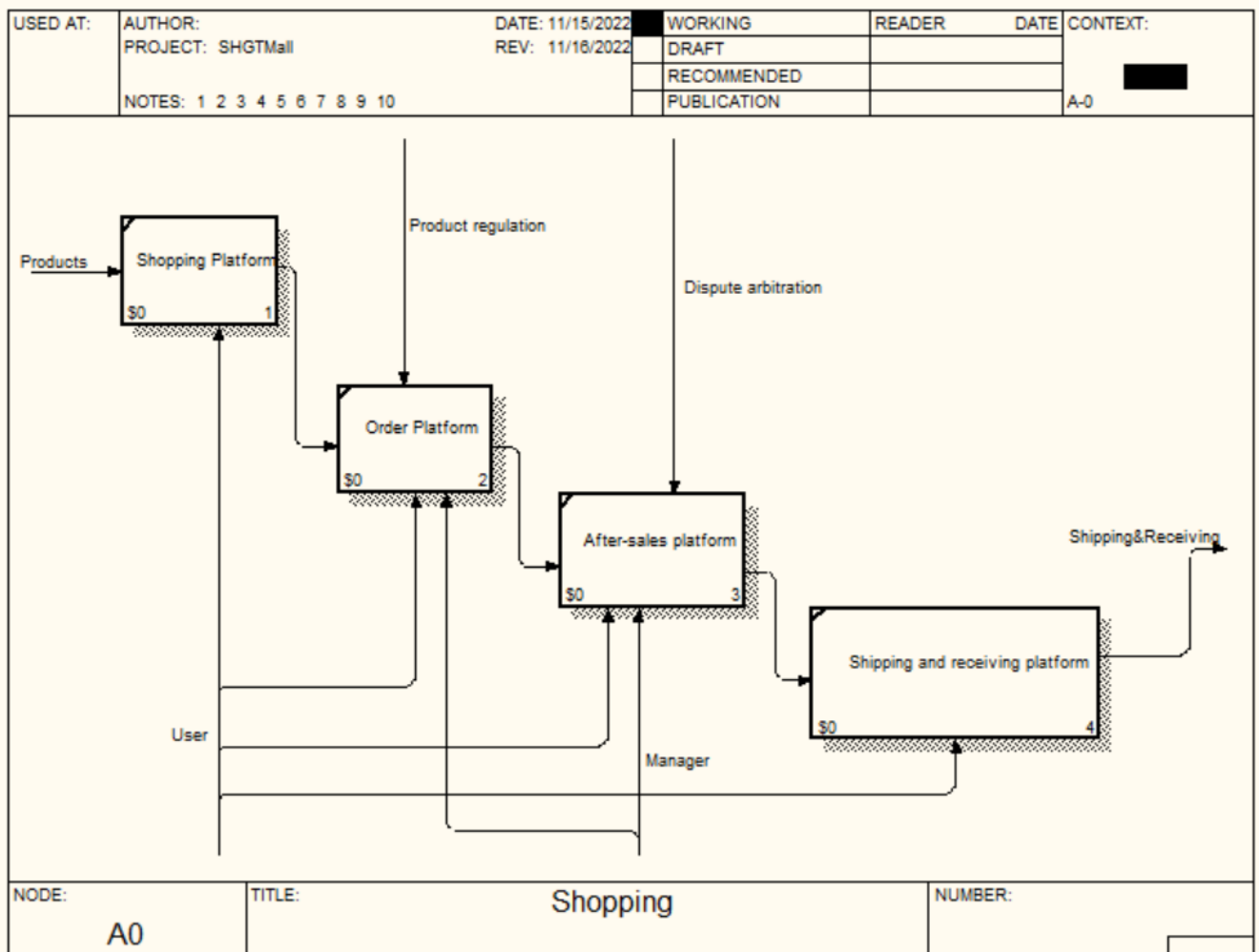

Figure 2.1 - IDEF0 Diagram

Figure 2.2 - IDEF0 Decomposition Diagram

Activities are represented by boxes containing a single, active verb plus a common noun that clarifies the objective of the activity from the viewpoint of the model (for example, Obtain Driver's License). You can use an adjective to further qualify the noun.

Arrows represent four types of information that are connected to an activity, and that are captured in IDEF0 models: Input arrow shows what an activity consumes or transforms. Output arrow shows what an activity produces or creates. Control arrow represents objects that govern the manner in which inputs are transformed but are not themselves transformed by the activity. Mechanism arrow represents objects that perform the transformation of inputs to outputs but are not themselves transformed by the activity.

In this used item trading system, there is an input named "product". Users and administrators are used on the mechanism arrows. Goods reconciliation and dispute arbitration are on the control arrow space. The output is just shipping and delivery.

The input of the goods is the shopping platform and then goes to the order platform. The order platform is under the control of the goods verification, then the after-sales platform, which is under the control of the arbitration of disputes, followed by the delivery platform, which directly outputs the shipment and receipt of goods. The user mechanism will be used in the shopping platform, the order platform, the after-sales platform, and the

9

shipping and receiving platform. The administrator mechanism will be used in the order platform and the after-sales platform.

2.2 DFD Diagram

Data flow diagrams, also known as DFD, are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.
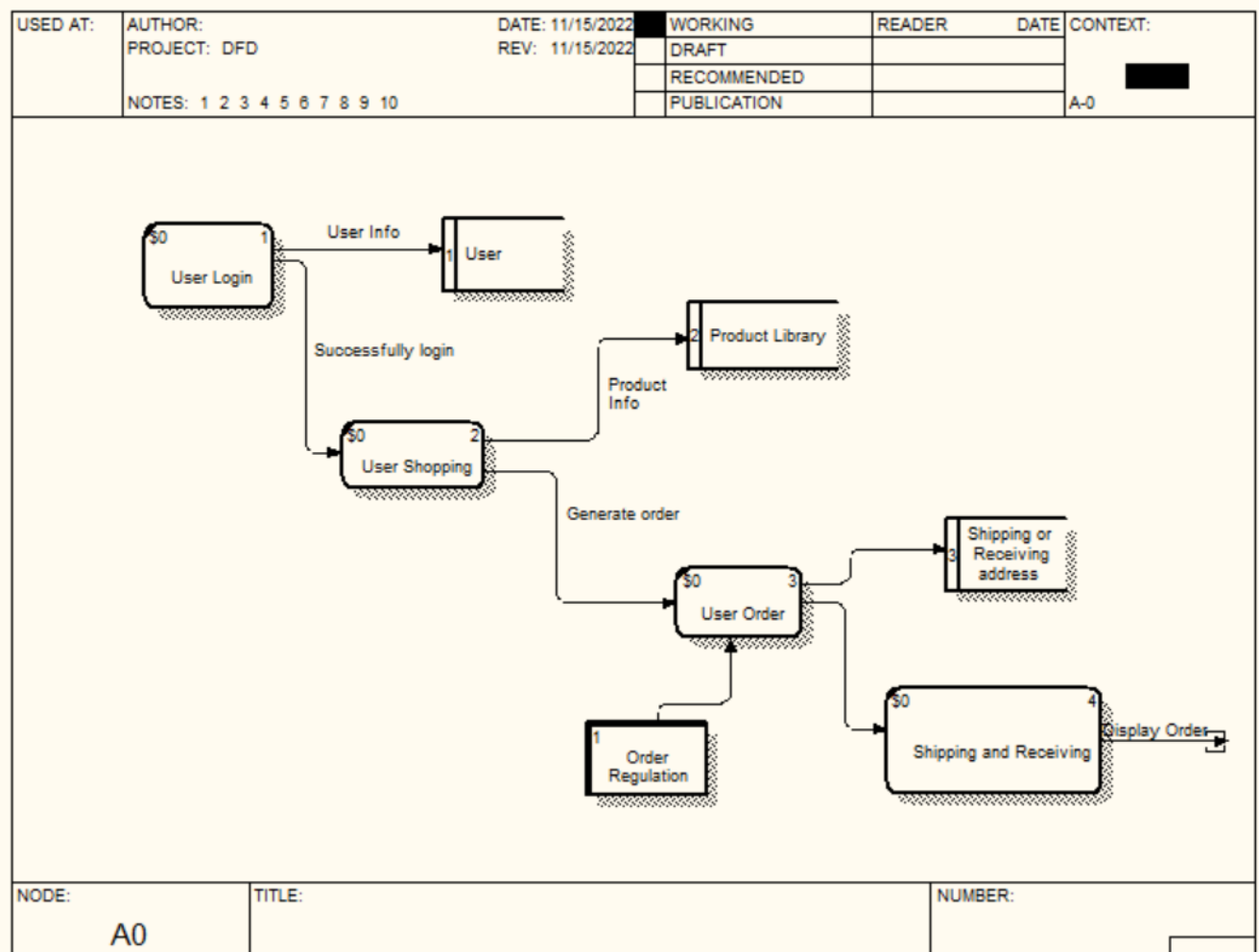

Figure 2.3 - DFD Diagram

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

The main process starts with a request for a transfer. At the end of the user process, the visitor's information is stored in the database. Then comes the user's process of purchasing goods. Some merchandise information will be collected in the merchandise database and at the same time enter the user order process, which is monitored by the

order check and the shipping and receiving information will be stored and modified in the database. Finally, it will enter the shipping and receiving phase. It will collect the shipping information and receiving information. Then it will display the order process, which can be terminated.

2.3 Database Model

This model captures the relationships between real-world entities much like the network model, but it isn't as directly tied to the physical structure of the database. Below the figure, there are detailed information about each table (field name, data type, field size, description).

There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database, data is collected into tables (called relations in relational theory).
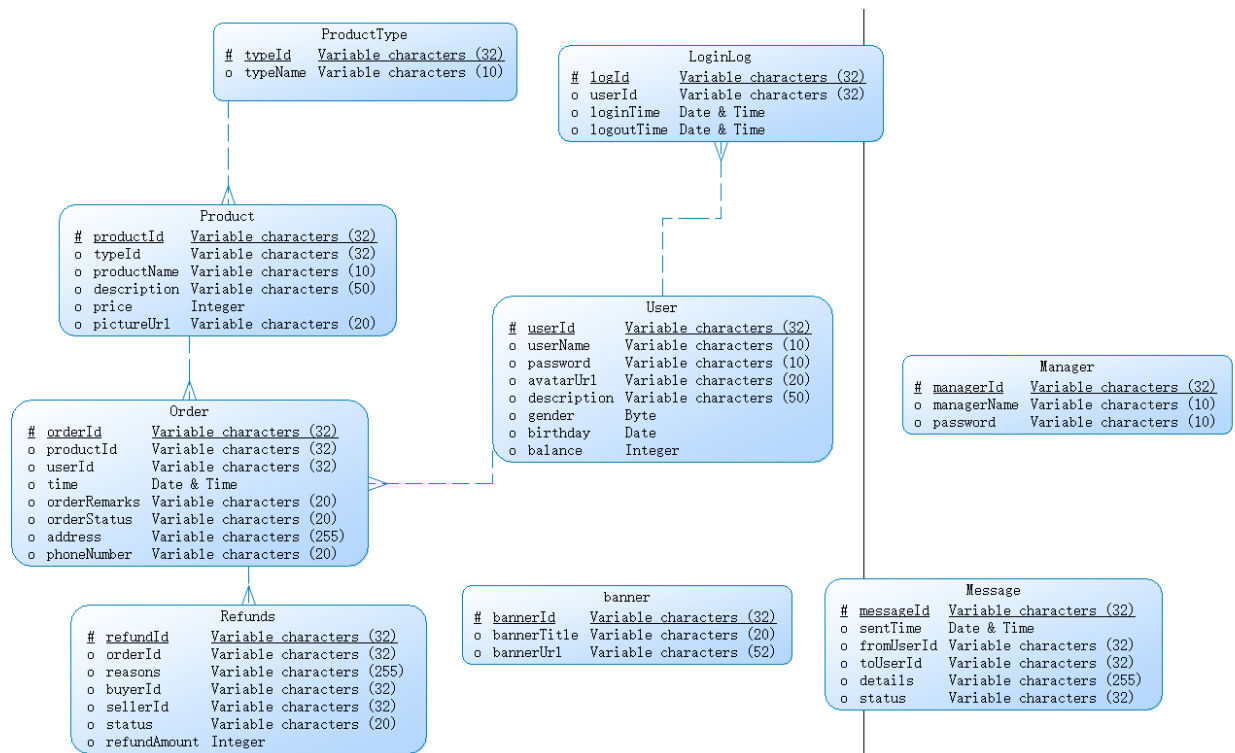

Figure 2.4 - Logical Diagram

For the logic model figure above, the following is related description.

Entity: There are 9 entities in the database, which are `ProductType`, `Order`, `Product`, `Refunds `, `banner`, `Message`, `Manager`, `User `, `LoginLog`. Detailed attributes can be seen in the Figure-1.4.

Attributes and Primary keys: They are illustrated in the figure, go see the figure for detail.

Relationships and Foreign Keys: In total, there are 5 relationships between entities. In `Order`, the `productId` is a foreign key related with the primary key of entity `Product`, which means that an item can exist in an order.
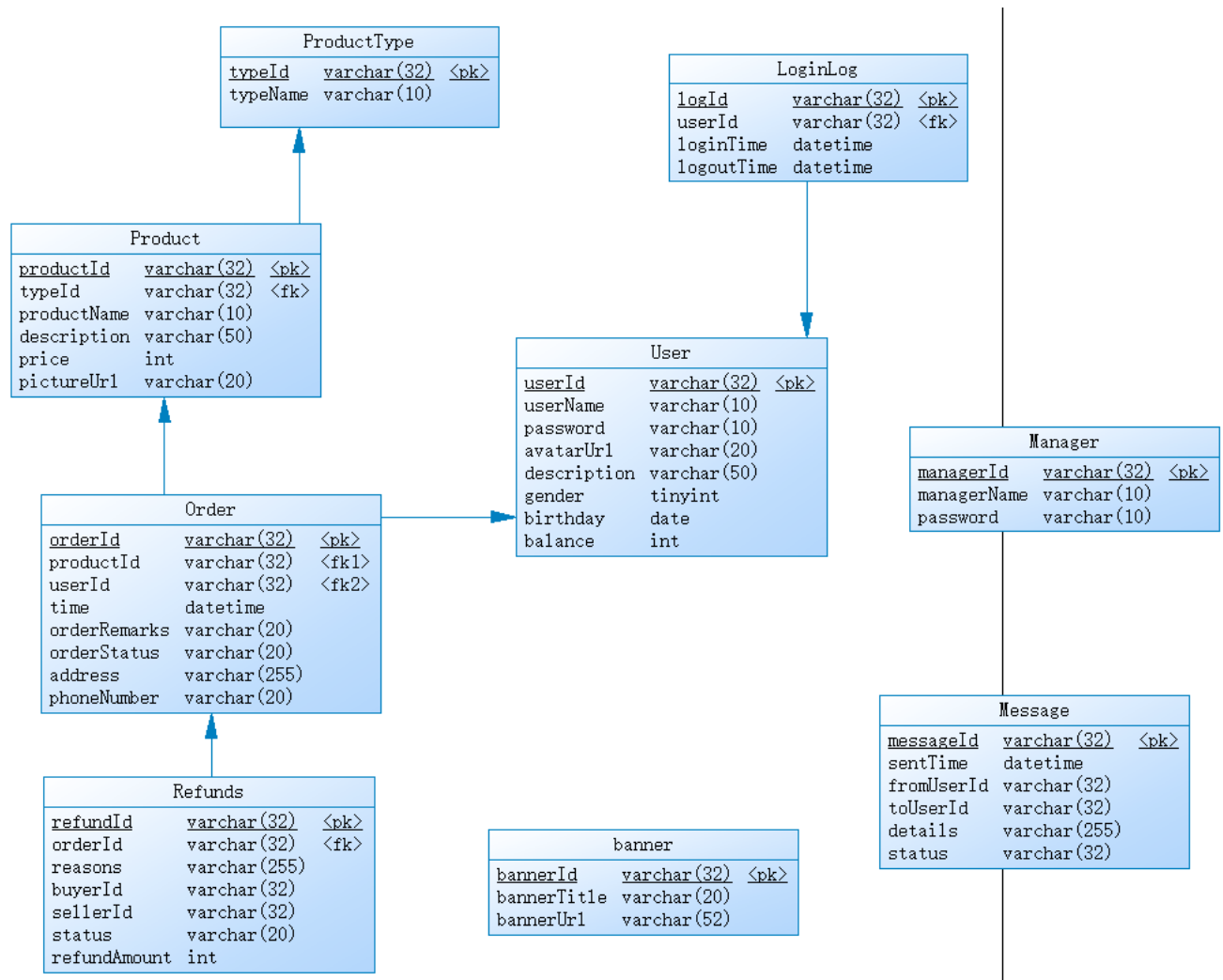
Figure 2.5 – Physical Diagram

Table 3.1 - Structure of "Product" table

Field name	Data type	Field size (bytes)	Description
productId	varchar	32	Product ID(PK)
typeId	varchar	32	Product Type ID(FK)
productName	varchar	10	Product Name
description	varchar	50	Product Description
price	int	4	Product Price
pictureUrl	varchar	20	Product Picture URL

The "Product" table stores information about product, shown in table 3.1.

Table 3.2 - Structure of "ProductType" table

Field name	Data type	Field size (bytes)	Description
typeId	varchar	32	Product Type ID(PK)
typeName	varchar	10	Product Type Name

The "ProductType" table stores information about product type, shown in table 3.2.

Table 3.3 - Structure of "Order" table

Field name	Data type	Field size (bytes)	Description
orderId	varchar	32	Order ID(PK)
productId	varchar	32	Product ID(FK)
userId	varchar	32	User ID(FK)
time	datetime		Order Create Time
orderRemarks	varchar	20	Orders Remarks
orderStatus	varchar	20	Order's Status
address	varchar	255	Receiving Address
phoneNumber	varchar	20	Receiving Phone Number

The "Order" table stores information about order, shown in table 3.3.

Table 3.4 - Structure of "Refunds" table

Field name	Data type	Field size (bytes)	Description
refundId	varchar	32	Refund ID(PK)
orderId	varchar	32	Order ID(FK)
reasons	varchar	255	Refund Reason
buyerId	varchar	32	Buyer ID
sellerId	varchar	32	Seller ID

Continuation of table 3.4

Field name	Data type	Field size (bytes)	Description
status	varchar	20	Refund Status
refundAmount	int	4	Refund Amount

The "Refunds" table stores information about daily refunds, shown in table 3.4.

Table 3.5 - Structure of "LoginLog" table

Field name	Data type	Field size (bytes)	Description
logId	varchar	32	Log ID(PK)
userId	varchar	32	User ID(FK)
loginTime	datetime		Login Time
logoutTime	datetime		Logout Time

The "LoginLog" table stores information about login logs, shown in table 3.5.

Table 3.6 - Structure of "User" table

Field name	Data type	Field size (bytes)	Description
userId	varchar	32	User ID(PK)
userName	varchar	10	User Name
password	varchar	10	User Password
avatarUrl	varchar	20	Avatar URL
description	varchar	50	User Description
gender	tinyint	1	User Gender
birthday	date		User Birthday
balance	int	4	User Balance

The "User" table stores information about User, shown in table 3.6.

Table 3.7 - Structure of "banner" table

Field name	Data type	Field size (bytes)	Description
bannerId	varchar	32	Banner ID(PK)
bannerTitle	varchar	20	Banner Title
bannerUrl	varchar	52	Banner URL

The "banner" table stores information about banner, shown in table 3.7.

Table 3.8 - Structure of "Manager" table

Field name	Data type	Field size (bytes)	Description
managerId	varchar	32	Manager ID(PK)
managerName	varchar	10	Manager Name
password	varchar	10	Manager Password

The "Manager" table stores information about system manager, shown in table 3.8.

Table 3.9 – Structure of "Message" table

Field name	Data type	Field size (bytes)	Description
messageId	varchar	32	Message ID(PK)
sentTime	datetime		Message Sent Time
fromUserId	varchar	32	Message Sender
toUserId	varchar	32	Message Receiver
details	varchar	255	Message Details
status	varchar	32	Message Status

The "Message" table stores information about message, shown in table 3.9.

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables

because it references the primary key of another table, thereby establishing a link between them.

The foreign key constraint is used to prevent actions that would destroy links between tables. A foreign key is a field (or collection of fields) in one table that refers to the primary key in another table. The table with the foreign key is called the child table, and the table with the primary key is called the reference d or parent table.

3. UML DIAGRAMS

UML is the abbreviation of Unified Modeling Language, which is a standardized modeling language composed of a set of diagrams. UML is used to help system developers clarify, display, build and record the output of software systems. UML represents a series of practices that have been proven successful in large and complex system modeling and is a very important part of the development of object-oriented software and software development. UML mainly uses graphical symbols to represent the design of software projects. Using UML can help project teams communicate, explore potential designs, and verify software architecture designs. Below we will introduce to you in detail what UML is, the history of UML and the description of each UML diagram type, supplemented by UML examples [2].

3.1 Use Case Diagram

Use case diagrams are diagrams that describe use cases, participants, and the relationships between them. The use case diagram is to describe the demand for the information system from the user's point of view and analyze the function and behavior of the product. The use case diagram defines and describes the externally visible behavior of the system, and is an important basis for analysis, design, and assembly testing. The use case diagram consists of the following concepts: Participants: roles, users of the system; System boundary: Determine the scope of the system. The boundary is a box, the use case is inside the boundary, and the participant is outside the boundary; use case: the service provided by the system; association: the relationship between the participant and the use case.

Use case diagrams are model diagrams of system functions that can be observed by external users called participants. It presents some participants and some use cases, as well as the relationships between them. They are mainly used for the functions of systems, subsystems or classes. The behavior is modeled, and the use case diagram shows how the use cases are related to each other and between the same use case participants. The main roles (User) of this system include User and manager(administrators).

Users have such functions: Search for products, include three methods: matching keywords、search record saving、type matching. Buy products, include four steps: consignee information、cancel the order、confirm receipt、apply for a refund. Modify posted product, include three steps: modify price、modify description、modify picture. Post used products: include four steps: set description、upload picture、set price、set product type. Edit personal information, include four steps: upload avatar、edit nickname、change password、what's up. Communication with seller. Communication with buyer. Administrators can also edit users' personal information, and in addition, administrators can discern disputes raised by users, maintain a library of product types, manage products posted by users, etc.
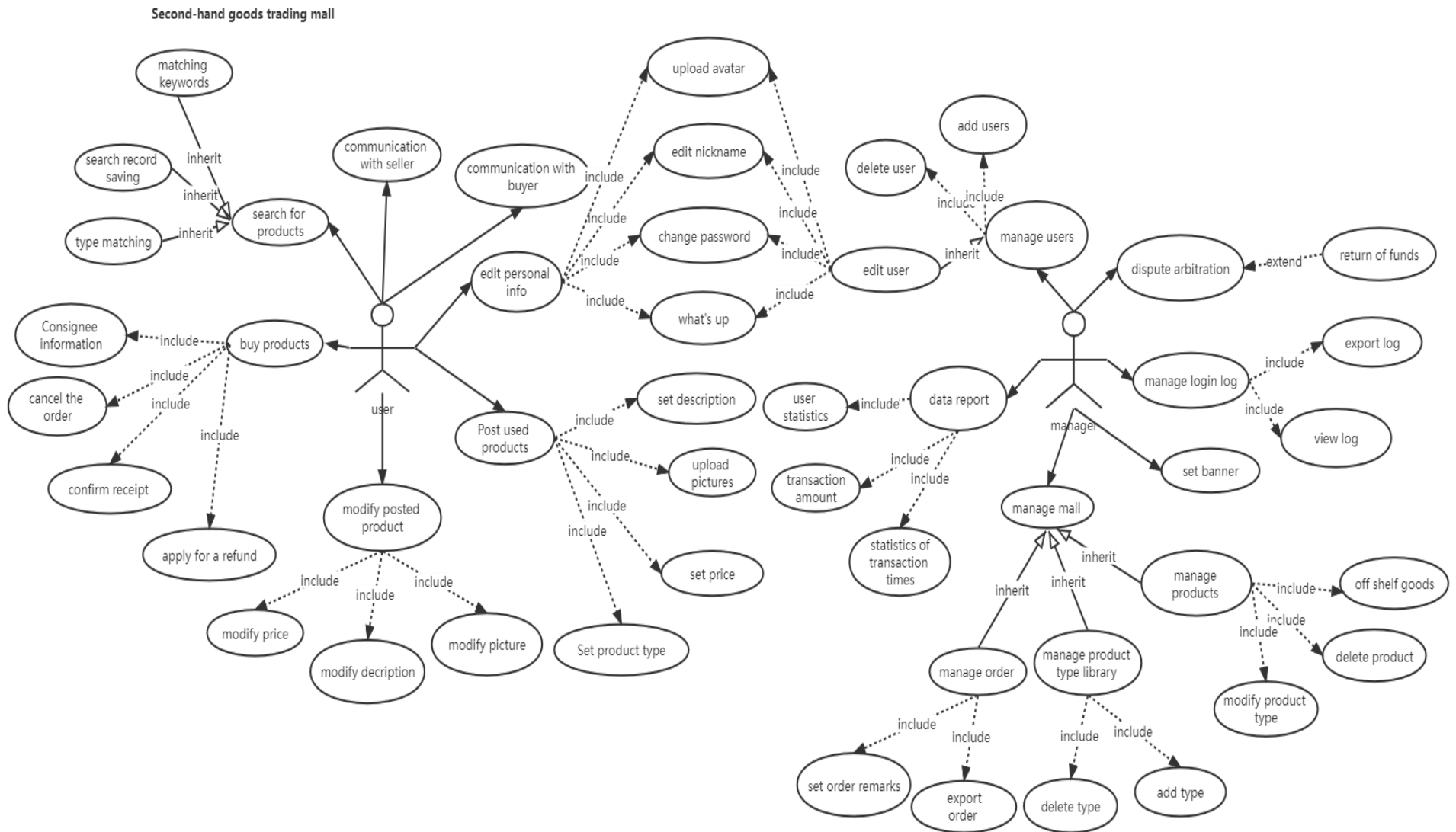
Figure 3.1 – Use Case Diagram

3.2 Sequence Diagram

UML sequence diagrams are interaction diagrams that detail how to perform operations. They capture the interaction between objects in the context of collaboration. Sequence diagrams are the focus of time. They visually display the sequence of interactions by using the vertical axis of the diagram to indicate the time and time of sending a message [3].

The sequence diagram captures the interactions that occur in the collaboration, which realizes the use cases or operations of high-level interactions between system users and systems, between systems and other systems, or between subsystems (sometimes called system sequence diagrams).

Purpose of Sequence Diagram:
- Model high-level interaction between active objects in a system.
- Model the interaction between object instances within a collaboration that realizes a use case.
- Model the interaction between objects within a collaboration that realizes an operation.
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of an interaction (showing just one path through the interaction).

The system can be started in three ways, i.e., user, administrator.

First, the user makes a login request, and then the server initiates an authentication process. The information entered by the user is checked against the account information in the database. After that, the server loads the product information from the database and displays the shopping page. The user places an order on the shopping platform while the platform reduces the stock quantity in the database and stores the order information. Visitors can enter the after-sales system to make after-sales requests for orders that have been placed.

Next, the administrator makes a login request, and then the server starts an authentication process. After that, the server reads the arbitration request sent from the user, and then the administrator processes the request and gives feedback on the result. In the shopping system the administrator can supervise the products posted by the users and also maintain the product type library and update the information into the database.
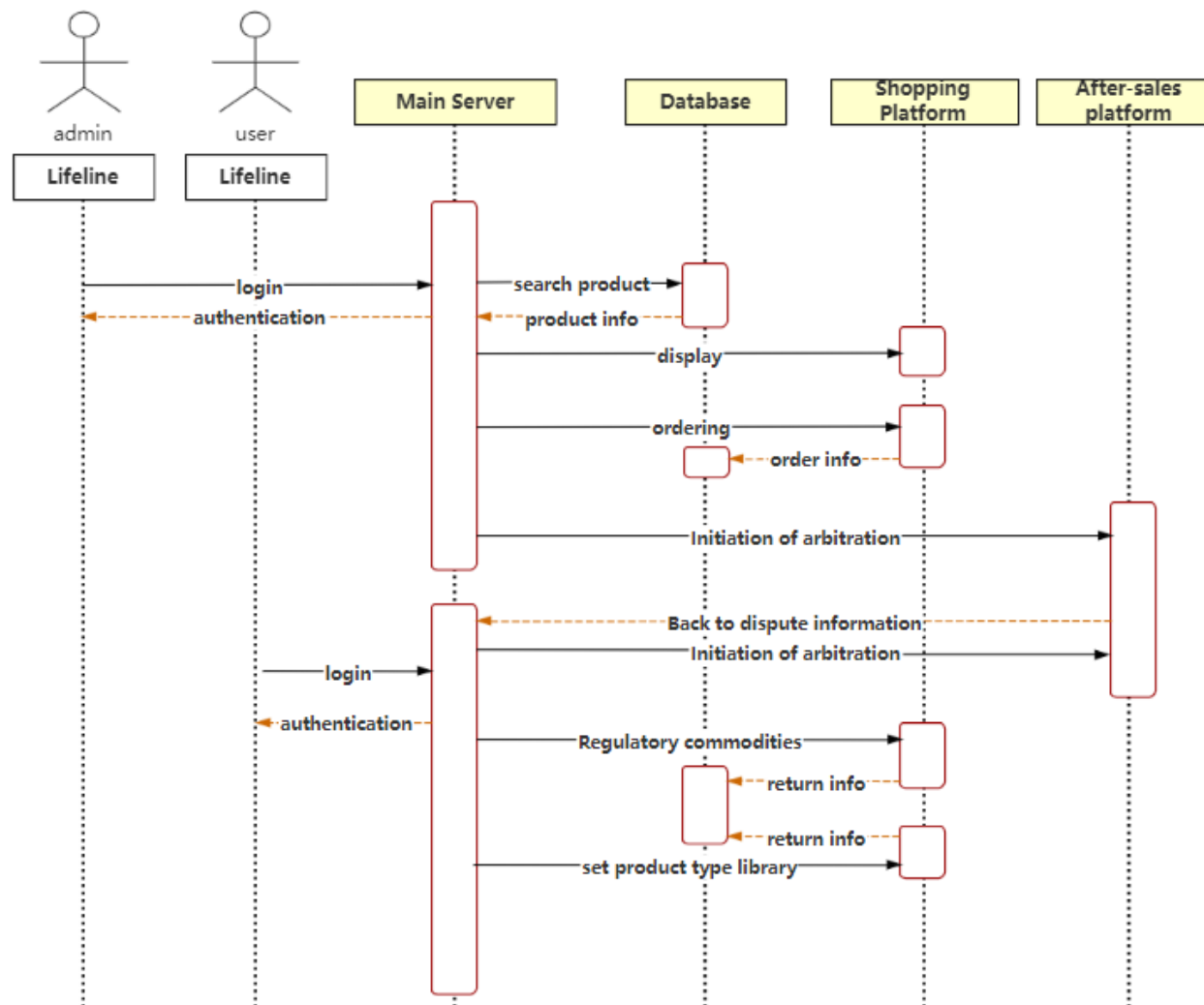
Figure 3.2 – Sequence Diagram

3.3 Activity Diagram

Activity diagram is another common tool used by UML to model the dynamic behavior of the system. It describes the sequence of activities and shows the flow of control from one activity to another. Activity diagram is essentially a flow chart. Activity diagram focuses on the flow of control from one activity to another, and is a process driven by internal processing [4].
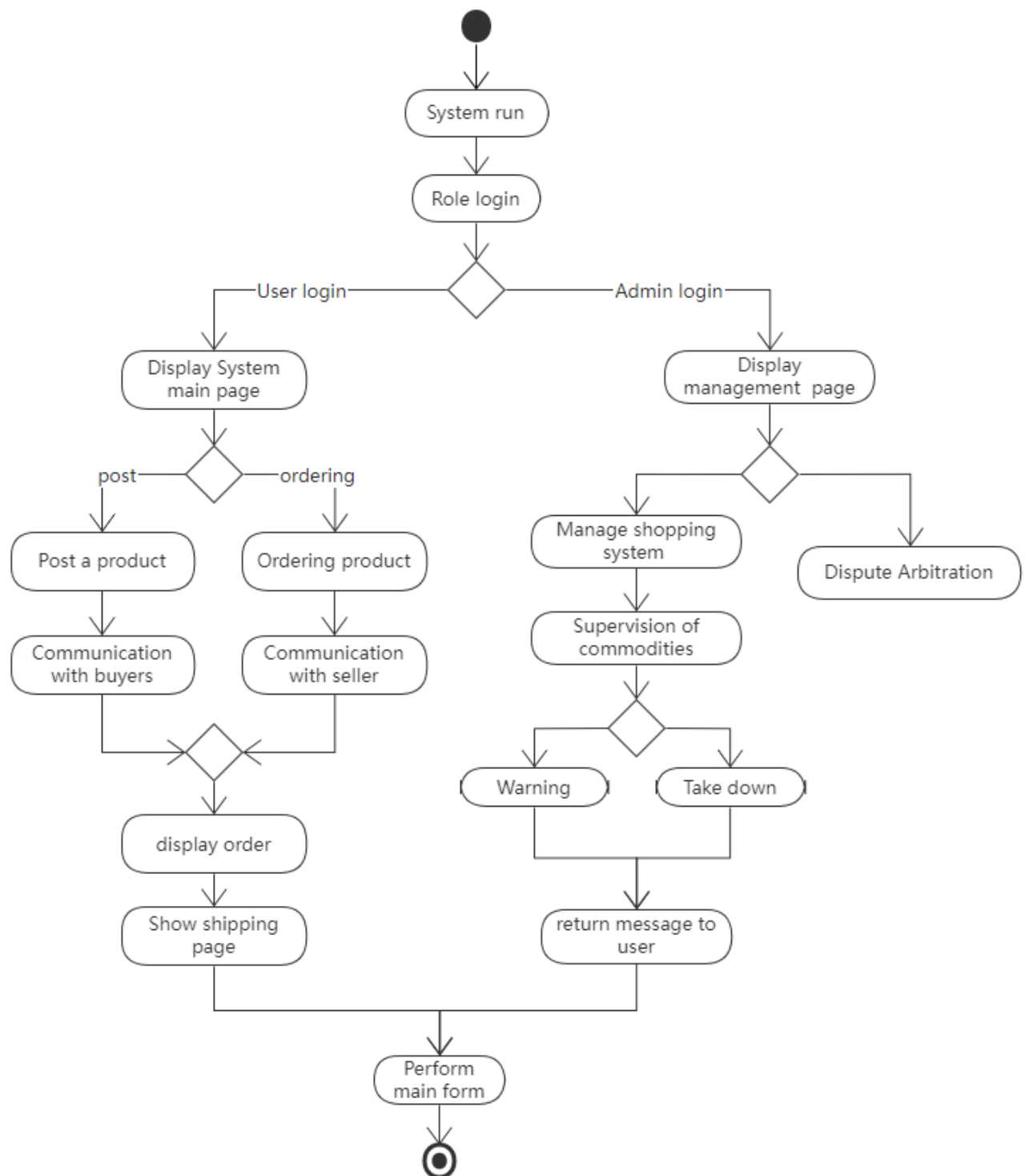

Figure 3.3 – Activity Diagram

The activity diagram describes the rules followed by the sequence relationship of object activities. It focuses on the behavior of the system rather than the processing process of the system. Activity diagrams can represent the situation of concurrent activities, and activity diagrams are object-oriented.

When a user uses a used goods trading system, he first logs in and then sees the shopping page homepage, on which he can select different types of goods and also choose to post his own goods, and in the process, he can communicate with the seller or buyer to discuss the specific information about the goods.

At the same time, the management of various user information, the setting of mall announcements, the handling of order disputes, and the management of products all need to be operated by the system administrator. The business flow chart of the trading mall is shown in the figure.

CONCLUSION

This paper mainly describes the design process of the second-hand goods trading system, including functional diagrams and UML diagrams, the system is used by users and administrators, users have both buyer and seller status, which largely enhances the convenience of second-hand goods trading, at the same time, the system is supervised and maintained by administrators, which can make the trading system more orderly. The system is monitored and maintained by the administrator, which makes the trading system more orderly.

In addition, the design of functional diagrams also caused some small mistakes, such as in IDEF0, User was wrongly used as the input of the system, but under the guidance of the tutor, the input was changed to commodity, because this is a second-hand goods trading mall, so commodity is the input source of the system.

Finally, I would like to thank my mentor Khajynova Natalia Vladimirovna for her careful guidance, so much so that the system of the second-hand goods trading was designed to perfection.

REFERENCES

[1] Automation of strategy using IDEF0A [Electronic resource] - Access mode: https://www.sciencedirect.com/science/article/pii/S2214716015000111

[2] How to make UML diagrams [Electronic resource] - Access mode: https://www.lucidchart.com/pages

[3] Understand UML class diagrams and sequence diagrams [Electronic resource] - Access mode: https://design-patterns.readthedocs.io/zh_CN/latest/read_uml.html

[4] Introduction to UML Modeling Activity Diagram [Electronic resource] - Access mode: https://www.cnblogs.com/ywqu/archive/2009/12/14/1624082.html