DEVELOPMENT OF CARGO DELIVERY MANAGEMENT SYSTEM FOR SMALL TRANSPORT COMPANY

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2018.308-578.GQW

Supervisor
Cand. Sci., Assoc. Prof.
_________ Zymbler M.L.

Author,
the student of the group CE-219
_________ K.D.H. Hamzah

Normative control
_________ Zymbler M.L
“___”___________ 2019

Chelyabinsk–2019
TASK
Of the master graduate qualification work
for the student of the group CE-219
KAMZAH KHALID DHAHIR

In master direction 02.04.02
“Fundamental Informatics and Information Technologies”
(master program “Database Technologies”)

1. **The topic** (approved by the order of the rector from 25.04.2019 No. 899)
Development of Cargo Delivery Management System for Small Transport Company

**The deadline for the completion of the work:** 05.06.2019.

2. **The source data for the work**

2.1. Official site of MySQL Server. [Electronic Resource] URL:


3. **The list of the development issues**

3.1. Study the problem statement.

3.2. Develop the database structure for the information desktop–application.

3.3. Develop graphic user interface forms for desktop–application.

3.4. Develop a use-case model of Boss and Dispatcher.
3.5. Develop graphical user interface forms for system.

3.6. Implement the desktop – application.

3.7. Create test set and perform testing of the System functions.

4. **Issuance date of the task:** 08.02.2019.

**Supervisor**

Cand. Sci., Assoc. Prof. M.L. Zymbler

**The task is taken to perform** K.D. Hamzah
TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................................................................................5

INTRODUCTION ..........................................................................................................................6

1. ANALYSIS OF THE SYSTEM ...............................................................................................15
   1.1. Semi-formal description of the subject domain .........................................................15
   1.2. Classes of the subject domain ....................................................................................16

CHAPTER 2. DESIGN OF THE SYSTEM ....................................................................................18
   2.1. Use case model ............................................................................................................18
       2.1.1. Use case diagram ...............................................................................................18
       2.1.2. Descriptions of actors .........................................................................................19
       2.1.3. Description of use cases ......................................................................................19
   2.2. Design of user interface .............................................................................................20

3. IMPLEMENTATION OF THE APPLICATION ....................................................................28
   3.1. Several fragments of C#-code for implementing the basic functionality .... 28
   3.2. Development of the database .....................................................................................31
   3.3. The used development tools .......................................................................................36

CHAPTER 4. TESTING OF THE SYSTEM .................................................................................38

CONCLUSION ..........................................................................................................................42

REFERENCE LIST ..................................................................................................................43
ACKNOWLEDGEMENTS

I would like to express thanks to my God to help me for do this system.

I want to thank my deepest thanks and gratitude to PhD, associate professor, Mikhail L. Zymbler, my supervisor for his guidance, support and his continuous enthusiasm and encouragement throughout the project.

I am also very grateful and extend my sincere thanks to all staff working in the Ministry of Higher Education and Scientific Research of Iraq for their cooperation.

Finally many thanks to friends, who have helped and given me suggestions, supports and corrections throughout the project.
INTRODUCTION

Almost all corporate body requires goods to be moved or distributed from one location to another, be it raw materials or manufactured goods. Also in day to day activities there arises the need to move assets from one location to another by air, land or sea. The destination for these goods ranges from states, countries and continents and may take hours, days, weeks or even months to reach. This observable fact brought about the need for owners to monitor the progress of their goods along the distribution network.

Tracking which means; “to follow the progress or development of something can stand as a definition for the need explained in the previous paragraph. For the course of this study we shall further define tracking as; “the process of gathering and presenting information on the location of delivery items in a distribution network or supply chain”.

Besides its role in providing information on shipment status, tracking can also have an impact on supply chain management as it provides the possibility of creating visibility or transparency to the material flowing in the supply chain. A comprehensive tracking system would enable a company to monitor the arrival critical components and plan its operation based on the estimated arrival of these components.

Topicality of Research

In order to transport goods or products from one territory to other, cargo system is important. It is fast, easy and reliable commercial means for transportation of goods and industrial products from one part of globe to another. This proposed project is simply an attempt to generate an application to automate different management activities in cargo business. The fundamental building blocks to cargo revenue maximization begin with sufficient, reliable and accessible data. Full air waybill capture is a good starting point. The waybill history should be stored in an accessible format in a database that can be updated daily.

The real challenge is not just to capture the complete life cycle of a shipment from original booking through to invoicing, but to do so in a dynamic and
timely way. With new developments in data handling and communications, it is now possible to build a data warehouse that receives seamless, real-time updates from reservation systems and with minimal changes to existing systems.

Cargo management system is the system through which you manage the shipping process with best desktop software. Cargo management system is the most complex and advanced management system as compared to others like School management system, laboratory or hospital management system.

Only because it has a lot of paper work which you have to manage a lot carefully.

In abroad to maintain the quality of cargo system they have graceful and well established digital software [14].

There are many reasons why I started my project this application desktop and not the Web and among these reasons, which is the Internet and the weakness of the Internet in Iraq and if I used a web had to be linked to the Internet and this is expensive and less protection because it is exposed to viruses.

**Goal and objectives of the research**

The goal of the research is to develop Development of Cargo Delivery Management System for Small Transport Company:

1. Study the problem statement;
2. Develop the database structure for the information desktop -application;
3. Develop graphic user interface forms for desktop –application
4. Develop a use-case model of Boss and Dispatcher
5. Develop graphical user interface forms for system
6. Implement the desktop -application;
7. Create test set and perform testing of the System functions.

The research has the following basic objectives:

1. We must create case model of the system.
2. We must design the user interface forms.
3. We must design the database structures.
4. We must designed and implemented System structure.
5. We must Test our system.

**Structure of the thesis**

The thesis consists of four chapters, introduction, conclusion and reference list.

In chapter 1, the problem statement is given in details;

In chapter 2, we make the comparative analysis of the existing analogical applications is given. We describe the used development took as well there.

There is a description of functional and non-functional requirements use case diagram, database scheme and the design of the application’s interfaces.

In chapter 3, we show several fragments of source code for implementing the basic functionality of the system.

Chapter 4 is devoted to the testing of the application. It contains the results of functional, and usability testing.

The thesis has 44 pages; the list of references contains 20 resources.

**Related work**

In paper [13] the management of a road transport system is multifaceted and multimodal. Practice in road transport management (RTM) draws upon knowledge from strategic planning, network operation, road safety, asset management and land use planning along with transport and vehicle technology. With the interconnectivity of the various disciplines as well as the emergence of Intelligent Transport Systems (ITS) and connected-autonomous vehicles, the RTM practice is complex and evolving rapidly.

The primary aim of this project is to review local and international practice in RTM in order to develop an agreed RTM framework, underpinned by principles and based on contemporary leading practice. RTM in Australia and New Zealand is predominantly the responsibility of state road agencies and local government (and private toll operators) who are vested with managing the public road network. Without an agreed and established RTM framework, the Austroads member organizations are facing inconsistency in planning, assessing and managing road transport infrastructure, systems and initiatives.
In paper [7] Road Transportation was viewed by the researcher as a sine qua non in the economy whether developed or developing. Road transportation is seen as the engine of an economy. It links producers, suppliers, consumers and commuters. But poor, ineffective and inefficient road transportation system have crippled development and the collapse of many economic and social activities which have resulted in dissatisfaction among the customers in the industry. This is characterized by poor management that has resulted in the loss of life and property including damages worth unquantifiable amount of money. The researcher used primary and secondary sources and methods of data collection in the study. The researcher presented the data in tables and used simple percentages to analyze the data obtained. The findings of the research among others were that road transportation is very essential for the movement of goods and people. Inefficient roads all over the country make commuters uncomfortable. Lack of maintenance culture by drivers always result in breaking down of vehicles and poor services rendered to customers' causes dissatisfaction. The researcher recommended that good roads must be put in place to address the situation. Regular maintenance of vehicles and roads and provision of quality customer services that will satisfy the commuters who constantly use the roads and vehicles for their economic activities.

In paper [1] the problem of the last kilometer deliveries (last mile) is important to improving the quality of life in the city. To solve this problem, it is necessary to look for an alternative for traditional means of transport used in the cargo distribution systems. The usage of freight bikes can be effective, but it needs a properly built system. The approach to modeling of distribution systems shown in this paper takes into account the stochastic nature of the demand for freight transport. Numerical parameters described in the paper, allow modeling distribution systems based on cargo bikes and other means of transport. Appropriate selection and modification of the proposed model allows to study and optimize existing processes of distribution. In addition, the presented approach allows researchers to evaluate the choice of means of transport relative to the load
being delivered. It causes that the tool can be practically used and enables to improve competitiveness of transport companies on the market. As the directions of further research, the development of the advanced procedures of demand simulations and the implementation of algorithms of such technological operations as vehicles scheduling and stacking of load units in a bicycle loading space should be mentioned.

In paper [11] authors presented Sensing Human Activity: GPS Tracking, the availability of so-called geopositioning devices such as GPS (Global Positioning System) devices has grown enormously in the last decade and is still increasing. More and more people own a navigation system such as a TomTom, a GPS for orientation for outdoor uses, biking and geo-caching or a mobile phone or other handheld communication device with built-in GPS. These devices are mainly used for orientation (determining where you are), navigation (determining where to go) and communication (exchanging information with others or accessing information services). But the devices can also be used for tracking, i.e. saving a travelled route into a track log. This ability makes the technology useful to collect spatial-temporal data and thus as ‘sensors’ for observing and measuring activities of people. GPS is a Global Navigation Satellite System (GNSS). GNSS is a system for location or position determination – so called geopositioning using a special receiver, a geoposition in space and time can be calculated based on the reception of satellite signals. The United States’ Global Positioning System (GPS) was the first available system using satellite Position Determination Technology (PDT) Other GNSSs are under development in Europe (Galileo) and Russia (Glonass). GNSS is an essential Positioning Determination Technology for many fields of study. Although in recent years the system has undergone a significant ‘modernisation’ to improve its quality, the capability for repositioning in the built-up (urban) environment is still one of its major weaknesses. In particular the availability of accurate indoor signals and with low speeds is limited. The future availability of Galileo is expected to increase the performance of GPS significantly.
In paper [11] authors presented Vehicle Tracking System Using GPS, in the present growing economy, the country also faces the uprising of crime rate. Car theft, which is the main concern for the conduct of this project, is one of the biggest crimes which is hard to eliminate. The latest trend of car theft involves the car being towed away, and also alarm signal capturing where the alarm disabler signal can be traced and duplicate by a thief with the device to capture the signal and use it to disable the alarm. There are many alternatives to prevent the car theft, common car alarm system which nearly all cars have the system installed, and also Global Positioning System (GPS) where the whereabouts of the car can be traced. The project and research is conducted for additional features in car alarm system. The device can be added to the present car alarm system without any major modification to it. Vehicle security system using SMS/GPRS is an advanced feature of the existing car security system. The project describe a practical model for routing and tracking with mobile vehicles in a large area outdoor environment based on the Global Positioning system (GPS) and Global system for mobile communication (GSM). The supporting device GSM modem GM862, are controlled by a 32 bit microcontroller LPC2148 implemented a new version ARM cortex M3 core. The device will collect position to supervised center by the SMS (Short Message Service) or GPRS (General Package radio service). A hardware device mounted on the vehicle is connected to the engine. Once, the vehicle is being stolen, the information is being used by the vehicle owner for further processing. Sitting at a remote place, a particular number is dialed by owner to the hardware kit which is installed in the vehicle. By reading the signals received by the mobile, one can control the ignition of the engine; say to lock it or to stop the engine immediately. We can modify this concept such that the vehicle owner also can lock the vehicle from his mobile phone.

In paper [17] authors presented design and development of GPS-GSM based tracking system with Google map based monitoring, In this urban life transportation is very common. A lot of mishappenings occur on the road every day. Therefore the need of security and monitoring is developed. To resolve
such problems, a system is developed using GPS and GSM technologies and an application is introduced in this research work. Various problems that we face:

1. In critical condition (when vehicle is stolen), one is confused what to do.

2. If one has something expensive and he wants to check it regularly.

3. To find the shortest path available.

All these problems are overcome by the system. This system has Global Positioning System (GPS) which will receive the coordinates from the satellites among other critical information. Tracking system is very important in modern world. This can be useful in soldier monitoring, tracking of the theft vehicle and various other applications. The system is microcontroller based that consists of a global positioning system (GPS) and global system for mobile communication (GSM). This project uses only one GPS device and a two way communication process is achieved using a GSM modem. GSM modem, provided with a SIM card uses the same communication process as we are using in regular phone. The system is not limited to find the location of the target but also calculates the distance travelled b/w two stations. This system is user friendly, easily installable, easily accessible and can be used for various other purposes. After installation system will locate target by the use of a Web application (HTML based application) in Google map. The system allows to track the target anytime and anywhere in any weather conditions.

In paper [18] The Brazilian cargo delivery system is quite different from most jurisdictions elsewhere in the world where the carrier delivers the cargo in exchange for the original bill of lading or a delivery note. In Brazil, the carrier discharges import cargo into Customs-bonded facilities that take over the duty to safe keep the cargo and hand it over to the consignee on completion of the clearance formalities and upon authority from the Customs. Apart from fines for incorrect or untimely declared cargo manifest, there have been recent regulatory changes in that it is possible, under certain circumstances, for importers to take delivery without the need to surrender the original bill of lading. These changes
in contrast with international delivery practices and, indeed, the standing legal system, have caused great concern amongst carriers who completely lose control over the cargo once it is discharged and have little or no say on whether and to whom it can be delivered. The peculiarities of the Brazilian cargo delivery system in contrast with other countries bring about numerous queries from carriers trading in the country on the responsibility for cargo reporting procedures, delivery and determination of liabilities for cargo loss or damage. Based on our hands-on experience, we prepared this guidance with a practical introduction to the Brazilian Customs regulations and practice pertaining to cargo import, clearance and delivery, the statutes governing cargo manifest reporting requirements and the apportionment of liability between carriers and bailees. The guide also covers the issue of Customs penalties and the defences and limitations available to the carriers. While this publication is not legal advice nor intends to be any comprehensive, we hope it is useful as a source of practical reference to our clients and associates.

In paper [3] urban freight transport is vital in sustaining urban economies and life, since it carries the required goods and the generated waste in cities. Meanwhile, it also has several negative external impacts, which are considerably affecting the lives and well-being of local populations. As urbanization continues throughout the world, these negative effects will be further enhanced because of growing transportation needs. The city of Gothenburg is also expanding, which prompted stakeholders, who are directly and indirectly involved in the freight transport system, to explore more sustainable urban freight transport solutions. One of these possible solutions is urban waterway transportation, as the city provides suitable infrastructure to accommodate both goods and waste transportation over the river. However, for a successful implementation of a transport service it’s not enough to prove the technical feasibility, the long-term economic feasibility also has to be demonstrated to convince the financially involved actors, which means the solution needs a viable business model. Therefore, the purpose of this thesis is to analyse the business aspects of operating an
urban waterway transport service of goods and waste in Gothenburg, and to examine which are the critical parts of the business model that influence the commercial feasibility of this service. In order to fulfil this purpose, a scenario analysis was conducted with observations and in-depth interviews with several actors who are knowledgeable about the Gothenburg freight transport situation, as well as have specialised experience concerning waterway transport. The findings suggest that the critical parts are the value proposition and externalities, key partners and cost structure of the business model. As a result, the thesis contributes to the research field of waterway transport business aspects and aid the city of Gothenburg in its search for possible sustainable freight transport solutions.

**Brief description of the thesis**

This thesis contents could be briefly described as follows. 0 is devoted to analysis design of the system and contains semi-formal description of the subject domain and description of the classes extracted during analysis. Error! Reference source not found. contains system’s use case model and design of user interface forms. In Error! Reference source not found. we place a description of the system’s implementation, information about the structure of database and system that we have designed, and deployment of the system. Error! Reference source not found. describes how we tested the system. Conclusion summarizes basic results of the thesis and shows directions of further research in relevant subject domain.
1. ANALYSIS OF THE SYSTEM

This chapter is devoted to analysis of the system. Here in section 1.1 we will give a semi-formal description of the subject domain.

In our description we will use Unified Modeling Language or the UML. UML is a graphical modeling language that provides us with syntax for describing the major elements of software system.

1.1. Semi-formal description of the subject domain

Road transport occupies the top position in the cargo and cargo system, but this arrangement has not been accompanied by it until it becomes the best sector, despite its importance.

Transportation of goods from one place to another easily and quickly, leading to the spread of goods in different parts of the country and this affects the movement of trade and economy in the country and land transportation is the basis of transport in Iraq and the coming period will see an increase in demand for this type of transport with an increase in the volume of internal trade And increase bilateral agreements between Iraq and other countries, which will lead to an increase in the volume of exports and imports.

Siddiqui's transport company is a company that connects and transports goods between different types of goods.

In the beginning, we will explain the mechanism of the company, which includes explaining the details of the company's operations from the manager, employees, drivers, vehicles, goods and delivery.

First, when the customer orders an order, he must go to the employee and enter all his information in the customer register. After that, the employee will enter all the order information in the second register, which is the order record. The application form will be filled out and the form will be sent to the manager by hand. The order and the type of truck that will be driven by the driver will be delivered. The name of the driver and the truck must be present in advance with
the driver on the list of available drivers, after which the form will be sent to the 
employee by hand.

After the goods are processed, they are delivered to the delivery site. The 
employee will then complete the order. The employee must document and ar-
chive the completed orders and available and unavailable drivers as well as 
available and unavailable trucks.

All of these above create problems in the functioning of the transport 
company. The problems faced by the shipping company's official are the diffi-
culty of handling, keeping and keeping paper documents properly. This process 
requires experience and skill, as well as the difficulty of extracting specific in-
formation from paper documents. The time may be lost and the documents may 
be damaged or lost.

1.2. Classes of the subject domain

Classes of the subject domain are shown in the Fig. 1.
Here we have the following list of classes: Driver, Order, Customers, and Trucks.

In Driver class we have 5 attributes name, gender, birthday, date, and phone.

In Order class we have 8 attributes customer, goods number, goods name, goods city, goods location, delivery city, delivery location and order date.

In customer class we have 3 attributes name, phone and address.

In customer class we have 4 attributes type, model, number and load.

Drivers class denotes keeping all driver data and also connected with orders class and used by it. Orders class denotes keeping all the data in Trucks class and also connected with Customers class and used by it. Orders class denotes usage Trucks class. Customer's class denotes common storage for three classes (Drivers, Trucks, and Orders).
CHAPTER 2. DESIGN OF THE SYSTEM

This chapter is devoted to design of the system. There is a description of use case model in Error! Reference source not found. Section. In section Error! Reference source not found. we will show a design of user interface.

2.1. Use case model

A use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. Unified Modeling Language (UML) enables IT professionals to model computer applications [16].

2.1.1. Use case diagram

Use case diagram are shown in the Error! Reference source not found.. In section Error! Reference source not found. we will talk about actors of the system. In section Error! Reference source not found. we will describe use cases. This diagram shows ten use cases. There are two types of user: BOSS and Dispatcher represented as actors. These actors are connected with these use cases by relationships to show the capability for each user in the system.

![Use Case Diagram](image-url)
2.1.2. Descriptions of actors

The first actor is "Boss" who is responsible for organizing and managing cargo flights and making appropriate decisions from assigning drivers cargo shipments with daily reports and feedbacks to drivers for making the appropriate decision.

The second actor is "Dispatcher" who is responsible for the introduction of cargo shipping information, the organization of its writing, the management of orders for customers (cargo delivery requests), the entry of goods information from its inception until arrival at the place of destination, the introduction of driver information.

2.1.3. Description of use cases

In the normal way, when a shipment is ordered to be written in the records manually, then the field is going to examine the drivers concerned and the trucks in charge of delivery, and the names of the customers can be lost.

It is difficult to know the manager about the reports and decision making. Because of the difficulty of establishing, the usual papers may lead to the loss of tracking the process through the damage of that paper, the system solve some of the problems of the field, which provides the Dispatcher easy to assign and manage the delivery shipment, Dispatcher are manages the operations of orders, enter information, delete and modify them. Dispatchers are Performs information management operations from its address, status, completeness, and driver information, information, deletion and modification.

Dispatchers are the information management operations full management to all Driver from the name, telephone number, status of the driver if available, evaluate each request to enter information, delete and modify them.

Dispatchers are full management to all Truck information management is done by its name, its status, if it is available to enter information, delete and
modify it. Dispatcher Performs customer information management by name, address, telephone number, information entry, deletion and modification.

2.2. Design of user interface

User Interface is the point at which a user or a user department or organization interacts with a computer system. The goal of user interface design is to make the user's interaction as simple and efficient as possible. [15].

Fig. 3 depicts flowchart of user interface of our system (here arrows denote possible transitions between dialogue forms).

Fig. 3. User interface flowchart

Our program consists of many interfaces each interface has a specific job. Firstly, main interface form shown in Fig. 4 shows the login form.
Second, Boss Windows interface consists of three main tabs. The first tab, User interface, is responsible to Assign Orders Displays in this interface the administrator all the orders to be displayed on it and display whether this order has been assigned to the driver or not and then assigns it to the driver as shown in fig. 5.

The second tab properties interface is responsible to Unassigned Orders In this interface, requests that are not assigned or performed are displayed on the manager, displaying their information and driver information as shown in fig. 6.
Third, Reports interface In this interface, reports are presented to the manager for appropriate decision making, providing the manager with the time and effort to track shipments and track drivers, and submit the following reports includes (Completed Order, Uncompleted Order, Available Drivers, Unavailable Drivers, Available Trucks, Unavailable Trucks)) interface as shown in fig. 7.

In Reports Interface have collection of reports Completed Order report all completed Order this report shows the manager the applications that have already been completed and displays the request and the driver who delivered or completed as shown in fig 8.
Uncompleted Order report all uncompleted Order. This report shows the manager requests that have not been completed and the reason for this and presents the driver assigned to transport this shipment to be a clear picture of the director helps in making the decision as shown in Fig 9.

Available Drivers report all available Drivers. In this report, the driver is offered the available drivers to make a decision to assign them to new tasks as shown in Fig 10.
Available Drivers report all unavailable Drivers. In this report, the driver is offered the drivers who are not currently available to make an appropriate decision as shown in Fig. 11.

Unavailable Trucks report all available trucks. In this report, the Manager is offered the available trucks as shown in Fig. 12.
Fig. 12. Unavailable Trucks report

In dispatcher window interface menu of operations.

First, Make Order to make Order of Customer (add new) in this interface, the dispatcher enters a new order, entering the name, number and address of the applicant as shown in fig. 13.

Fig. 13. Make Order form

In dispatcher window interface menu of operations.

Second, Crud Order to Edit and delete Order of Customer and Add Driver Rating. In this interface, the dispatcher displays all the requests and can enter new applications. After entering the name of the applicant, enter the address that
the shipment is sent to this place as well as the amendment and enter and activate the cases if the application has been completed or not completed and enters an evaluation of the driver who carried out the process as we shown in fig. 14.

Fig. 14. Crud Order form

In dispatcher window interface menu of operations.

Third, Crud Driver To Edit and delete In this interface, the dispatcher will enter the driver's name, driver's phone number, birth date and gender, and the driver's status will indicate whether it is currently available or not as we shown in Fig. 15.

Fig. 15. Crud Driver form

In dispatcher window interface menu of operations.
Forth, Crud Truck To Edit and delete In this interface, the dispatcher will enter all the trucks of the company's version, type and name and enter the case in if the truck is currently available or not as shown in fig. 16.

Fig. 16. Crud Truck form

In dispatcher window interface menu of operations.

Fifth, Crud Customer To Edit and delete In this interface, customers are managed where they receive information from a name, telephone number, address and can be modified or deleted as shown in fig. 17.

Fig. 17. Crud Customer form
3. IMPLEMENTATION OF THE APPLICATION

This chapter contains description of the system’s implementation. In section Error! Reference source not found. we view several fragments of C#-code for implementing that includes the main functions. In section Error! Reference source not found. we describe the structure of database that we have designed. Section Error! Reference source not found. we describe the use development tools.

3.1. Several fragments of C#-code for implementing the basic functionality

**CRUD truck.** Fig. 18 shows the function for adding a truck, the dispatcher can add a truck to the system and also can edit and delete the truck from the system.

```csharp
public partial class trkCRUD : UserControl
{
    //call
    private static trkCRUD _instance;

    public static trkCRUD Instance
    {
        get
        {
            if (_instance == null)
                _instance = new trkCRUD();
            return _instance;
        }
    }

    //Create Edit option boolean
    public bool editMode = false;

    //Create driver ID int
    public int trk_ID;

    public trkCRUD()
    {
        InitializeComponent();
    }
}
```

Fig. 18. CRUD truck code
Report order (fig. 19) shows the function for making report for order; the boss can make report for all orders which contain information about the trucks and drivers.

```csharp
public partial class repOdrView : Form
{
    
    public repOdrView()
    {
        InitializeComponent();
    }
    
    int LX, LY, SW, SH;
    //Window Movement
    [DllImport("user32.dll", EntryPoint = "ReleaseCapture")]
    public extern static void ReleaseCapture();

    [DllImport("user32.dll", EntryPoint = "SendMessage")]
    public extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int lParam);

    private void repOdrView_Load(object sender, EventArgs e)
    {
    }
```  

Fig. 19. Report order code

Complete order (fig. 20). Shows the function for completing order, where the report process is conducted by the manager from the window of the reports in the program and the report includes the name of the driver and the truck and the shipment number and the location of the goods and the location of delivery of the goods and the status of the order.
Assign order (fig. 21). Shows the function for assigning the order, when the order completed, it reaches to the boss and the boss will determine the driver and the type of truck required.

```csharp
public partial class odrComplete : UserControl
{
    //call control
    private static odrComplete _instance;
    public static odrComplete Instance
    {
        get
        {
            if (_instance == null)
                _instance = new odrComplete();
            return _instance;
        }
    }
    //
    public odrComplete()
    {
        InitializeComponent();
    }
    private void odrComplete_Load(object sender, EventArgs e)
    {
    }
}
```

Fig. 20. Complete order code

Fig. 21. Assign order code
Make order (fig. 22) shows the function for making order, The Dispatcher prepares the order for the Boss by adding a customer, the number of items, location, place of delivery and the date of the order.

It is then approved by the Boss who will identify the driver and truck, after approval the order will be ready.

```csharp
public partial class odrCRUD : UserControl
{
    // cal control
    private static odrCRUD _instance;
    10 references
    public static odrCRUD Instance
    {
        get
        {
            if (_instance == null)
                _instance = new odrCRUD();
            return _instance;
        }
    }
}
```

Fig. 22. Make order code

### 3.2. Development of the database

A database system is its structure described in a formal language supported by the database management system DBMS and refers to the organization of data as a blueprint of how a database is constructed (divided into database tables in the case of relational databases) [6].

The formal definition of database schema is a set of formulas sentences called integrity constraints imposed on a database. All constraints are expressible in the same language [2].

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database’s logical structure.

Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery.
Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity [6].

We will describe structure and semantic of every table below.

1. The structure of Driver table

You need to create tables within the databases and then we will define the names of the fields, their types of data and their other equivalents. In the driver's table we have 6 columns which define the property of an object or element. We start with the Driver ID column and here the type must be the primary key.

The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only one primary key, which may consist of single or multiple fields. And data type integer.

The identity specification property is activated from the properties of the column. In the drvOdrstatus column we will use bit data type and this indicates whether something is true or false. The value 0 or 1 is stored.

We have constructed a database schema shown in the Fig. 23.
The tblDriver table stores data about drivers, its structure is depicted in Tab. 1.

Tab. 1. Structure of drivers table.

<table>
<thead>
<tr>
<th>No</th>
<th>attribute</th>
<th>Key</th>
<th>Type</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>drvID</td>
<td>Primary</td>
<td>Int</td>
<td>Unique identity of driver</td>
</tr>
<tr>
<td>2</td>
<td>drvName</td>
<td></td>
<td>nvarchar(50)</td>
<td>Name of Driver</td>
</tr>
<tr>
<td>3</td>
<td>drvAge</td>
<td></td>
<td>Int</td>
<td>Age of Driver</td>
</tr>
<tr>
<td>4</td>
<td>drvPhone</td>
<td></td>
<td>Int</td>
<td>Unique of driver</td>
</tr>
<tr>
<td>5</td>
<td>drvOdrStatus</td>
<td></td>
<td>Bit</td>
<td>Order Status check if driver has order or not</td>
</tr>
<tr>
<td>6</td>
<td>drvNote</td>
<td></td>
<td>nvarchar(150)</td>
<td>Notes of Driver</td>
</tr>
</tbody>
</table>

2. The structure of Customer table

The database table is a customer consisting of 4 columns and will definitely start with cusID and its properties are the same as the properties of trkID and drvID because they are similar in function to each other. CusName The user
data type is nvarchar. Using an n character means using a Unicode, otherwise it means using ASCII [10].

In the Windows code page take numbers from 0 to 255. Latin characters and basic characters occupy numbers from 0 to 127, while the rest of the numbers from 128 to 255 depend on the language used in the device if it is Arabic, this is the Arabic alphabet, although French is the French alphabet and so on. The varchar can only store from 0 to 255 any other language and language.

In nvarchar, any language and language can be stored between languages. In this column we use the integer data type because it contains the correct numbers. In this column we use the integer data type because it contains the correct numbers. In Nvarchar data, so it is possible to have non-English address names.

The tblCustomer table stores data about Customers, its structure is depicted in the Tab. 2.

Tab. 2. Structure of Customers table

<table>
<thead>
<tr>
<th>No</th>
<th>Attribute</th>
<th>Key</th>
<th>Type</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cusID</td>
<td>Primary</td>
<td>Int</td>
<td>Unique identity of Customer</td>
</tr>
<tr>
<td>2</td>
<td>cusName</td>
<td></td>
<td>nvarchar(50)</td>
<td>Name Of Customer</td>
</tr>
<tr>
<td>3</td>
<td>cusPhone</td>
<td></td>
<td>Int</td>
<td>Unique of Customer</td>
</tr>
<tr>
<td>4</td>
<td>cusAddress</td>
<td></td>
<td>nvarchar(100)</td>
<td>Address Of Customer</td>
</tr>
</tbody>
</table>

3. The structure of Truck table

In the truck's spreadsheet we have 7 columns. We have explained in the previous tables what are the columns, but now we will explain what columns are inside the table and start with the trkID column which is for consonant numbering.

trkModel is to insert the type of vehicle, trkModel is to know the vehicle model, trkNo to insert the vehicle number and certainly will be the integer data type, trkLoad and this column for the composite load will also be its integer type, trkOdrStatus and this column will be bit type, trkNote This column is used if there is any note.
The tblTruck table stores all the data about Truck, its structure is depicted in the Tab. 3.

Tab. 3. Structure of the tblTruck table.

<table>
<thead>
<tr>
<th>No</th>
<th>Attribute</th>
<th>Key</th>
<th>Type</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>trkID</td>
<td>Primary</td>
<td>Int</td>
<td>Unique identity of Truck</td>
</tr>
<tr>
<td>2.</td>
<td>trkType</td>
<td>varchar(50)</td>
<td></td>
<td>Type of Truck</td>
</tr>
<tr>
<td>3.</td>
<td>trkModel</td>
<td>varchar(50)</td>
<td></td>
<td>Model of Truck</td>
</tr>
<tr>
<td>4.</td>
<td>trkNo</td>
<td>Int</td>
<td></td>
<td>Unique of Truck</td>
</tr>
<tr>
<td>5.</td>
<td>trkLoad</td>
<td>Int</td>
<td></td>
<td>Number of can Truck load</td>
</tr>
<tr>
<td>6.</td>
<td>trkOdrStatus</td>
<td>Bit</td>
<td></td>
<td>Check if track has order or not</td>
</tr>
<tr>
<td>7.</td>
<td>trkNote</td>
<td>nvarchar(150)</td>
<td></td>
<td>Notes of Truck</td>
</tr>
</tbody>
</table>

4. The structure of Order table

The table of the order contains 11 columns. The first column is ordID. This column is also the primary key. Note that there are columns containing FK (the foreign key) and that the foreign key is a key used to connect two tables together.

A foreign key is a field (or group of fields) in one table that points to the primary key in another table.

The table that contains the foreign key is called the child table, and the table containing the candidate key is called the table referred to or the original.

The advantage of FK is that you can not create a single large table to contain all the data. This is very difficult and will make things difficult for you when you need to generate reports or find individual pieces of data. That is why data is separated into different logical tables [19].

When we make a unique key on a particular field or column for the purpose of not repeating it again within the table. If we want to make a column with the Unique feature, we press Right click, choose Indexes / key, then add, and then specify the column that we want to work on, for example the phone column or e-mail because it is not repeated and then change the is Unique property from
NO to Yes and the type is Unique key. The Dispatch tblOrder stores all the data of orders.

Tab. 4. Structure of the tblOrder table

<table>
<thead>
<tr>
<th>No</th>
<th>Attribute</th>
<th>Key</th>
<th>Type</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>odrID</td>
<td>primary</td>
<td>Int</td>
<td>Unique identity of Order</td>
</tr>
<tr>
<td>2</td>
<td>trkID</td>
<td>FK</td>
<td>Int</td>
<td>Unique identity of Truck</td>
</tr>
<tr>
<td>3</td>
<td>drvID</td>
<td>FK</td>
<td>Int</td>
<td>Unique identity of Driver</td>
</tr>
<tr>
<td>4</td>
<td>cusID</td>
<td>FK</td>
<td>Int</td>
<td>Unique identity of Customer</td>
</tr>
<tr>
<td>5</td>
<td>odrGoodsNo</td>
<td>Int</td>
<td></td>
<td>Number of Goods</td>
</tr>
<tr>
<td>6</td>
<td>odrGoodsLocation</td>
<td>nvarchar(50)</td>
<td></td>
<td>Location Of Customer</td>
</tr>
<tr>
<td>7</td>
<td>odrDelvryLocation</td>
<td>nvarchar(50)</td>
<td></td>
<td>Location Of Delvry</td>
</tr>
<tr>
<td>8</td>
<td>Odrdate</td>
<td>Date</td>
<td></td>
<td>Date Of order</td>
</tr>
<tr>
<td>9</td>
<td>odrType</td>
<td>Bit</td>
<td></td>
<td>Type of Order</td>
</tr>
<tr>
<td>10</td>
<td>odrStatus</td>
<td>Bit</td>
<td></td>
<td>Check if Order is Complete</td>
</tr>
<tr>
<td>11</td>
<td>Driver_Rating</td>
<td>Int</td>
<td></td>
<td>Driver Rating of Order</td>
</tr>
</tbody>
</table>

3.3. The used development tools

I have chosen C# as a programming language for the implementation of my project. C# is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time [20]. C# syntax simplifies many of the complexities of C++ and provides powerful features such as null able value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct [4].

I have chosen SqlServer as DBMS for the implementation of my project.
A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified -- and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures.

Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery, Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity [5].

The DBMS provides a central store of data that can be accessed by multiple users in a controlled manner.
4. TESTING OF THE SYSTEM

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest [8].

Software testing is the art of investigating software in a systematic fashion so as to find deep-rooted defects in it. In addition to that, software testing also checks the quality and correctness of the software. After the errors are identified, it becomes easier to develop bug-free and user-friendly software [6].

As software applications get ever more complex and intertwined and with the large number of different platforms and devices that need to get tested, it is more important than ever to have a robust testing methodology for making sure that software products/systems being developed have been fully tested to make sure they meet their specified requirements and can successfully operate in all the anticipated environments with the required usability and security [9].

We will take three main functions for testing.

1. Test Order Report.

Test the order report: After the process of creating a specific order and after entering all the information related to the application will store information in the database of the program where after pressing the button "Reports" and then on "All Orders" will be provide with a special report of all orders established in advance as shown in fig 24.

![Fig. 24. Order Report](image-url)
2. Test the order completed report.

Test the order completed report: After the process of creating a specific order and after entering all the information related to the application will store information in the database of the program where after pressing the button "Reports" and then on "Completed Orders" will be provide with a special report of Completed orders established in advance as shown in fig 25.

Fig. 25. Order Report

3. Test drivers available.

To provide a report for available drivers we need to press on the button "Reports" and then on "Available driver" will be provide with a special report of available driver established in advance as shown in fig 26.

Fig. 26. Available driver
4. Test drivers unavailable.
To provide a report for unavailable drivers we need to press on the button "Reports" and then on "Unavailable Driver" will be provide with a special report of unavailable driver established in advance as shown in fig 27.

![Unavailable Drivers Report](image)

Fig. 27. Unavailable driver

5. Test the available trucks.
To provide a report for available trucks we need to press on the button "Reports" and then on "Available trucks" will be provide with a special report of available trucks established in advance as shown in fig 28.

![Available Trucks Report](image)

Fig. 28. Available trucks
6. **Test the Unavailable trucks**

   To provide a report for unavailable trucks, we need to press on the button "Reports" and then on "Unavailable trucks" will be provided with a special report of unavailable trucks established in advance as shown in fig 29.

   ![Unavailable Trucks Report](image)

   **Fig. 29. Unavailable trucks**
CONCLUSION

This thesis was aimed to development of Cargo Delivery Management System for Small Transport Company.

The following basic results have been achieved:

1. The use case model of the system has been created.
2. The user interface forms has been designed.
3. The database structures has been designed.
4. System structure was designed and implemented.
5. Test cases have been created and the system has successfully passed all the tests.
REFERENCE LIST


