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

The Belarusian State University Informatics and Radio Electronics

**Faculty** of Information Technology and Control

**«APPROVED»**

**The Head of Dpt.**

(signature)

**«15» September 202**

**2 years.**

**STATEMENT**

Of the course-project given to student of the gr.№920611 Li Boyi

1. Course-project name THE SECOND-HAND GOODS TRADING SYSTEM

2. Completion date of the project 15.11.2022

3. Background to the project Group of system users: Administrator The method of interaction – MySQL, Visual Studio 2019, XAMPP

4. Contents of the explanatory notes (the list of issues to be developed)

INTRODUCTION

1. FUNCTIONAL DIAGRAMS

2. UML DIAGRAMS

CONCLUSION

REFERENCES

5. List of graphic material (with a precise indication of mandatory drawings and graphs)

6. The project consultant (specifying the sections of the project) Khajynova Natalia Vladimirovna

7. Statement issue date 15.12.2021

8. The schedule of the project design \_\_\_\_

01.10.2021 г. – 30%\_\_ the 1st checkup

15.10.2021 г. – 60%\_\_the 2nd checkup

01.12.2021 г. – 90%\_\_the 3d checkup

15.12.2021 г. – 100%\_\_project defense\_\_

PROJECT MANAGER / Khajynova N.V.

(signature)

Statement accepted for execution 15.12.2021 /

(date and signature of the student)

**CONTENTS**

[INTRODUCTION 5](#_Toc89434613)

[1. Functional Diagrams 8](#_Toc89434614)

[1.1 IDEF0 Diagram 8](#_Toc89434615)

[1.2 DFD Diagram 10](#_Toc89434616)

[1.3 Database Model 11](#_Toc89434617)

[2. UML Diagrams 16](#_Toc89434618)

[2.1 Use Case Diagram 16](#_Toc89434619)

[2.2 Sequence Diagram 18](#_Toc89434620)

[2.3 Activity Diagram 20](#_Toc89434621)

[2.4 Class Diagram 21](#_Toc89434622)

[CONCLUSION 24](#_Toc89434623)

[REFERENCES 25](#_Toc89434624)

# 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 Diagrams (DFDs) 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.

Class diagrams are one of the most useful types of diagrams in UML because they clearly depict the structure of a particular system by modeling the relationships between classes, properties, operations, and objects.

# 1. FUNCTIONAL DIAGRAMS

## 1.1 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.

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



Figure 1.1 - IDEF0 Diagram



Figure 1.2 - 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 ordering system, there are one input called Visitor. Manager and Staff are used the mechanism arrow. Order regulation and Daily stock in is on the control arrow space. The output is only the Order & Delivery.

Visitor is the input of Visitor Login, then comes to the Order Platform. Order Platform is under the control of Daily stock in and Order regulation, meanwhile Manager is the mechanism. Then goes to Cart Check, which is controlled by Daily stock in and Order regulation. It also uses the mechanism Manager. After that is Delivery Platform. It outputs Order & Delivery with the Manager mechanism.

## 1.2 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 1.3 - 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 the request of the visitor. After the visitor login process finishing, the information of visitor will be stored in the database. Then come to the visitor order process, which under the manner of order regulation. Some order information will be collected in the item cart database, at the same time process goes to the visitor check process. Stock and delivery information will be stored and modified in the database. At one time, management process can modify stock information. The process is going to the delivery. It will collect delivery information and send them to the order form database. Then it will display order process and the process can be terminated.

The process starts with the request of the manager. After the manager login process, it will into the management page. Manager manages daily stock and order regulation.

## 1.3 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).

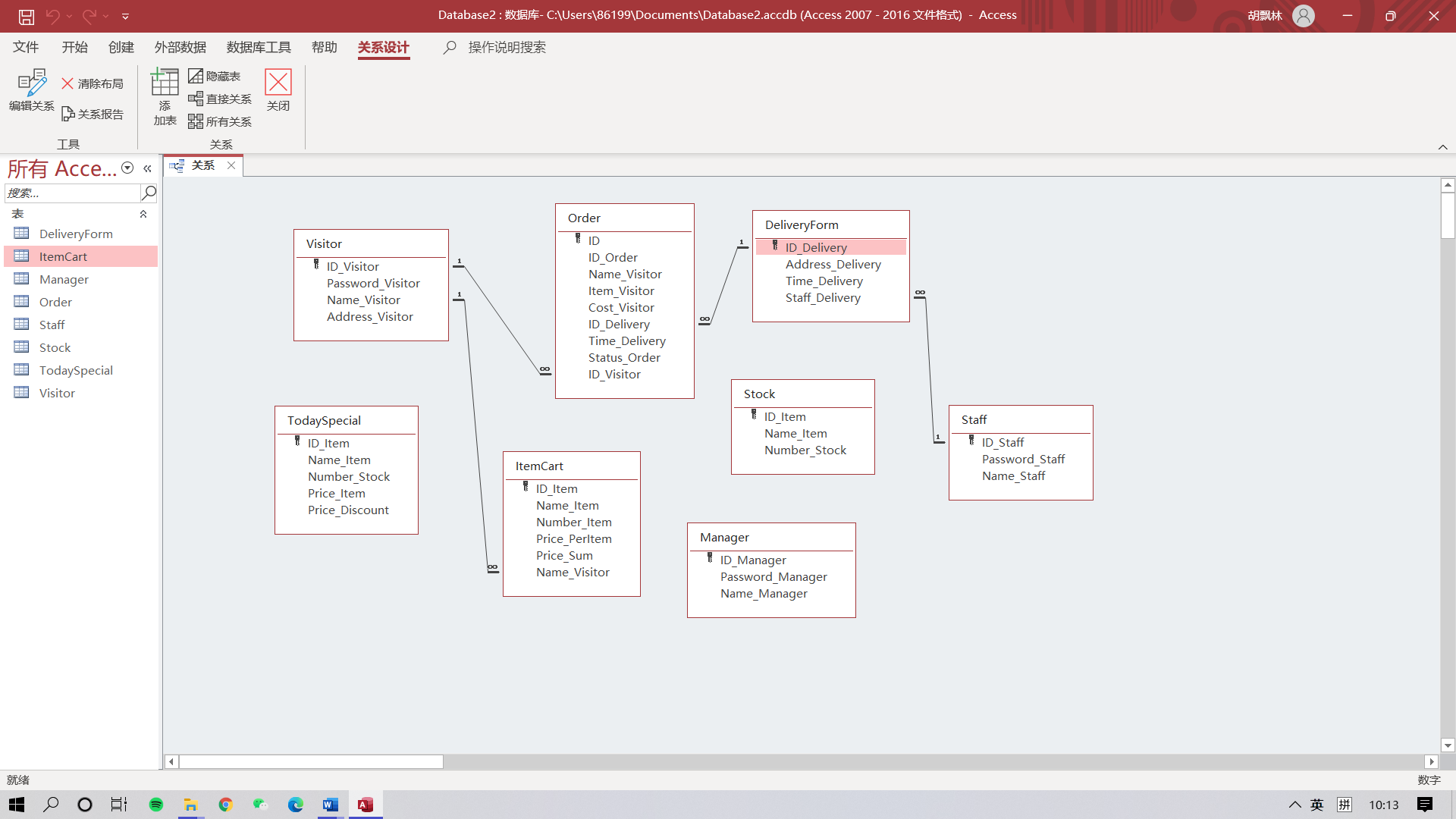


Figure 1.4 - Logical Diagram

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

Entity: There are 8 entities in the database, which are `Visitor`, `Order`, `TodaySpecial`, `ItemCart`, `DeliveryForm`, `Stock`, `Manager`, `Staff`. Detailed attributes can be seen in the figure.

Domains: 3 domains in the logic model.

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

Relationships and Foreign Keys: In total, there are 4 relationships between entities. In `Order`, the `ID\_Visitor` is a foreign key related with the primary key of entity `Visitor`, which means an order only has one visitor to make.

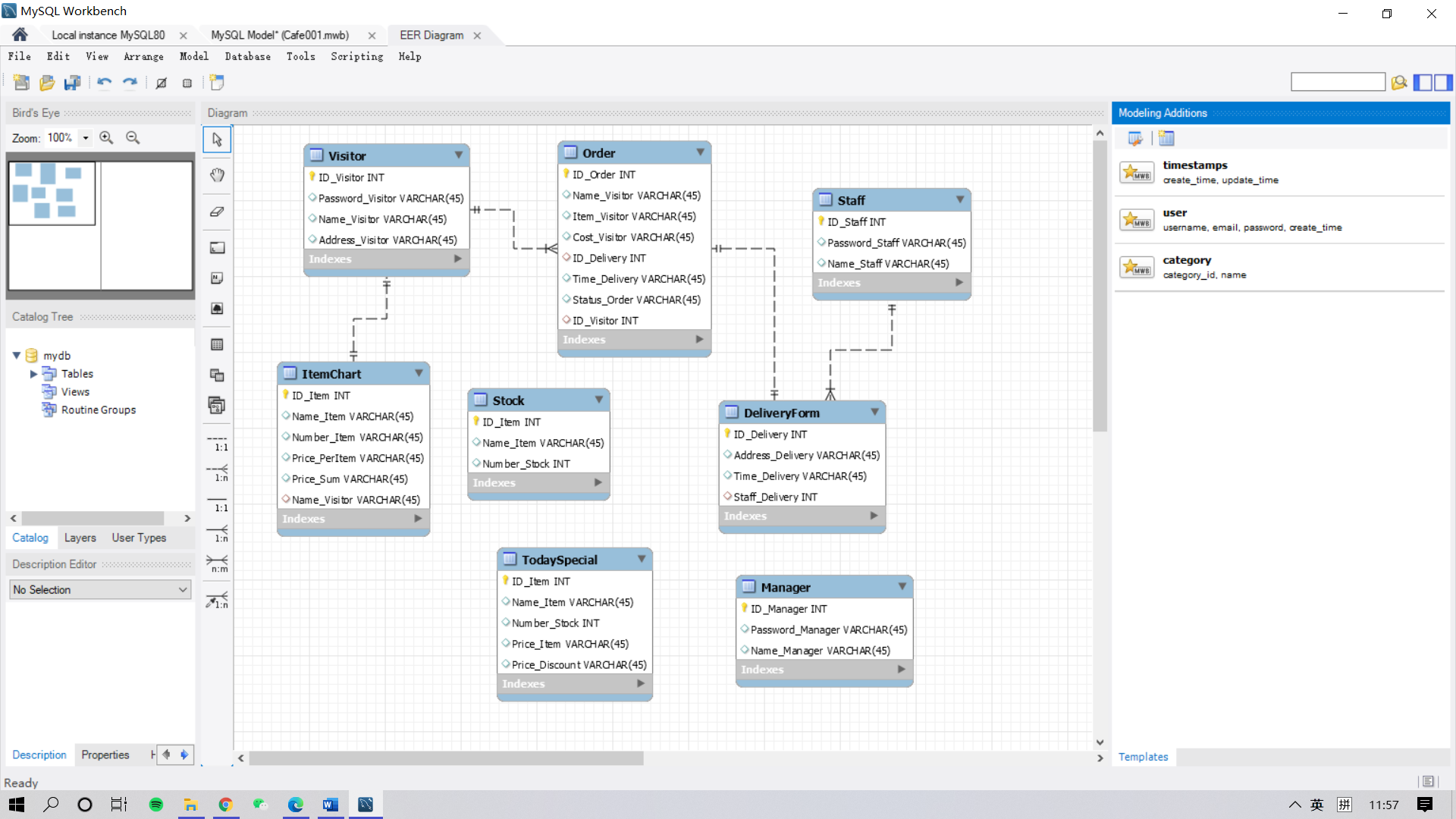


Figure 1.5 – Physical Diagram

Table 3.1 - Structure of “Visitor” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Visitor | int | 4 | Visitor ID |
| Password\_Visitor | varchar | 45 | Visitor Password |
| Name\_Visitor | varchar | 45 | Visitor Name |
| Address\_Visitor | varchar | 45 | Visitor Default Address |

The “Visitor” table stores information about visitor, shown in table 3.1.

Table 3.2 - Structure of “Manager” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Manager | int | 4 | Manager ID |
| Password\_Manager | varchar | 45 | Manager Password |
| Name\_Manager | varchar | 45 | Manager Name |

The “Manager” table stores information about manager, shown in table 3.2.

Table 3.3 - Structure of “Staff” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_ Staff | int | 4 | Staff ID |
| Password\_ Staff | varchar | 45 | Staff Password |
| Name\_ Staff | varchar | 45 | Staff Name |

The “Staff” table stores information about staff, shown in table 3.3.

Table 3.4 - Structure of “Stock” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Item | int | 4 | Item of Food ID |
| Name\_Item | varchar | 45 | Food Name |
| Number\_Stock | int | 4 | Stock Food Number |

The “Stock” table stores information about daily stock, shown in table 3.4.

Table 3.5 - Structure of “ItemCart” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Item | int | 4 | Item of Food ID |
| Name\_Item | varchar | 45 | Food Name |
| Number\_Item | int | 4 | Food Number |
| Price\_PerItem | varchar | 45 | Price Per Kind Food |
| Field name | Data type | Field size | Field name |
| Price\_Sum | varchar | 45 | Sum Price of Food |
| Name\_Visitor | varchar | 45 | Visitor Name |

The “ItemCart” table stores information about item cart, shown in table 3.5.

Table 3.6 - Structure of “DeliveryForm” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Delivery | int | 4 | Delivery ID |
| Staff\_Delivery | int | 4 | Staff ID |
| Address\_Delivery | varchar | 45 | Delivery Address |
| Time\_Delivery | varchar | 45 | Shipping Time |

The “DeliveryForm” table stores information about delivery form, shown in table 3.6.

Table 3.7 - Structure of “Order” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Order | int | 4 | Order ID |
| Name\_Visitor | varchar | 45 | Visitor Name |
| Item\_Visitor | varchar | 45 | Visitor Name |
| Cost\_Visitor | varchar | 45 | Visitor Cost |
| ID\_Delivery | int | 4 | Delivery ID |
| Time\_Delivery | varchar | 45 | Shipping Time |
| Status\_Order | varchar | 45 | Order status |
| ID\_Visitor | int | 4 | Visitor ID |

The “Order” table stores information about order, shown in table 3.7.

Table 3.8 - Structure of “TodaySpecial” table

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Field size  (bytes) | Description |
| ID\_Item | int | 4 | Item of Food ID |
| Name\_Item | varchar | 45 | Food Name |
| Number\_Stock | int | 4 | Stock Food Number |
| Price\_Item | varchar | 45 | Food Original Price |
| Price\_Discount | varchar | 45 | Food Discount Price |

The “TodaySpecial” table stores information about discount items, shown in table 3.8.

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.

# 2. UML DIAGRAM

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.

## 2.1 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 (Actors) of this system include visitor and manager.

Participants refer to objects outside the system, but directly interacting with the system, namely actors, also called executors and activists. Participants are represented by human-shaped symbols, and the participant's role name is marked under the human-shaped symbols. Participants are not only personnel, but also information systems and equipment. Use cases are functions that user expect the system to have. Each use case describes a service or function that a system provides to its users. The goal of a use case is to define a behavior of the system but does not show the internal structure of the system. The use case name is generally a verb-object phrase, and the symbol is an ellipse plus the use case name.

Visitors participate four main functions. Firstly, “Visitor Login” visitor can register or directly login with confirming the data from the database. Secondly, “Visitor Make Order” from this function visitor make order by browsing the order page. This function includes three cases, from Today’s Specials visitor get the information of the discount meal that is daily changed by manager depending on the daily stock. Visitor can choose meal to add it to the shopping cart also can regret order by clear cart function. Thirdly, “Visitor Check Order” only when visitor obey the order regulation established by manager can successfully confirm and make an order. Lastly, “Delivery” includes address confirm and delivery time. Visitors fill the address by the information stored in the visitor database when registered account. Visitor can use the new address when confirming address and they can get the delivery time set by the staff. Manager participates four functions. Today’s Specials, Daily Stock, Order regulation and View order. Manager inputs the stocks daily and make daily discount. The regulation of order is about only when meet the accurate sum amount of meal can make an order. Manager can view all the orders in the café and can delete them. Staff participate “View Order” also but have no the ability to delete the order. While they can set the delivery time and check the order in finish.

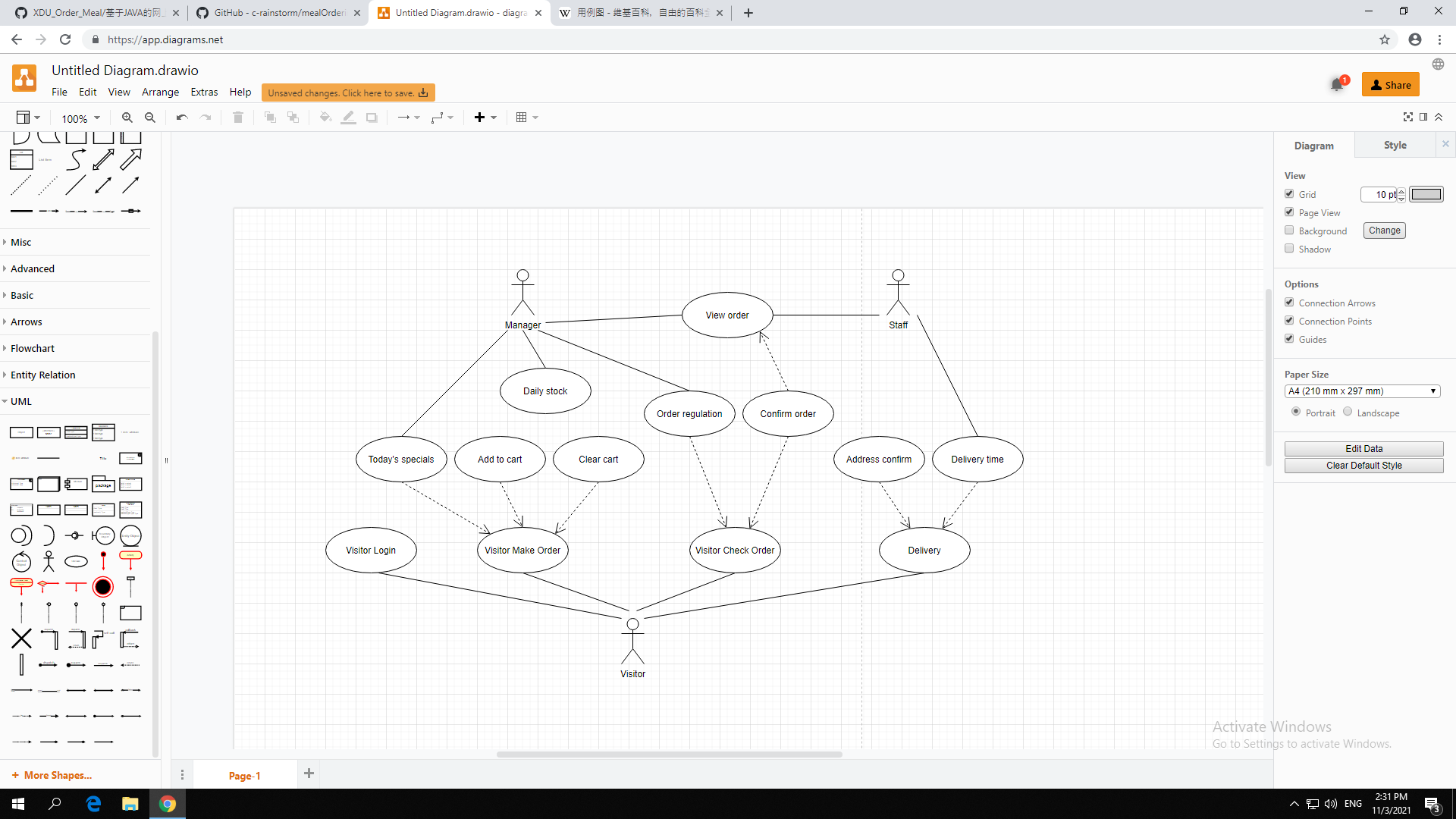


Figure 2.1 – Use Case Diagram

## 2.2 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.

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 start in three routes, which are visitor, manager and staff.

First, visitor make a request for login, then server launch an authentication process. The information visitor entered will be checked align the account information from the database. After that, server loads order process and display order page from the database. Visitor makes an order in the order platform, in the meantime platform reduce the stock quantity in the database. Visitor goes into the cart system, click the check button, make a request of checking, cart system will store the information of order in the database.

Second, manager make a request for login, then server launch an authentication process. Manager launches a process to enter the daily stock information to the database then database launch relevant modification. And manager launches a process to set or modify order regulation in the order platform then platform make a modification process and stored them to the database. Manager can make a query to get all the order information, after request database launch a process to display order information.

Third, staff make a request for login, then server launch an authentication process. Staff make a query to get all the order information, after request database launch a process to display order information.

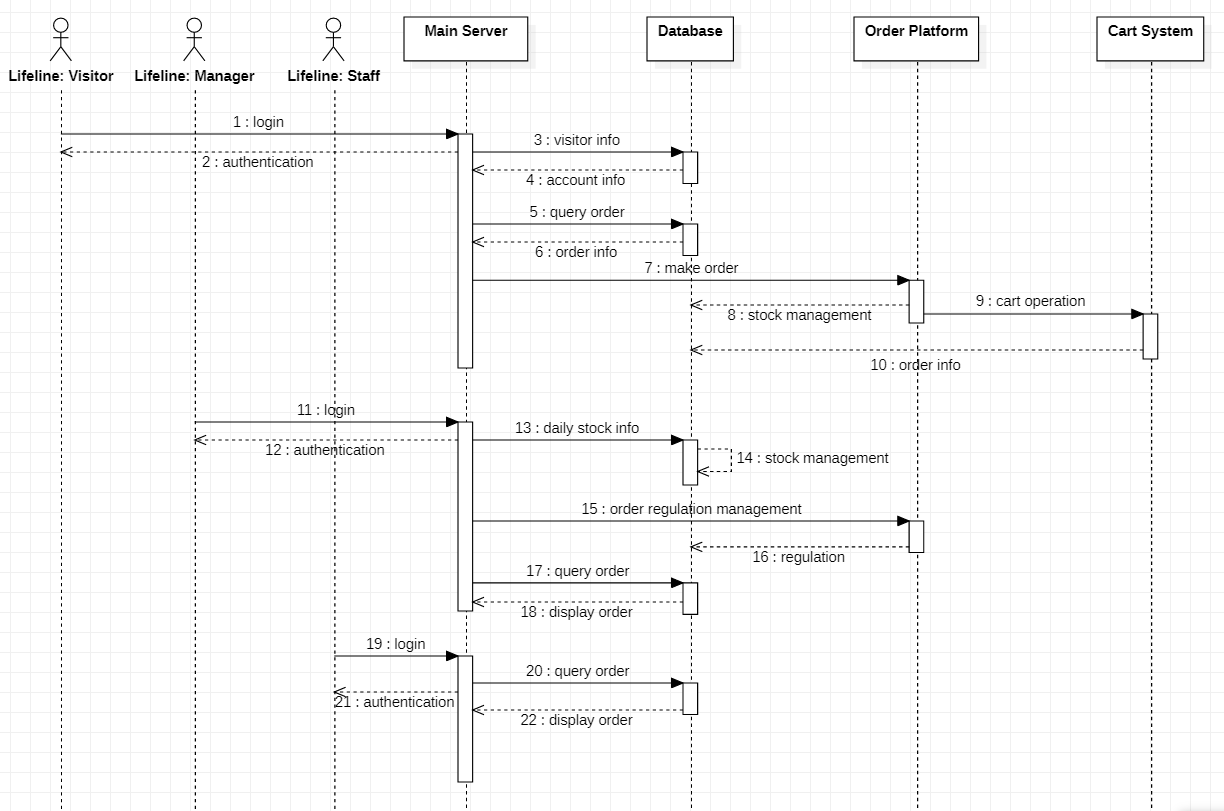


Figure 2.2 – Sequence Diagram

## 2.3 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.

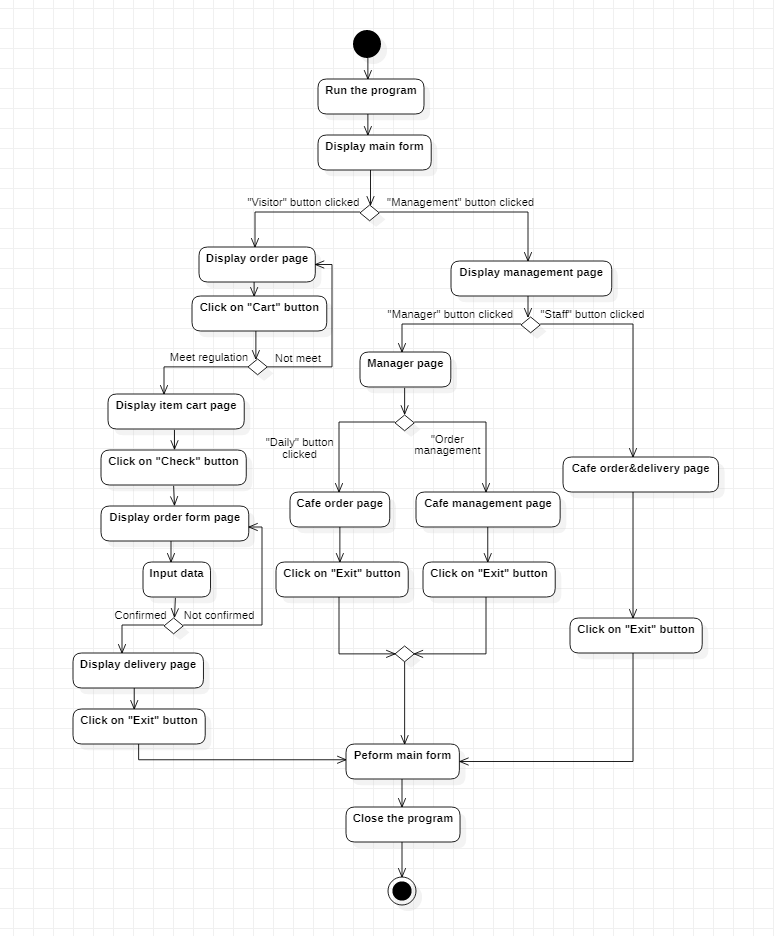


Figure 2.3 – Activity Diagram

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.

The first thing a user sees when browsing a website is the homepage of the website. On the homepage, users can learn about the restaurant and some special meals. If the user wants to order a meal and register as a member, after the user has viewed the product information, he can add the product he wants to buy into the order, or he can modify it at any time, and clear the goods in the order. If you are satisfied, you can place the order directly (you must fill in the information to place the order and meet a certain amount). At this time, the relevant program module will record all the information submitted by the user in the database so that the system administrator can check the information and prepare the meal. The special delivery personnel deliver the meal to the user and collect cash. At this time, the entire order is the process is over. In the back-end management of online meal ordering, it is also necessary to add, modify, and delete meals. At the same time, the management of various user information, the setting of mall announcements, the release of station news and the processing of orders, etc. need to be operated by system administrators. Online meal ordering the business flowchart is shown in the figure。

## 2.4 Class Diagram

Class diagrams are composed of many (static) descriptive model elements (such as classes, packages and their relationships, and these elements and their content are connected to each other). Class diagrams can be organized in (and belong to) packages, and only display relevant content in a particular package. Class diagram is the most commonly used UML diagram, showing classes, interfaces and the static structure and relationship between them; it is used to describe the structured design of the system. The most basic element of a class diagram is the class or interface.

Class diagrams are mainly used in the analysis and design stages of object-oriented software development to describe the static structure of the system. The class diagram illustrates all the entities of the constructed system, the internal structure of the entities, and the relationships between the entities. That is, the class diagram contains the classes abstracted from the user's objective world model, the internal structure of the class, and the relationship between the class and the class. It is the basis for constructing other design models. Without class diagrams, there will be no other UMI dynamic model diagrams such as object diagrams, state diagrams, and collaboration diagrams, and thus cannot express the dynamic behavior of the system. Class diagrams are also the starting point and basis for object-oriented programming.

Class diagrams are used to describe the classes contained in the system and their mutual relationships, to help people simplify the understanding of the system. It is an important product of the system analysis and design stage, and it is also an important model basis for system coding and testing.

图片包含 表格

描述已自动生成

Figure 2.4 – Class Diagram

Login class has four attributes: accountNumber, passwordLogin, loginID, con. Login class has two operations: Login, failLogin.

Register class has five attributes: registerAccount, registerPassword, registerAddress, registerID, con. Register class has three operations: succesRegister, failRegister, goLogin.

DisplayMenu class has eight attributes: itemID, itemName, itemPrice, itemStock, itemSource, itemDiscount, informationDiscount, con. DisplayMenu has four operations: displayMenu, addCart, goCart, diplaySum.

CheckOperation class has six attributes: orderID, costOrder, addressOrder, timeDelivery, phoneDelivery, con. CheckOperation class has six operations: confirmOrder, backCart, modifyAddress, modifyPhone, reduceStock, goDelivery.

QueryOperation class has six attributes: orderID, visitorID, nameVisitor, costOrder, itemOrder, con. QueryOperation has two operations: deleteOrder, exit.

DiscountOperation class has seven attributes: itemID, itemName, priceOriginal, priceDiscount, itemStock, itemSource, con. DiscountOperation class has six operations: modifyDiscount, increaseStock, reduceStock, deleteItem, addItem, exit.

CartOperation class has eight attributes: itemID, itemName, itemPrice, itemStock, itemSource, visitorSum, orderRegulation, con. CartOperation class has six operations: checkAll, deleteItem, addItem, clearAll, failCheck, backMenu.

StockOperation class has six attributes: itemID, itemName, itemPrice, itemStock, itemSource, con. SstockOperation class has five operations: increaseStock, reduceStock, deleteItem, addItem, exit.

OrderOperation class has three attributes: orderID, statusOrder, con. OrderOperation class has two operations: terminalOrder, exit.

# CONCLUSION

In this course project, the methods of designing automated systems for visitors of café are investigated. The method of graphical representation of an automated system using diagrams is shown. To describe the operation of the system, diagrams of the methodology IDEF0, DFD and UML diagrams have been created. Suggested options for using the system are suggested, which makes it easier for the code designer to work on creating the system.

For the “automated systems for visitors of café” four IDEF0, DFD and UML diagrams were developed. The documentation of the system describing the main, alternative, and erroneous flows is created. The concepts of class, attribute, operation, connection, essence, etc. were commanded.

The system provides a platform for the visitor to make an order of what they want. This system provides opportunity for administrator to manage and modify the system itself. This system is easy for any people to use, does not require much acquaintance of the use of PC.

This model allows you to generate the code of the programming language for which it is built. The classes and methods are contained that can apply to the system.

# REFERENCES

1. A Beginner's Guide to Data Flow Diagrams [Electronic resource] - Access mode: <https://blog.hubspot.com/marketing/data-flow-diagram>
2. What is Data Flow Diagram? [Electronic resource] - Access mode: <https://www.visual-paradigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/>
3. Automation of strategy using IDEF0A [Electronic resource] - Access mode: https://www.sciencedirect.com/science/article/pii/S2214716015000111
4. How to make UML diagrams [Electronic resource] - Access mode: <https://www.lucidchart.com/pages>
5. Understand UML class diagrams and sequence diagrams [Electronic resource] - Access mode: <https://design-patterns.readthedocs.io/zh_CN/latest/read_uml.html>
6. Introduction to UML Modeling Activity Diagram [Electronic resource] - Access mode: <https://www.cnblogs.com/ywqu/archive/2009/12/14/1624082.html>
7. Quickly learn sequence diagrams: introduction, drawing of sequence diagrams [Electronic resource] - Access mode: http://www.woshipm.com/ucd/607593.html