**MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE**
**Sumy State University**
Academic and Research Institute of Business, Economics and Management
(full name of the institute/faculty)
Economic Cybernetics Department
(full name of the department)

«Admitted to the defense»
Head of Department
_____ Vitaliia KOIBICHUK
(signature) (First and LAST NAME)
_____ _____ 20__ p.

# QUALIFICATION WORK
**to obtain an educational degree** ___bachelor___
(bachelor / master)
from the specialty 051 Economics,
(code and name)
educational-professional programs Business Analytics
(educational-professional / educational-scientific) (the name of the program)
on the topic: Development of an Information System Supporting Electronic Payments

Winner(s) of the group AB-91a.an Nagorny Mykola Vitaliyovych
(group cipher) (full name)

The qualification work contains the results of own research. The use of ideas, results and texts of other authors are linked to the corresponding source

_____ Mykola NAGORNY
(signature) (Name and SURNAME of the acquirer)

Head
Candidate of technical sciences, associate professor
Kostiantyn HRYTSENKO _____
(position, academic degree, academic title, Name and SURNAME) (signature)

Consultant_____ _____
(position, academic degree, academic title, Name and SURNAME) (signature)

**Sumy – 2023**

# SUMMARY

bachelor's thesis on the topic

“Development of an Information System Supporting Electronic Payments”

student Nagorny Mykola Vitaliyovych

(last name, first name, patronymic of the student)

In today's world prone to automation, it is impossible to count the number of different kinds of software products. Every day, a huge number of applications appear on the software market, and their relevance is growing exponentially.

Based on this situation of the software market, it can be noted that software development is a relevant and necessary topic.

The main goal of this course project is to create a software implementation of a system for supporting electronic payments via the Internet.

The object of research is methods and means of creating software products.

The subject of the study is the creation of a software implementation of a system for supporting electronic payments via the Internet.

To fulfill the set goal, the following tasks should be completed:

• conduct an analysis of the subject area;

• analyze existing solutions;

• analyze the tools;

• design the system;

• develop an information structure;

• develop a class diagram;

• develop an interface and examples of work.

To achieve the set goal and tasks of the research, the following were used: fundamental concepts of theoretical and methodological research on ways of developing applied software interfaces. A set of general practices for database design.

The information base of the qualifying bachelor thesis was made up of the results of the pre-diploma practice.

The main scientific results of the bachelor's work are a developed project that allows you to get a convenient application that allows you to quickly perform banking operations.

Practical development was carried out in VS Code.

The obtained results can be used by economic and analytical and IT departments.

Keywords: Ssharp, visual studio, generation, database, .NET, development optimization, CLI, client-server, UML, GUI, database.

The content of the master's thesis is presented on 36 pages. The list of used sources from 27 names, placed on 3 pages.

The year of Bachelor’s thesis fulfillment is 2023.

The year of Bachelor’s thesis defense is 2023.

Ministry of Education and Science of Ukraine
Sumy State University
Academic and Research Institute of Business,
Economics and Management
Department of Economic Cybernetics

APPROVE
Head of Department
Candidate of Economics,
associate professor
_____ Koibichuk V.V.
"__" _____ 2023.

TASKS FOR THE QUALIFICATION WORK OF THE BACHELOR
specialty 051 "Economics" (Business Analytics)
4th year student, group AB-91a.an
Nagorny Mykola Vitaliyovych
(last name, first name, patronymic of the student)

1. Topic of the work: Development of an Information System Supporting Electronic Payments
approved be order of the university «15» 06 2023 № 0674-VI
2. The deadline for submission of completed work by the student «16» 06 2023
3. The purpose of the qualification work: development of an information system supporting electronic payments.
4. The object of study: the process of electronic payments
5. The subject of study: modern software tools, models and techniques for development information systems supporting electronic payments.
6. The qualification work is performed on materials: based on various studies in the field of electronic payments.
7. Indicative plan of qualification work, terms of submission of sections to the head and the maintenance of tasks for performance of the set purpose

Chapter 1. STUDY OF THE STATE OF BUSINESS PROCESS AUTOMATION AND INFORMATION SYSTEM DESIGN
(name – submission deadline)

In chapter 1.
1.1 Characteristics of the research object, analysis of the state of automation of business processes

1.2 Formation of requirements for the information system
1.3 Architecture of the information system and technologies for solving the tasks
1.4 Functional structure of the task and subsystems for providing functional part
(the content of specific tasks to the section that the student must perform)

Chapter 2. IMPLEMENTATION OF THE PROTOTYPE OF THE INFORMATION SYSTEM
(name – submission deadline)

In chapter 2.
2.1 Structure and features of the implementation of information support
2.2 Structure and features of implementation of algorithmic support
2.3 Test case and instructions for use
(the content of specific tasks to the section that the student must perform)

8. Consultations on work:

| Chapter | | Surname, initials and position of consultant | Signature, date | |
|---|---|---|---|---|
| | | | Task issued by | Task accepted by |
| 1 | | | | |
| 2 | | | | |

9. Date of issue of the task: «03» 04 2023

Supervisor _____ Hrytsenko K.G.
(signature) (initials, surname)

Received the task to perform _____ Nagorny M.V.
(signature) (initials, surname)

# CONTENT

# INTRODUCTION

In today's world prone to automation, it is impossible to count the number of different kinds of software products. Every day, a huge number of applications appear on the software market, and their relevance grows exponentially.

Based on this situation of the software market, it can be noted that software development is a relevant and necessary topic.

The main goal of this course project is to create a software implementation of a system for supporting electronic payments via the Internet.

The object of research is methods and means of creating software products.

The subject of the study is the creation of a software implementation of a system for supporting electronic payments via the Internet.

To fulfill the set goal, the following tasks should be completed:

- conduct an analysis of the subject area;
- analyze existing solutions;
- analyze the tools;
- design the system;
- develop an information structure;
- develop a class diagram;
- develop an interface and examples of work.

# 1 STUDY OF THE STATE OF BUSINESS PROCESS AUTOMATION AND INFORMATION SYSTEM DESIGN

## 1.1 Characteristics of the research object, analysis of the state of automation of business processes

Management, also known as governance, encompasses the processes and practices involved in overseeing and directing an organization, whether it be a business, nonprofit, or government agency. It combines both art and science to effectively utilize various resources within the organization.

The core activities of management involve formulating the organization's strategy and coordinating the efforts of its employees or volunteers to achieve the desired goals. This requires leveraging available resources such as financial, natural, technological, and human resources. Two key concepts in management are "doing business" and "changing business"[1] The former refers to the continuous provision of goods or services, while the latter focuses on adapting goods or services to meet evolving customer needs in response to market trends.

The term "management" can also refer to the individuals responsible for running the organization, commonly known as managers. Many individuals pursue formal education in management by studying at colleges or universities. Popular management degrees include Bachelor of Commerce (B.Com.), Bachelor of Business Administration (BBA.), Master of Business Administration (MBA.), Master of Management (MSM or MIM), and Master of Public Administration (MPA) for those in the public sector. Those aiming to become management specialists, researchers, or professors can pursue advanced degrees such as Doctor of Management (DM), Doctor of Business Administration (DBA), or PhD in Business Administration or Management. In recent years, there has been a growing emphasis on evidence-based management, which relies on empirical research and data-driven decision-making.

In larger organizations, a hierarchical structure with three levels of managers is commonly observed. At the top are the top leaders, such as the board

of directors, CEO, or president, who establish the organization's strategic goals, policies, and overall operational framework. Senior managers, who report to top leaders, oversee and manage the organization's operations. Middle managers, including branch managers, regional managers, department heads, and section managers, act as a bridge between top leaders and first-line managers, ensuring the communication and implementation of strategic goals [2].

First-line managers, such as supervisors and front-line managers, directly supervise regular workers or volunteers and provide guidance in their day-to-day tasks. They are responsible for implementing the organization's policies and are considered part of the workforce rather than the management class.

In smaller organizations, managers often have broader roles and may perform multiple functions typically divided among different managers in larger organizations.

Sociologists also study management as an academic discipline, exploring various aspects such as social organization, organizational adaptation, and leadership within organizations [4].

Etymology:

The origin of the English verb "steer" can be traced back to the 15th-century French verb "mesnager," which often referred to holding the reins of a horse in equestrian language. Another possible root is the Italian term "maneggiare," meaning to handle, especially tools or a horse. Similarly, in Spanish, "manejar" can also mean managing horses [6]. All three terms stem from the combination of two Latin words: "manus" (hand) and "agere" (to act).

The French word for household, "ménagerie," is derived from "ménager," which means "to keep house" and is related to "ménage" meaning "household." It also encompasses the care of domestic animals. "Ménagerie" serves as a French translation of Xenophon's renowned book "Economics" [7] (Greek: Οἰκονομικός) that focuses on household and household management. The French term "mesnagement" or "ménagement" had an influence on the semantic development of the English word "management" during the 17th and 18th centuries [8]

Levels:

In most organizations, management is structured into three levels: first-level managers, middle-level managers, and top-level managers. First-level managers are at the lowest level and oversee the work of non-managerial individuals directly involved in the organization's production or creation process. They are commonly referred to as supervisors, but they may also be called line managers, office managers, or foremen.

Middle managers encompass all levels of management between the first-line managers and the top level of the organization. Their responsibilities include supervising first-line managers and holding positions such as department managers, project managers, plant managers, or division managers.

Top managers are responsible for making decisions that impact the entire organization and setting plans and goals accordingly. They hold positions such as executive vice president, president, managing director, chief operating officer, chief executive officer, or chairman of the board. Top managers play a crucial role in making organization-wide decisions, developing strategic plans, establishing company policies, and providing overall direction.

Managers are arranged in a hierarchical structure based on authority and have distinct roles and tasks. In many organizations, the distribution of managers at each level resembles a pyramid shape. The specific duties and job titles associated with each level vary, and the following section provides further details [26].

Senior Management:

Senior management refers to a small group within an organization, including the board of directors (comprising non-executive directors, executive directors, and independent directors), the president, vice president, CEOs, and other C-level executives. The composition of this group may differ across organizations, potentially including positions such as CFO (Chief Financial Officer) or CTO (Chief Technology Officer). Senior managers are responsible for overseeing and supervising the entire organization's activities. They establish the

"tone at the top," develop strategic plans, company policies, and make decisions regarding the overall direction of the organization. Additionally, senior managers play a significant role in mobilizing external resources. They are accountable to shareholders, the general public, and government agencies responsible for overseeing corporations and similar organizations. Some members of the top management may also serve as the public face of the organization, giving speeches to introduce new strategies or speaking in marketing efforts.

A board of directors typically consists mainly of non-executives who have a fiduciary duty to shareholders and are not directly involved in the organization's day-to-day operations. However, the level of involvement may vary based on factors such as the organization's type (public or private), size, and culture. These directors have a legal responsibility for fulfilling their duties and are usually protected by directors and officers liability insurance. Fortune 500 CEOs are estimated to spend around 4.4 hours per week on board duties, and their average compensation was $212,512 in 2010. The board is responsible for setting corporate strategy, making significant decisions like major acquisitions, and hiring, evaluating, and terminating top-level managers, including the CEO [27]. While the CEO usually handles hiring for other positions, such as the CFO, board involvement in these appointments has increased [28]. In a 2013 survey, weaknesses identified in CEOs by CEOs and directors of public and private companies included "mentoring skills" and "board involvement," and 10% of companies never evaluated their CEO's performance [29]. The board may also oversee certain employees, such as internal auditors, or directly hire independent contractors. For instance, the board, typically through the audit committee, often selects the auditor.

The skills necessary for senior management positions vary based on the type of organization but generally include [30] a comprehensive understanding of competition, the global economy, and politics. Additionally, the CEO is responsible for implementing and defining the organization's broad policies within the framework set by the board. Executive management handles day-to-

day operational details, including tasks such as preparing department budgets, establishing procedures and schedules, appointing middle managers (such as department heads), coordinating departments, managing relations with the media and government, and communicating with shareholders.

Middle management consists of general managers, branch heads, and department heads who are accountable to senior management for their department's functions. They devote more time to organizational and management functions. Their roles include implementing organizational plans aligned with company policies and top management objectives, conveying information and policies from top management to lower-level management, and most importantly, inspiring and guiding lower-level managers for improved performance.

Middle management serves as an intermediate layer within categorized organizations, situated between top management and the highest levels of operational staff. An operations manager can be a well-defined middle manager or a non-managerial employee responsible for specific organizational policies. The performance of middle managers is crucial as they bridge the gap between upper-level and lower-level staff. Their functions encompass developing and implementing effective group and intergroup work and information systems, defining and monitoring performance indicators at the group level, diagnosing and resolving problems within and between work groups, and designing and implementing reward systems that promote cooperation. They also make decisions and share ideas with top managers.

Line management includes supervisors, department heads, managers, and group leaders who focus on the control and direction of full-time employees. Their responsibilities typically involve assigning employee tasks, overseeing day-to-day activities, ensuring product or service quality and quantity, providing recommendations and suggestions to employees, and escalating employee concerns that cannot be resolved to middle managers or other administrators. Some first-line managers may also lead by example and perform tasks alongside employees, especially during busy periods. For example, in certain restaurants,

first-line managers may also serve customers during peak hours. Overall, line managers are considered part of the workforce rather than the formal management class within organizations, regardless of their traditional managerial functions.

First-line managers typically fulfill the following roles:

- Providing training to new employees.
- Offering basic supervision to ensure tasks are completed effectively.
- Motivating employees to enhance their performance.
- Providing feedback on performance and leadership to guide improvement. Some top managers may also engage in career planning for employees who aspire to advance within the organization.

An e-commerce payment system, also known as an e-payment system, facilitates the acceptance of electronic payments for online transactions. It is considered a subcomponent of Electronic Data Interchange (EDI) and has gained significant popularity due to the widespread use of online shopping and banking services.

Credit cards remain the most prevalent form of payment for e-commerce transactions. In North America, approximately 90% of online retail transactions utilized credit cards as of 2008 [1]. Due to their extensive usage, online stores find it challenging to operate without supporting credit and debit card payments. Compliance with regulations from credit and debit card issuers, such as Visa and Mastercard, is crucial for online sellers to ensure adherence to banking and financial guidelines in the countries [2] where the services operate.

For most payment systems available on the public Internet, securing basic authentication (of the financial institution on the receiving end), data integrity, and confidentiality of electronic information exchanged over public networks require obtaining a certificate from an authorized Certificate Authority (CA). This certificate is based on a public-key infrastructure (PKI) that provides an open-source infrastructure. While Transport Layer Security (TLS) is implemented to protect transactions over public networks, especially in payment systems, the

customer-facing website must be meticulously coded to prevent credential leakage and safeguard customer personal data from theft.

While credit card usage is widespread in North America, certain countries like China and India still face credit card security challenges. To enhance security, measures such as Card Verification Numbers (CVN) have been implemented. CVN compares the verification number on the card's signature strip with the information stored at the issuing bank to detect fraud.

Various companies specialize in online financial transactions, offering services for credit card processing (e.g., Stripe), direct online bank payments (e.g., Smartpay), and alternative payment methods at checkout (e.g., PayPal). Intermediaries enable consumers to swiftly set up accounts and transfer funds between their online and traditional bank accounts, often utilizing Automated Clearing House (ACH) transactions.

The popularity of cyber-broker accounts, despite associated risks of theft and abuse, stems from their ease of use. However, resolving issues with these accounts can be challenging. The information asymmetry within large financial institutions regarding information safeguards leaves end users with limited understanding of the system. Consequently, when funds are misused, users often accuse intermediaries of negligence or illegal behavior. Trust between the public and banking corporations is further strained when major financial institutions are found to have abused their power asymmetrically, as seen in the Wells Fargo fraud scandal in 2016.

Electronic payment methods encompass various types, including online credit card transactions, electronic wallets, electronic cash, and wireless payment systems. Credit cards are widely used but can be costly for merchants due to transaction fees. Debit cards provide a secure alternative at a lower cost. Additionally, alternative payment methods have emerged, with some even claiming market leadership.

Bank payments are a cardless system used by clients connected to internet banking. Instead of entering card details on the buyer's website, users select their

bank through the payment gateway. They are then redirected to the bank's website, where they authenticate and confirm the payment. This method generally incorporates two-factor authentication and is considered more secure than credit cards since hackers find it more difficult to obtain credentials than credit card numbers. Offering the option to pay with bank accounts reduces abandoned carts for e-commerce merchants, enabling transactions without relying solely on credit cards.

In developing countries, where banking services are limited, mobile money wallets have gained prominence. For instance, in India, where there are more mobile phone users than active bank account holders, telecom operators offer mobile money wallets. These wallets allow users to top up using their existing mobile prepaid numbers at physical top-up points, converting cash into mobile wallet currency. These mobile wallets can be used for online transactions and e-commerce purchases.

Banks, as financial institutions, accept deposits from the public and provide loans, creating demand deposits in the process. Due to their crucial role in financial stability and the economy, banks are subject to extensive regulation in most jurisdictions. Fractional reserve banking, where banks hold liquid assets that represent only a fraction of their liabilities, is a common practice. Banks are also subject to minimum capital requirements based on international standards such as the Basel Accords, along with other regulations designed to ensure liquidity.

Banking, as we understand it today, originated during the fourteenth century in the prosperous cities of Renaissance Italy. However, it can be seen as a continuation of ancient credit and borrowing concepts. Throughout history, several influential banking dynasties, including the Medici, Fuggers, Welsers, Berenbergs, and Rothschilds, played significant roles in shaping the banking industry over many centuries. The oldest existing retail bank is Banca Monte dei Paschi di Siena, established in 1472, while the oldest existing commercial bank is Berenberg Bank, founded in 1590.

The definition of a bank varies from one country to another, and specific details can be found in the relevant country's regulations and laws. Under English common law, a banker is described as an individual who conducts the business of banking by maintaining current accounts for customers, honoring checks drawn on those accounts, and collecting checks on behalf of customers [17].

Most common law jurisdictions have a Bills of Exchange Act that encompasses the laws related to negotiable instruments, including cheques. Within this act, there is a statutory definition of the term "banker": "A banker includes a body of persons, incorporated or unincorporated, engaged in the business of banking" (Chapter 2, Interpretation). While this definition may appear specific, it serves a functional purpose by ensuring that the legal framework for banking transactions, such as checks, is not dependent on the bank's structure or regulation.

In many common law countries, banking is not explicitly defined by legislation but rather by common law, as described earlier. However, in some English common law jurisdictions, there are statutory definitions of banking. It is important to note that these definitions pertain to the specific legislation and its regulatory objectives, rather than defining banking in a general sense. Nonetheless, these statutory definitions often align closely with the common law definition. Here are a few examples of statutory definitions:

- "Banking activities" refers to activities related to receiving money into current or deposit accounts, processing payment and collection of checks issued or paid by customers, providing advances to customers, and any other activities prescribed by the relevant authority for the purposes of the act (Banking Act, Singapore, Chapter 2, Interpretation).
- "Banking activity" encompasses the activity of either or both of the following:
    - Receiving money from the public for current, deposit, savings, or similar accounts that are repayable on demand or within a period

shorter than three months or with a notice period shorter than this period.

- Paying or receiving checks written or paid by customers [18].

As electronic funds transfer at the point of sale (EFTPOS), direct credit, direct debit, and internet banking have become prevalent, the prominence of checks as a payment system tool has diminished in most banking systems. This has led experts to suggest that the definition of banking, which traditionally revolved around checks, should be expanded to include financial institutions that maintain current accounts for customers and facilitate the receipt and payment of funds to third parties, even if checks are not involved. Banks, acting as agents of payment banks, maintain current accounts for customers, process checks issued by customers, and handle checks deposited into customers' accounts. Additionally, banks enable customers to conduct payments using alternative methods such as automated clearinghouse (ACH), bank transfers, telegraphic transfers, EFTPOS, and automated teller machines (ATMs).

Banks engage in lending activities by accepting funds deposited into current accounts, offering term deposits, and issuing debt instruments like banknotes and bonds. They provide loans by extending advances on current accounts and offering installment loans, while also investing in various forms of market debt instruments and other types of lending.

Payment services are a crucial offering provided by banks, and both businesses and individuals generally consider having a bank account indispensable. Non-banking organizations that offer payment services, such as companies facilitating monetary remittances, are typically not seen as adequate substitutes for traditional banking accounts.

When banks provide loans, they create new money in the process. In modern regulatory banking systems, governing bodies establish minimum reserve requirements that banks must maintain to ensure they can meet their deposit obligations arising from the financing of these loans. Banks can acquire these

reserves by attracting new deposits, selling assets, or borrowing from other banks, including the central bank.

Banks employ various business models to generate income, including interest, transaction fees, and financial advice. Traditionally, the primary source of income has been the interest charged on the capital lent to customers. Banks profit from the spread between the interest they pay on deposits and other funding sources and the interest they charge for their lending activities. This spread is known as the interest rate spread or the cost of funds. However, banks also derive revenue from fees and financial advisory services, which provide a more stable income stream and help to smooth out their financial performance.

In response to the increasingly volatile market conditions of the past 20 years, American banks have implemented various strategies to maintain profitability. These measures include:

1. The Gramm-Leach-Bliley Act, which permits banks to merge with investment and insurance companies. This integration of banking, investment, and insurance functions allows banks to meet consumer demand for a comprehensive range of services, thereby increasing their potential for cross-selling products and enhancing profitability.
2. The expansion of risk-based pricing from business lending to consumer lending. This involves charging higher interest rates to customers considered higher credit risks, which helps offset potential losses from loan defaults. It also lowers the cost of loans for customers with better credit and enables the provision of loan products to high-risk customers who would otherwise be denied credit.
3. The diversification of payment processing methods available to the general public and business customers. This includes the introduction of debit cards, prepaid cards, smart cards, and credit cards. These payment options enhance convenience for consumers and facilitate smoother consumption patterns over time. In some countries with underdeveloped financial

systems, cash-only transactions are still prevalent, but the introduction of electronic payment methods reduces reliance on physical currency.

4. However, the ease of accessing credit also poses risks, such as consumers mismanaging their finances and accumulating excessive debt. Banks generate revenue from card products [22]. through interest charges and fees imposed on cardholders, as well as transaction fees charged to retailers who accept the bank's credit and/or debit cards for payments [23]. This contributes to profitability and overall economic development.

Moreover, in the face of challenges from financial technologies, banks have explored new business models such as freemium, data monetization, white-label banking, and cross-selling additional products. These endeavors aim to adapt to evolving market dynamics and maintain a competitive edge [24].

Banking crisis:

Banks face various risks that can lead to systemic crises. These risks include liquidity risk (when depositors demand more funds than the bank has available), credit risk (the probability of borrowers defaulting on their obligations to the bank), and interest rate risk (the bank becoming unprofitable due to increased interest rates that result in higher interest payments on deposits compared to interest earned on loans).

Throughout history, banking crises have occurred when there were one or multiple risks that affected the banking sector as a whole. Notable examples include bank runs during the Great Depression, the savings and loan crisis in the USA during the 1980s and early 1990s, the Japanese banking crisis in the 1990s, and the subprime mortgage crisis in the 2000s.

Size of the global banking industry:

The assets of the world's 1000 largest banks reached a record 96.4 trillion US dollars in the fiscal year of 2008-2009, representing a 6.8% increase. However, profits decreased by 85% to 115 billion US dollars during the same period. The growth in assets during challenging market conditions was largely driven by recapitalization efforts

In terms of regional distribution, EU banks accounted for the largest share at 56%, compared to 61% in the previous year. The proportion of Asian banks increased from 12% to 14%, and US banks saw an increase from 11% to 13%. In 2009, commission income from global investments in banking amounted to 66.3 billion US dollars, a 12% increase from the previous year.

The United States has the largest number of banks in the world in terms of institutions (5,330 as of 2015) and possibly branches (81,607 as of 2015) [31]. This is due to the geographic and regulatory structures of the US banking system, which accommodate a large number of small and medium-sized institutions. China's leading banks have over 67,000 branches (ICBC:18000+, BOC:12000+, CCB:13000+, ABC:24000+) with an additional 140 smaller banks of uncertain quantity branches, and Japan has 129 banks and 12,000 branches. Germany, France, and Italy had more than 30,000 branches each in 2004, while the United Kingdom had 15,000 branches.

Mergers and acquisitions:

Between 1985 and 2018, banks were involved in approximately 28,798 mergers or takeovers as either buyers or targeted companies. The estimated cost of these deals is around 5,169 billion US dollars. There were two major waves of mergers and acquisitions in 1999 and 2007, with a peak cost of 460 billion US dollars in the US, followed by a significant decline of 82% from 2007 to 2018.

## 1.2 Formation of requirements for the information system

The essence of the problem is the storage and processing of data related to electronic payments.

The main objects are the database tables linked to the project.

The main entities of the database:

- client;
- customer account;
- outgoing transaction;
- incoming transaction;

Figure 1.1 shows the information model of the problem.


Figure 1.1 — Information model of the problem

The information model of the problem illustrates the operation of the system, which consists in processing requests to the database.

The output information is data related to transactions, which the system as a whole is aimed at saving and processing.

Table 1.1

List and description of outgoing messages

| No | Name | Identifier | Application form and requirements for it | Publication frequency | Issue period and permissible delay time | Information users |
|---|---|---|---|---|---|---|
| 1 | Incoming transaction | Incoming transaction | In the form of records in the database | Upon request | Upon request | Any system user |
| 2 | Outbound transaction | Outbound transaction | In the form of records in the database | Upon request | Upon request | Any system user |

All indicators do not have numerical equivalents and are only informative.

Input information is data about customers and their accounts that are used to generate transactions.

Table 1.2

List and description of outgoing messages

| No | Name | Identifier | Application form and requirements for it | Publication frequency | Issue period and permissible delay time | Information users |
|---|---|---|---|---|---|---|
| 1 | Customer data | Customer data | In the form of records in the database | Upon request | Upon request | Any system user |
| 2 | Account data | Account data | In the form of records in the database | Upon request | Upon request | Any system user |

All indicators do not have numerical equivalents and are only informative.

### 1.3 Architecture of the information system and technologies for solving the tasks

C# (pronounced C♯ note but written in numeric symbols) is a general multi-paradigm programming language that includes strongly typed, lexically scoped, imperative, declarative, functional, generic object-oriented (class-based), and component-oriented disciplines programming. It was developed by Microsoft as part of the .NET plan around 2000 and later approved as international standards by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2018). Mono is the name of a free, open source project used to develop a compiler and runtime for the language. C# is one of the programming languages developed for Common Language Infrastructure (CLI).

C# was developed by Anders Halesberg , whose development team is currently headed by Mads Torgersen . The latest version is 8.0, which will be released with Visual Studio 2019 version 16.3 in 2019.

During the development of the .NET Framework , the class library was originally written using a managed code compiler called Simple Managed C (SMC). In January 1999, Anders Halesberg formed a team and created a new language called Cool , which means "C-type object-oriented language". Microsoft decided to keep the name " Cool " as the final name of the language, but chose not to because of trademarks. When the .NET project was publicly announced at the Professional Developers Conference in July 2000, the language was renamed C# and the ASP.NET class library and runtime were moved to C#.

Halesberg is the lead designer and chief architect of C#. He designed the Turbo Pascal , Embarcadero Delphi (formerly CodeGear Delphi , Inprise Delphi and Borland Delphi ) and Visual J++. In his interviews and technical articles, he stated that the shortcomings of most major programming languages (such as C++, Java , Delphi , and Smalltalk ) led to the creation of the Common Language Runtime (CLR), which in turn led to the development of the C# language itself.

James Gosling , who created the Java programming language in 1994, and Bill Joy , co-founder of the Java developer Sun Microsystems called C# an

"imitation" of Java ; Gosling went on to say that “C# is a kind of technology that removes reliability, performance, and security. Java ". Klaus Kreft and Angelika Langer (author of the book on C++ threads) stated in their blog that " Java and C# are almost the same programming language. There is no boring repetition of innovation", "Almost no one would argue that Java or C# are revolutionary programming languages, they changed the way programs were written," "C# borrowed a lot from Java —and vice versa. Now C# supports packaging AND, unpacking, we'll have very similar functionality in Java . In July 2000, Halesberg stated that C# "is not a Java clone " and "closer to C++" in its design.

Since the release of C# 2.0 in November 2005, C# and Java have evolved on increasingly different trajectories, becoming two completely different languages. The first major deviation is adding a generalization to two languages with completely different implementations. C# uses reification to provide "first-class" generic objects that can be used like any other class and perform code generation when the class is loaded. Additionally, C# adds several major features to functional programming, and finally, the LINQ extension released in C# 3.0 and supported lambda expressions, extension methods, and anonymous type structures. These functions allow C# programmers to use functional programming techniques such as blocking when using them. Extending and importing LINQ functions can help developers reduce the amount of boilerplate code involved in common tasks (such as querying a database, parsing XML files, or looking up a data structure) and shift the focus to the actual application logic to improve readability and maintainability.

By design, C# is a programming language that directly maps the underlying common language infrastructure (CLI). Most of its internal types correspond to value types implemented by the CLI. However, the language specification contains no requirements for the generation of compiler code, that is, there is no provision that the C# compiler must target a common language runtime, or generate a common intermediate language (CIL), or generate any other specific

format. In theory, the C# compiler can generate machine code like traditional C++ and Fortran compilers .

C# supports strict type declarations of implicit variables using the var keyword and implicitly typed arrays using the new [] keyword followed by a collection initializer .

C# supports the strict boolean data type bool . Statements that accept conditions such as while and if require expressions of types that implement true operators, such as boolean types. Although C++ also has boolean types, it can be freely converted to integers and integers as well as expressions . For example , if (a) requires only a boolean value and allows a to be an int or a pointer. C# prohibits this "integer true or false" method on the grounds that forcing programmers to use expressions that only return bool can prevent certain types of programming errors, such as if (a = b) (use the assignment = instead of the equal sign == ).

As far as types are concerned, C# is safer than C++. By default, the only implicit conversions are those that are considered safe, such as integer expansions. This is used at compile time, at JIT time, and in some cases at runtime. There is no implicit conversion between booleans and integers, and between enumeration members and integers (except for the literal 0, which can be implicitly converted to any enumeration type). Any managed conversion must be clearly marked as explicit or implicit, which is different from C++'s implicit copy constructor and default conversion operator.

C# has explicit support for covariance and contravariance in generic types, unlike C++, only because the return type semantics of virtual methods have some degree of support for contravariance. The listed participants are placed in the field of view.

C# does not allow global variables or functions. All methods and members must be declared in a class. Static members of public classes can replace global variables and functions.

Unlike C and C++, local variables cannot hide variables contained within a block.

Delegate keyword. Like the pseudo-C++ framework Qt , C# also has semantics around publish-subscribe style events, for example, although C# uses delegates for this.

Managed memory cannot be freed; instead, it is collected automatically. Garbage collection solves the problem of memory leaks and reduces the responsibility of the programmer to release memory that is no longer needed.

Unlike C++, C# does not support multiple inheritance, although a class can implement any number of interfaces. This is a design decision made by a leading language architect to avoid complexity and simplify the architectural requirements of the entire CLI. When implementing multiple interfaces that contain methods with the same signature, that is, two methods with the same name take the same parameter type in the same order. C# allows you to implement each method according to the interface that calls that method, or like Java , which allows you to implement that method once and call it every time through any class interface.

However, unlike Java , C# supports operator overloading. Only the operators that are most frequently overloaded in C++ can be overloaded in C#.

C# can use LINQ through the .NET Framework . Developers can query any IEnumerable <T> object, XML document, ADO.NET dataset, and SQL database. [60] Using LINQ in C# provides benefits such as Intellisense support , powerful filtering capabilities, type safety against compile-time errors, and consistency of query data from different sources. There are several different language frameworks that can be used with C# and LINQ. These are query expressions, lambda expressions, anonymous types, implicitly typed variables, extension methods, and object initializers .

## 1.4 Functional structure of the task and subsystems for providing the functional part

The client-server model is a framework for distributed applications that involves the distribution of tasks or workloads between service providers (servers) and service requesters (clients). Typically, the client and server communicate over a computer network, although they can also reside on the same system. The server host runs one or more server programs that share resources with the client, while the client primarily requests content or services from the server without sharing its own resources. In this model, the client initiates a communication session with the server, which waits for incoming requests. Various computer programs, such as email, network printing, and the World Wide Web, rely on the client-server model.

The client-server relationship describes the interaction between programs in a system. A server component offers a specific function or service to one or more clients that initiate requests for that service. Servers are categorized based on the services they provide. For instance, a web server hosts web pages, and a file server manages computer files. The shared resources in this context can include server software, electronic components, programs, data, processors, and storage devices. Sharing server resources is a fundamental aspect of the services provided by the server.

The general prototype of the system is shown in Figure 1.3.

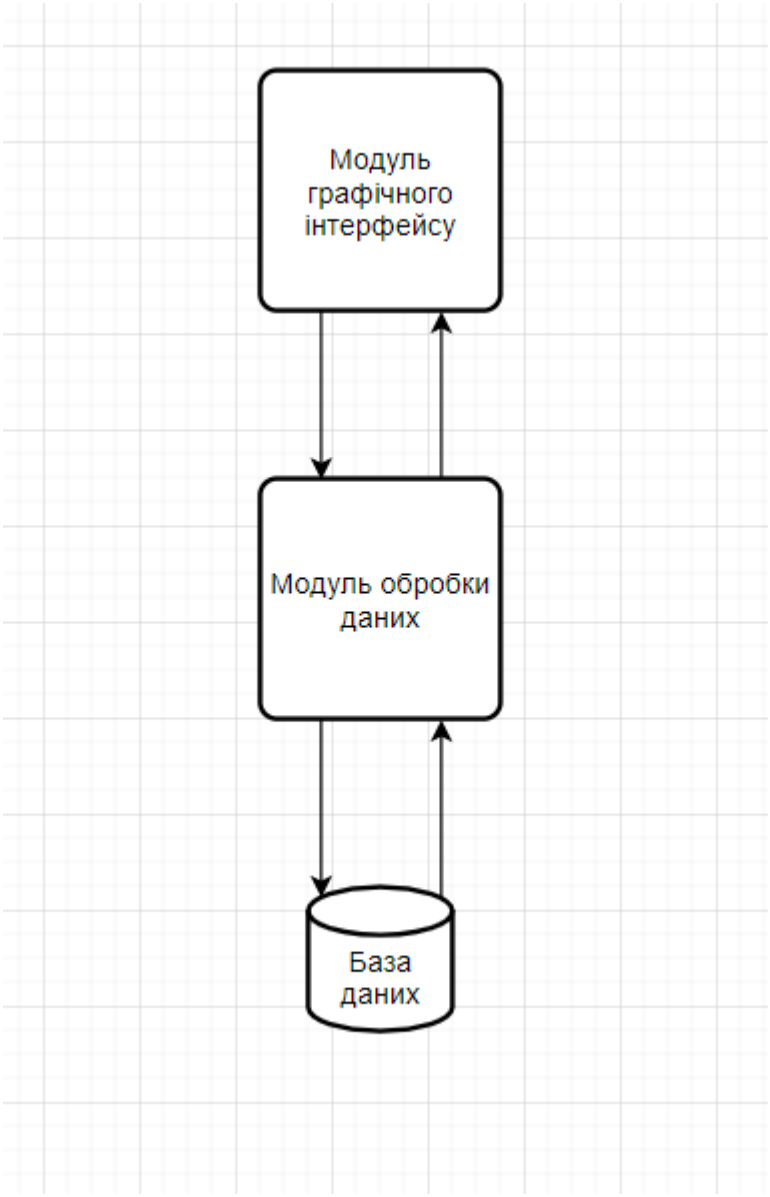


Figure 1.3 — System prototype

# 2 IMPLEMENTATION OF THE PROTOTYPE OF THE INFORMATION SYSTEM

## 2.1 Structure and features of the implementation of information support

The structure of the database of the information system consists of four tables, namely:

- client;
- score;
- transaction;
- payment.

The scheme of the database is shown in Figure 2.1.

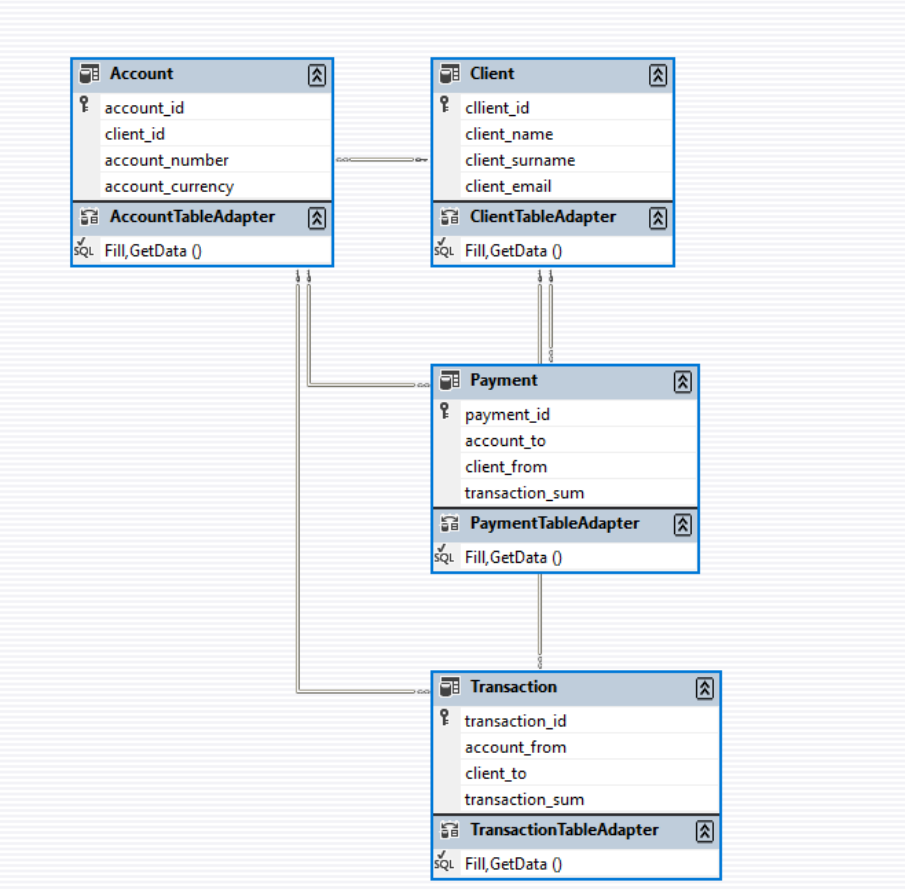


Figure 2.1 — Database scheme

For a more detailed understanding of the structure of the information system, you should use the class diagram (Figure 2.2)

To establish the internal structure of the system, a UML class diagram was created to visually represent the project's inner workings.

In the realm of UML, class diagrams are one of the six types of structure diagrams and play a crucial role in the process of object modeling. They primarily depict the static structure of a system. Depending on the system's complexity, a single class diagram can be used to model the entire system, or multiple class diagrams can be employed to represent different system components.

Class diagrams serve as blueprints for systems or subsystems. They allow for the modeling of system objects, showcasing the relationships between these objects, and describing their functionalities and provided services.

Class diagrams prove to be valuable throughout various stages of system design. During the analysis phase, a class diagram aids in understanding the requirements of the problem domain and identifying its components. In object-oriented software projects, the class diagrams created at an early stage often translate into actual software classes and objects as the code is written. Later on, the preliminary analysis and conceptual models can be refined into class diagrams that specifically depict various aspects of the system, such as user interfaces, logic implementations, and more. These class diagrams then serve as a snapshot that accurately describes the system's functionality, the relationships between system components at different levels, and the planned implementation approach.

Class diagrams have multiple applications in visualizing, defining, and documenting structural features within models. For instance, during the analysis and design stages of the development cycle, class diagrams can be created to fulfill the following functions:

- Collect and define the structure of classes and other classifiers.
- Define relationships between classes and classifiers.
- Illustrate the model's structure using attributes, operations, and signals.

- Depict the general roles and responsibilities of classifiers that determine the system's behavior.
- Show the implementation classes within a package.
- Illustrate the structure and behavior of one or more classes.
- Display the inheritance hierarchy among classes and classifiers.
- Represent workers and entities as business object models.

During the implementation phase of the software development cycle, class diagrams are employed to convert models into code and vice versa.

The following topics describe the model elements encountered in class diagrams:

1. Class: In UML, a class represents an object or a group of objects that share common structure and behavior. Classes and instances of classes are commonly used model elements within UML diagrams.
2. Objects: Objects in UML models are model elements that represent instances of classes. Objects can be added to the model to represent specific instances, whether they are concrete or prototype instances. A concrete instance represents a real person or thing in the real world. For example, a concrete instance of the Customer class represents an actual customer. On the other hand, a prototype instance of the Customer class contains data that represents a typical customer.
3. Packages: Packages group related model elements of various types, including other packages.
4. Signals: Signals are model elements in UML diagrams that exist independently of the classifiers processing them. They define one-way asynchronous communication between active objects.
5. Enumeration: Enumerations are model elements in class diagrams that represent user-defined data types. Enumerations consist of named identifiers representing the values within the enumeration, which are called enumeration literals.

6. Data Types: Data types are model elements in UML diagrams that define data values. They are commonly used to represent primitive types (e.g., integers or strings) as well as enumerations and user-defined data types.
7. Artifacts: Artifacts in UML models represent physical entities within a software system. They serve as tangible units of implementation, such as executables, libraries, software components, documents, and databases.

Connections within class diagrams involve relationships between model elements. These relationships in UML enhance the model by defining the structure and behavior between elements.

Association qualifiers, in UML, are optional properties of binary associations. Qualifiers are part of association ends and contain a list of association attributes, each with a name and type. Association attributes model keys used to index subsets of relation instances.

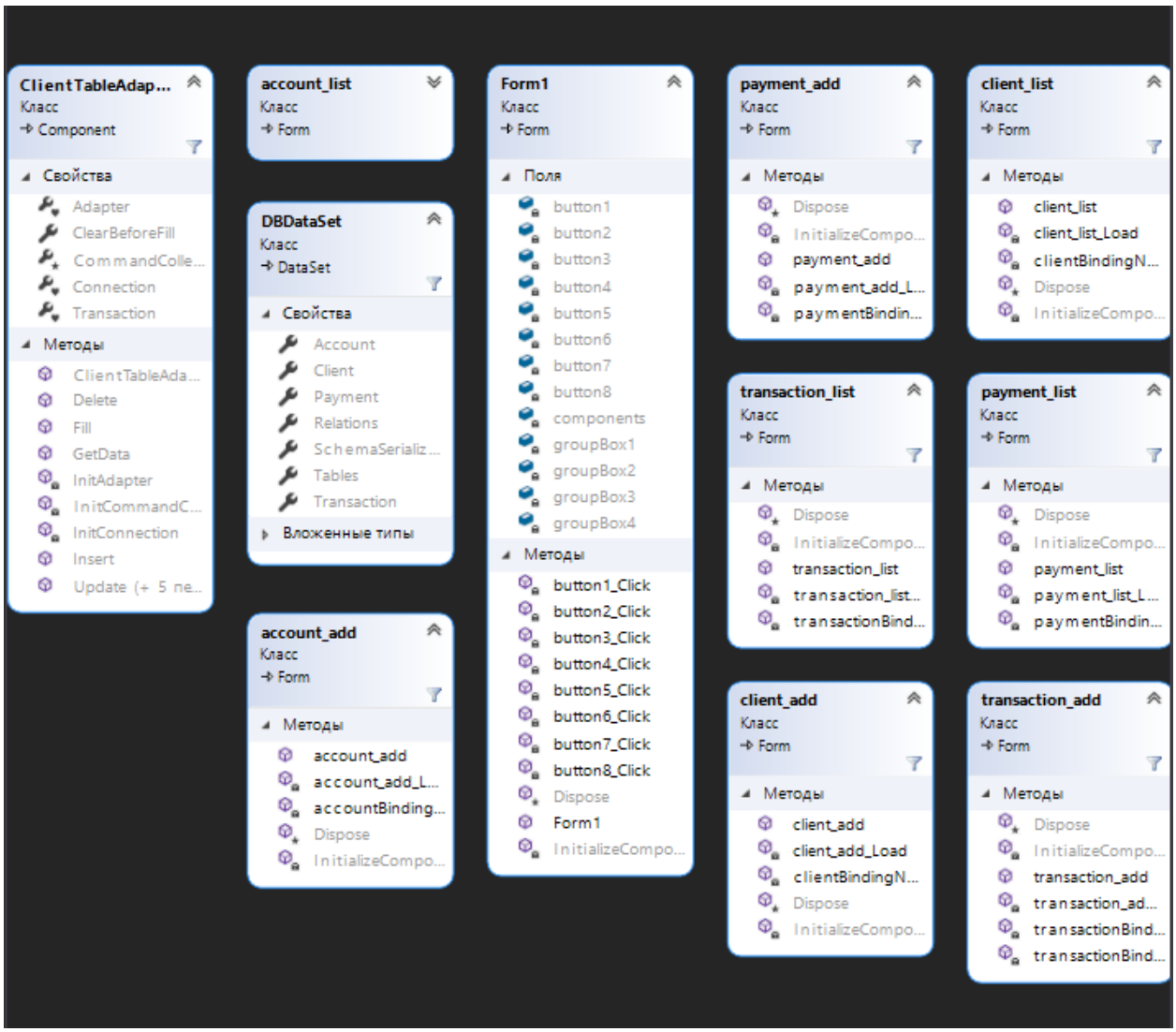


Figure 2.2 — Class diagram

## 2.2 Structure and features of implementation of algorithmic support

For a more detailed understanding of the methodology of the program, it is advisable to use the IDEF 0 diagram, which is shown in Figure 2.3.

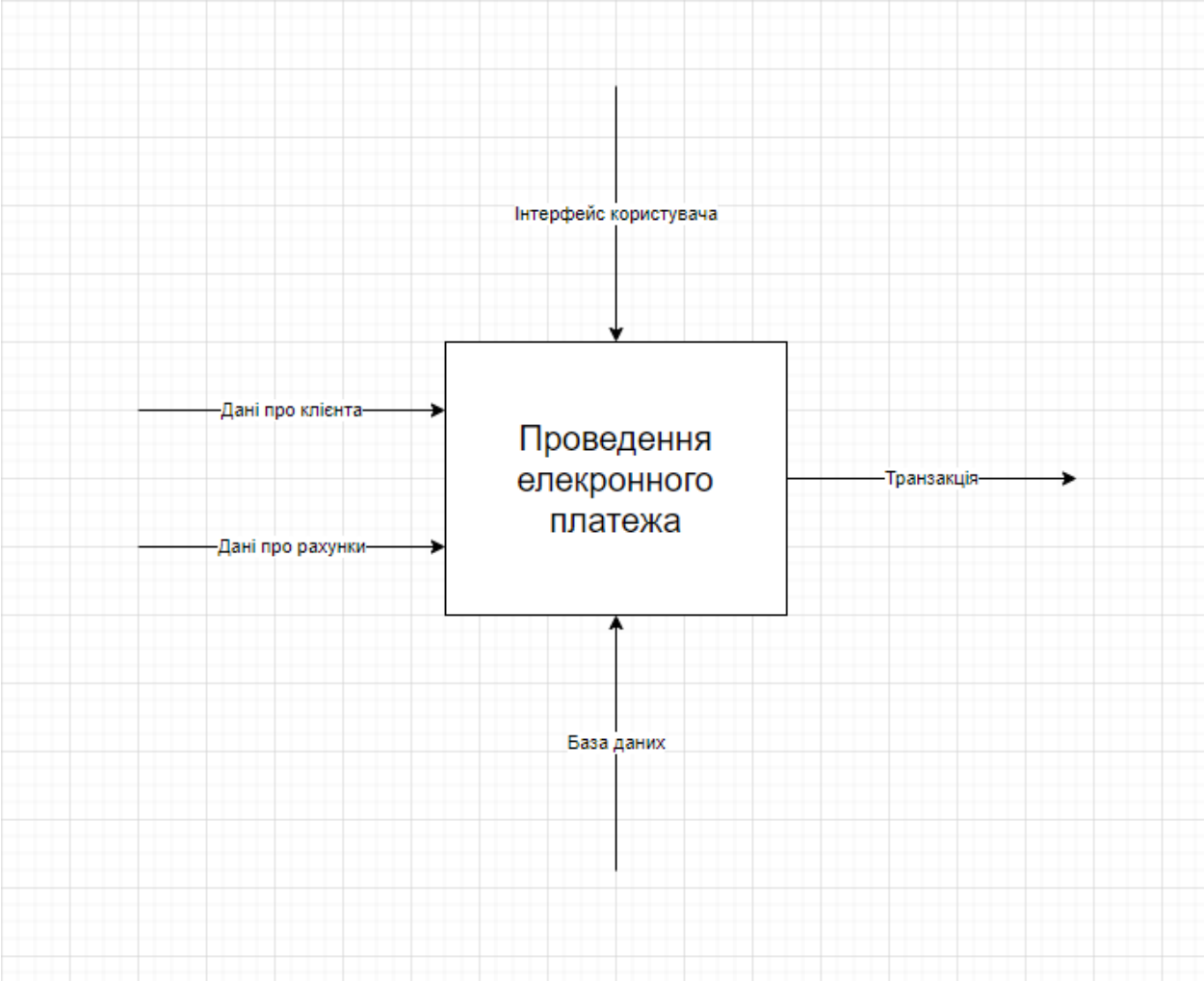


Figure 2.3 — IDEF 0 diagram

A component diagram (Figure 2.4) should be used to illustrate the structure of the software.

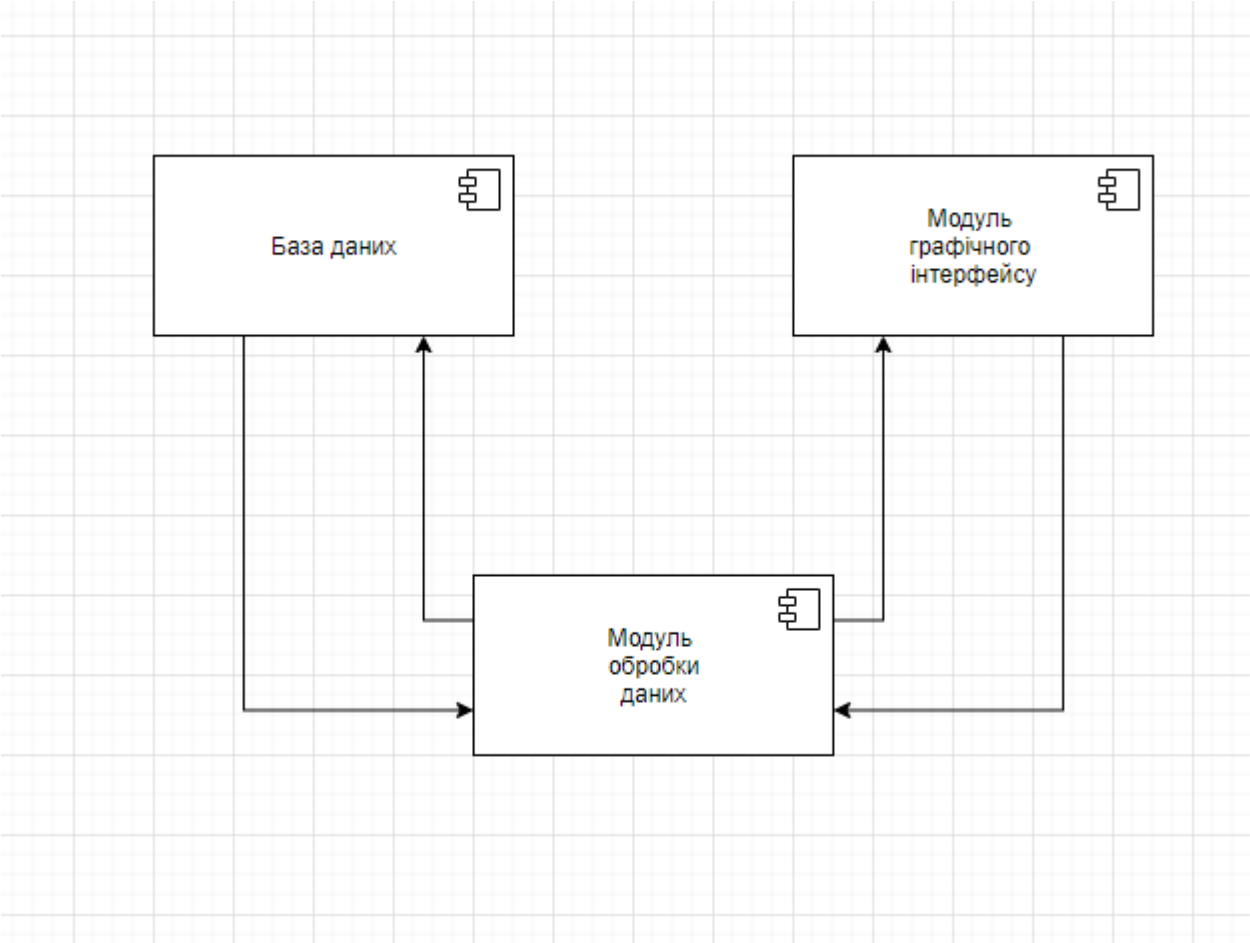


Figure 2.4 — Diagram of system components

To describe the technical support, it is advisable to use the deployment diagram shown in Figure 2.5.

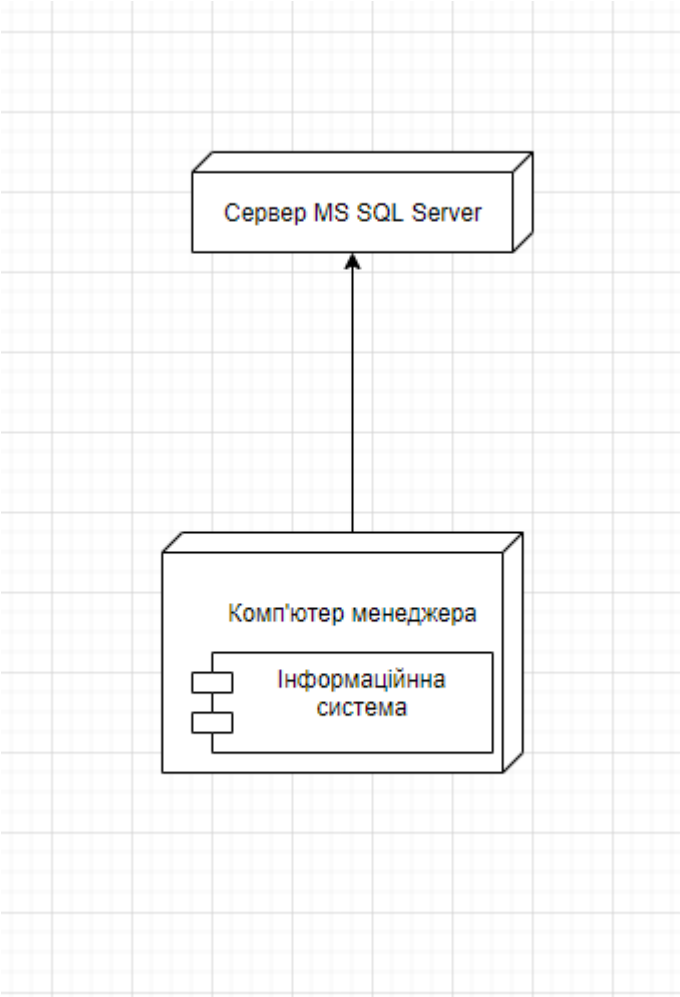


Figure 2.5 — Deployment diagram

**2.3 Test case and instructions for use**

A graphical user interface ( GUI ) is a complex system of visual components used in computer software. The GUI displays objects which _ transfer information and represent available actions for the user . When interacting user with these objects are happening changes in them color , size or visibility _

The first development graphic interface user was carried out in 1981 at Xerox PARC by Alan Kay , Douglas Engelbart and others researchers . Later , on January 19 , 1983, Apple introduced the computer Lisa with her own graphic interface user _

Graphic interface includes various _ graphic elements such as icons, cursors , and buttons. These elements sometimes additionally are strengthened sound effects or visual features , for example , transparency or shadows _ Thanks to hereby objects user may interact with a computer , optional taking off teams by heart

Below is an image of the Windows 7 desktop and an example operating system with a graphical user interface. In this example, you can use the mouse to move the pointer and click the program icon to launch the program.

What are the elements of a graphical user interface?

To make the GUI as user-friendly as possible, there are various elements and objects that the user uses to interact with the software. Below is a list of each with a brief description.

- A button is a graphical representation of a button that, when pressed, performs an action in the program
- A dialog box is a type of window that displays additional information and prompts the user for input.
- An icon is a small graphic image of a program, function, or file.
- Menu - a list of commands or options offered to the user through the menu bar.
- Menu bar – a thin horizontal bar containing menu labels.

- The Ribbon is a replacement for file menus and toolbars that group program actions.
- A tab is a clickable area at the top of a window that displays another page or area.
- A toolbar is a row of buttons, often at the top of an application window, that controls software functions.
- A window is a rectangular part of a computer display that shows the program currently in use.

How does the GUI work?

A GUI uses a variety of elements, such as windows, icons, and menus, to perform various commands, such as opening, deleting, or moving files. The main way to navigate a GUI operating system is with the mouse, but you can also use the keyboard by using keyboard shortcuts or arrow keys.

For example , to open programs in the system with graphics interface you need to point to the user mouse over the icon programs and twice to click her _ With use command line interface you, on the other hand , need to know the commands to go to the directory where it is located program , displaying a list of files and then running the file.

What are the advantages of GUI?

A graphical user interface is considered more user-friendly than a text-based command-line interface such as MS-DOS or the shell of Unix -like operating systems.

Unlike command-line operating systems such as Unix or MS - DOS , graphical user interface ( GUI ) operating systems have the advantage of being easier to learn and use. They do not demand memorization of commands and do not need knowledge programming . Thanks to my simplicity use and modern external view , operating rooms systems with graphics interface user have become dominant in the modern market.

How does the user interact with the GUI?

A pointing device, such as a mouse, is used to interact with almost all aspects of the GUI. More modern (and mobile) devices also use a touch screen.

The system includes a main menu from which you can access all parts of the system (Fig. 2.6).

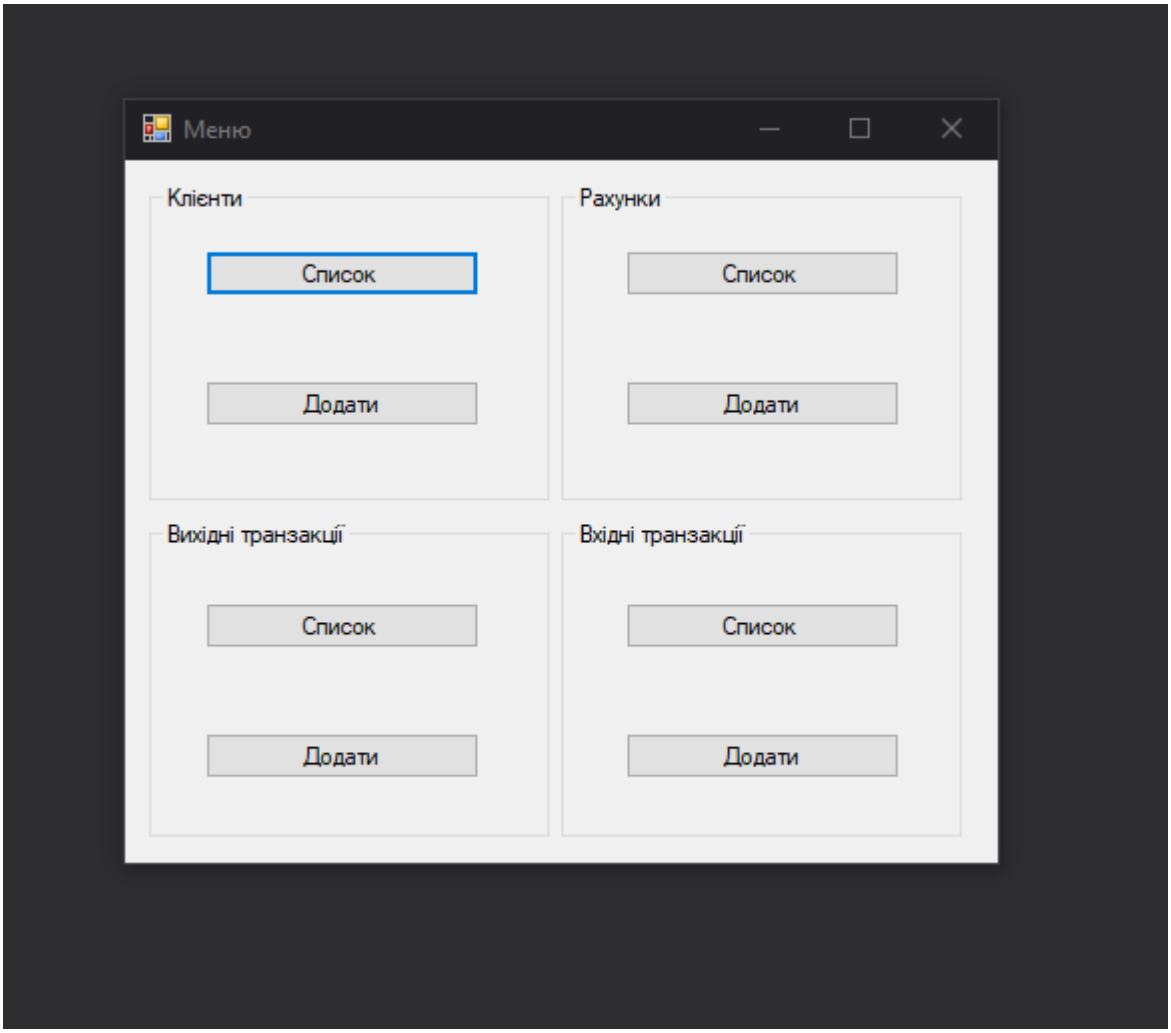


Figure 2.6 — Main menu

For each entity of the software system, there are 2 separate forms of the same type: the list form (Fig. 2.7) and the addition form (Fig. 2.8).

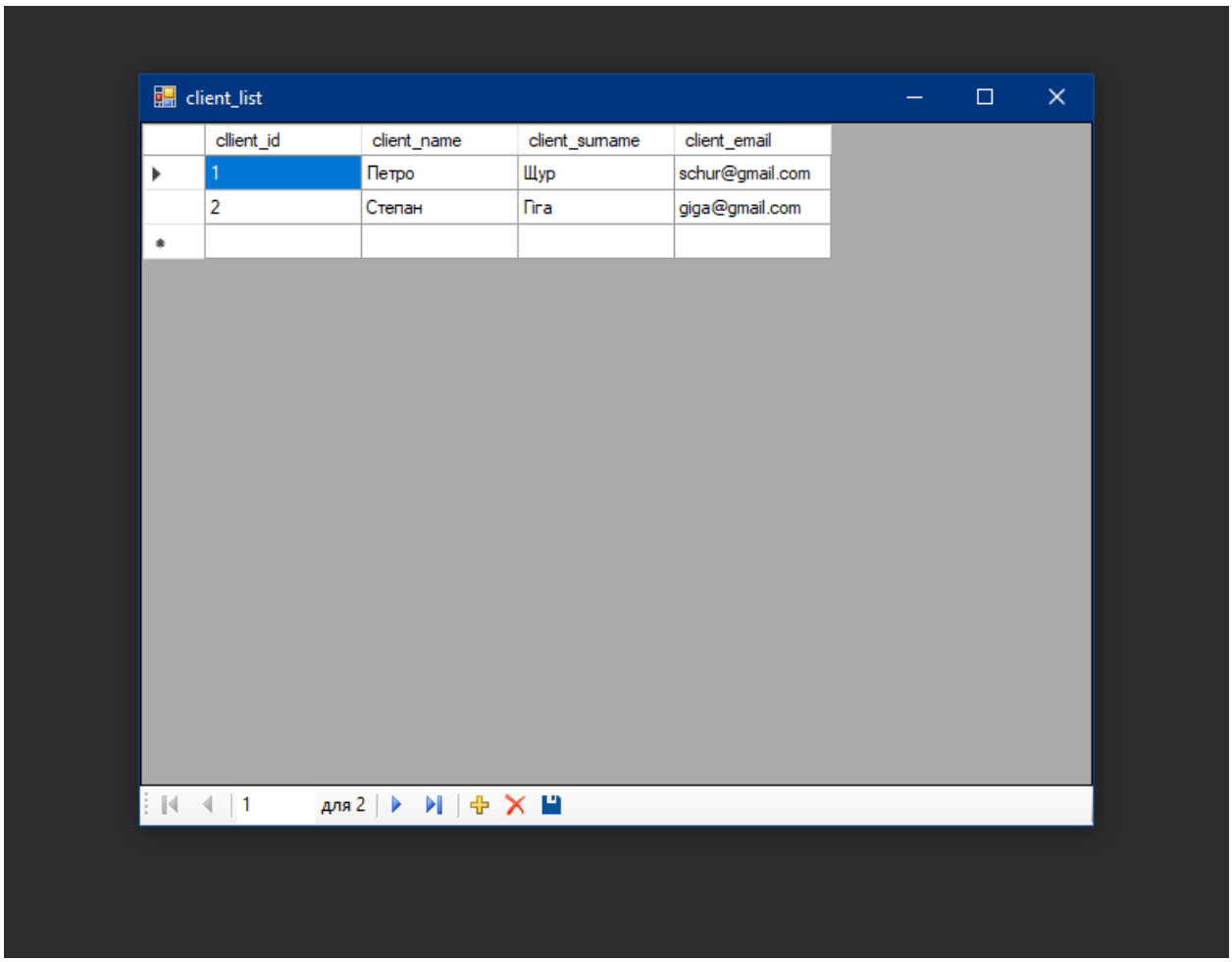


Figure 2.7 — List form

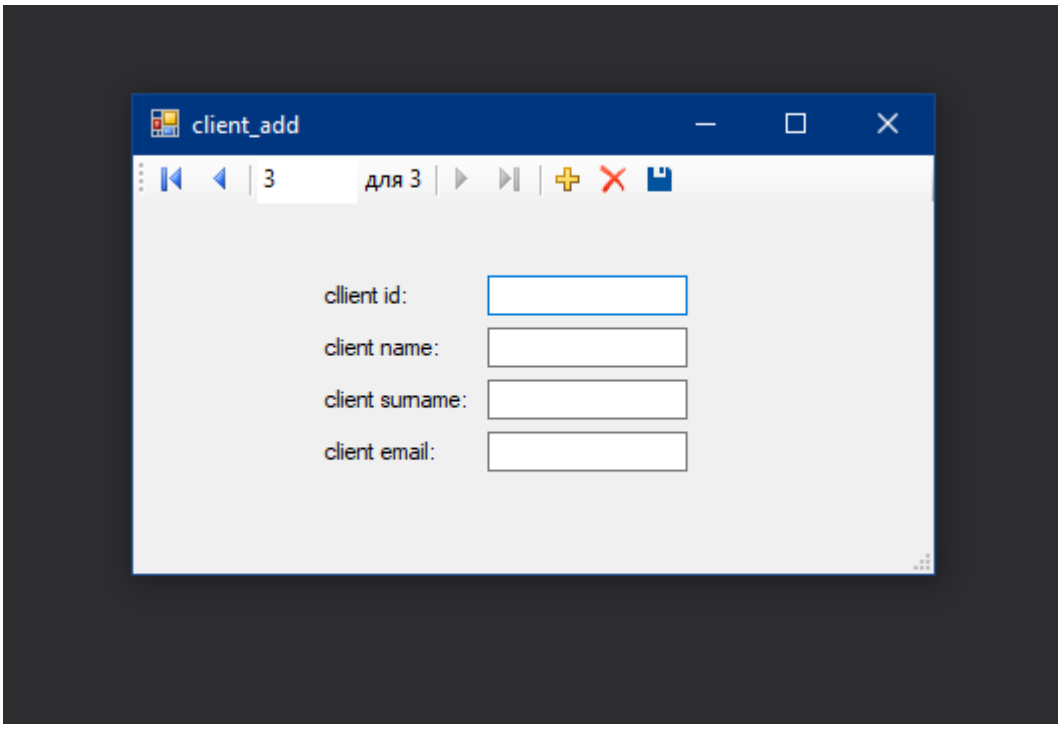


Figure 2.8 — Addition form

# CONCLUSIONS

The purpose of this work was to create a software implementation of a system for supporting electronic payments via the Internet.

In progress there was work on this project performed the following tasks :

- conduct an analysis of the subject area;
- analyze existing solutions;
- analyze the tools;
- design the system;
- develop an information structure;
- develop a class diagram;
- develop an interface and examples of work.

Thanks to the clear implementation of the tasks set at the beginning of the work, as a result of the work, a full-fledged system was obtained that is capable of performing the functionality laid in it and is ready for use in real conditions.

# LIST OF USED SOURCES

1. "PayPal Leadership". www.paypal.com. Retrieved April 7, 2019.

2. "PayPal reports Fourth Quarter and Full Year 2019 Results", investor.paypal-corp.com, January 29, 2020.

3. "Archived copy". Archived from the original he July 4, 2019. Retrieved April 29, 2019.

4. "PayPal Goes Public Again". Forbes . Retrieved August 27, 2017.

5. "EX-10.5". www.sec.gov. Retrieved June 16, 2018.

6. Mack , Ryan ( May 1, 2012). " Reid Hoffman And Peter Thiel In Conversation : Finding The Best Candidates For The Job ". Forbes . Retrieved February 28, 2018.

7. Wolverton , Troy ( October 3, 2002). " It's official : eBay weds PayPal ". CNET News . Retrieved March 16, 2014.

8. Richtel , Matt ( July 9, 2002). " eBay that Buy PayPal , a Rival in Online Payments ". The New York Times . Retrieved November 23, 2014.

9. Seetharaman , Deepa ; Mukherjee , Supantha ( September 30, 2014). " eBay follows Icahn's advice , plans PayPal spinoff in 2015". Reuters . Retrieved April 16, 2021.

10." Fortune 500 Companies 2021". fortune.com. Retrieved August 23, 2021.

11.Forrest , Conner ( June 30, 2014). " How the ' PayPal Mafia ' redefined success in Silicon Valley ". TechRepublic . Retrieved November 11, 2014.

12.Cohan , Peter S. ( September 5, 2013). " What PayPal's Rocky Beginnings Can Teach You About Startup Success ". Entrepreneur Magazine . Retrieved November 11, 2014.

13." History of PayPal Inc. _ – FundingUniverse ".

14.Penenberg , Adam L. ( August 9, 2012). " Reid Hoffman He PayPal's Pivoted Path That Success ". Fast Company . Retrieved November 11, 2014.

15.Jackson , Eric M. (2006). The PayPal Wars : Battles with eBay , the Media , the Mafia and the Rest of Planet Earth . WND Books . ISBN 978-0-9778984-3-5.

16.Gardner , James A. ( April 20, 2011). Innovation and the Future Proof Bank : A Practical Guide that Doing Different Business-as-usual . John Wiley & Sons . ISBN 978-0-470-68521-1.

17.Kidder , David ; Hoffman , Reid ; Hindi , Hanny (2012). The Startup Playbook : Secrets of the Fastest Growing Start-Ups from the founding Entrepreneurs . San Francisco , CA: Chronicle Books . pp . 214–216. ISBN 9781452105048.

18.Locke , Taylor ( November 2, 2019). " Why Elon Musk says taking vacations _ will kill you '". CNBC. Retrieved March 31, 2020.

19.Rosoff , Matt ( November 12, 2011). " Where Are They Now ? The PayPal " Mafia " Is More Powerful Than Ever ". Business Insider . Retrieved November 11, 2014.

20.SEETHARAMAN, DEEPA; MUKHERJEE, SUPANTHA ( September 30, 2014). " eBay follows Icahn's advice , plans PayPal spinoff in 2015". Reuters . Retrieved November 11, 2014.

21.Statt , Nick ( July 10, 2017). Elon _ Musk now owns X.com, the defunct domain of his second startup ". The Verge . Retrieved March 31, 2020.

22.Keith Regan ( February 15, 2002). "PayPal IPO Off that Spectacular Start". ECommerce Times . Retrieved November 11, 2014.

23." eBay 2002 Annual Report " (PDF).

24.Kane , Margaret ( July 8, 2002). eBay _ picks up PayPal for $1.5 billion ". CNET News.com. CNET Networks . Retrieved November 13, 2007.

25.Austin Carr ( October 16, 2014). "UNFINISHED BUSINESS: MAX LEVCHIN'S QUEST TO BUILD THE NEXT PAYPAL". Fast Company . Retrieved November 11, 2014.

26." How eBay's purchase of PayPal changed Silicon Valley ". Venturebeat . Venture Beat . October 27, 2012. Retrieved October 28, 2012.

27.Keith Regan ( October 11, 2005). " Eyeing Expansion of PayPal , eBay Buys VeriSign Payment Gateway ". ECommerce Times . Retrieved November 11, 2014.

# APPENDIXES

## Appendix A
## ABSTRACT OF QUALIFICATION WORK

### Summary

Nahornyi Mykola Vitaliyovych "Development of an Information System Supporting Electronic Payments" - Bachelor's qualifying thesis. Sumy State University, Sumy, 2023.

The main goal of the work is to create a software implementation of a system for supporting electronic payments via the Internet.

The work reflects the research process, the choice of technology, the development algorithm for achieving a practical result.

The main scientific results of the bachelor's work are a developed project that allows you to get a convenient application that allows you to quickly perform banking operations.

Keywords: C#, visual studio, generation, database, .NET, development optimization, CLI, client-server, UML, GUI, database.

### АНОТАЦІЯ

Нагорний Микола Віталійович «Розробка інформаційної системи підтримки електронних платежів» - кваліфікаційна робота бакалавра. СумДУ, Суми, 2023.

Основною метою роботи є створення програмної реалізації системи підтримки електронних платежів через Інтернет.

У роботі відображено процес дослідження, вибір технології, алгоритм розробки для досягнення практичного результату.

Основними науковими результатами бакалаврської роботи є розроблений проект, який дозволяє отримати зручний додаток, що дозволяє швидко здійснювати банківські операції.

Ключові слова: C#, візуальна студія, генерація, база даних, .NET, оптимізація розробки, CLI, клієнт-сервер, UML, GUI, база даних.

**Appendix B**

**(informational)**

```
/// <summary>
///Represents a strongly typed in-memory cache of data.
///</summary>
[global::System.Serializable()]
[global::System.ComponentModel.DesignerCategoryAttribute("code")]
[global::System.ComponentModel.ToolboxItem(true)]
[global::System.Xml.Serialization.XmlSchemaProviderAttribute("GetTypedDataSetSchema")]
[global::System.Xml.Serialization.XmlRootAttribute("DBDataSet")]
 [global::System.ComponentModel.Design.HelpKeywordAttribute("vs.data.DataSet")]
public partial class DBDataSet : global::System.Data.DataSet {

    private AccountDataTable tableAccount;

    private ClientDataTable tableClient;

    private PaymentDataTable tablePayment;

    private TransactionDataTable tableTransaction;

    private global::System.Data.DataRelation relationFK_Account_Client;

            private global::System.Data.DataRelation relationFK_Transaction_ToAccountTo;

            private global::System.Data.DataRelation relationFK_Transaction_ToClientFrom;
```

```
            private global::System.Data.DataRelation
relationFK_Transaction_ToAccountFrom;

            private global::System.Data.DataRelation
relationFK_Transaction_ToClientTo;

            private global::System.Data.SchemaSerializationMode
_schemaSerializationMode =
global::System.Data.SchemaSerializationMode.IncludeSchema;

        [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
        [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Dat
a.Design.TypedDataSetGenerator", "16.0.0.0")]
        public DBDataSet() {
            this.BeginInit();
            this.InitClass();
                global::System.ComponentModel.CollectionChangeEventHandler
schemaChangedHandler = new
global::System.ComponentModel.CollectionChangeEventHandler(this.Schema
Changed);
            base.Tables.CollectionChanged += schemaChangedHandler;
            base.Relations.CollectionChanged += schemaChangedHandler;
            this.EndInit();
        }
```