

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Проектування системи ідентифікації клієнтів банку при проведенні
процедур фінансового моніторингу

Виконав студент групи К-41
спеціальність 122 Комп'ютерні науки
Ходжагулиєв Тіркіш

Керівник д. т. н., професор каф. АСМНСІ
Великодний Станіслав Сергійович

Консультант _____

Рецензент д. ф.-м. н., доцент каф. ВПМ
Буяджи Василь Володимирович

Одеса 2022

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	6
1 Аналіз вихідних даних і вимог завдання на кваліфікаційну роботу.....	8
1.1 Аналіз вихідних даних.....	8
1.2 Аналіз заходів фінансового моніторингу вкладів та операцій у банках .	9
2 Розробка структурної схеми додатка	13
2.1 Розробка заходів з автоматизації фінансового моніторингу у банку ...	13
2.2 Розробка структурної схеми	19
3 Розробка системи управління базою даних.....	25
3.1 Вибір платформи проектування	25
3.2 Розробка структури таблиць	32
4 Розробка програмного засоби	37
4.1 Розробка SQL-запитів до обробки таблиць БД.....	37
4.2 Розробка інтерфейсу програмного засобу	39
4.3 Розробка підсистеми інформаційного забезпечення програми.....	42
4.4 Аналіз вхідних даних за допомогою автоматизованої системи	46
4.5 Тестування програмного забезпечення.....	48
Висновки	51
Перелік джерел посилання	53

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних.

ЄДРПОУ – єдиний державний реєстр підприємств та організацій України;
їни;

ІАС – інформаційно-аналітична система.

ІКБ – ідентифікація клієнта банку.

НБУ – Національний банк України.

ОС – операційна система.

САПР – система автоматизованого проектування.

СУБД – система управління базою даних.

ТЗ – технічне завдання.

CASE – Computer-Aided Software Engineering.

SQL – Structured Query Language.

ВСТУП

Сьогодні однією з найбільш актуальних тем, що обговорюються, є впровадження, оптимізація та реалізація (здійснення) системи фінансового моніторингу у всіх галузях фінансової системи України. Основним нормативним документом є Положення щодо здійснення банками фінансового моніторингу Правління Національного банку України (НБУ) – постанова №189 [1] від 14.05.2003 р.

Під ідентифікацією фізичної особи розуміються дії працівників банку, створені задля встановлення власності документів тій особі, які ця особа пред'являє на посвідчення своєї особи і своїх повноважень під час встановлення клієнтських відносин із банком. Пред'явлені документи повинні бути при цьому з розумною ретельністю перевірені щодо їх дійсності та наявності всіх необхідних реквізитів. Вимога ідентифікації клієнта виключає законну можливість відкриття банківського рахунку та здійснення банківських операцій після пред'явлення ксерокопії документа, документа на чуже ім'я або документів із явними ознаками підробки.

Якщо клієнтом є юридична особа, то завдання співробітників банку значно ускладнюється. Ідентифікація юридичної особи полягає у перевірці та оцінці документів, поданих на посвідчення її правового статусу та підтверджуючих її державну реєстрацію, у встановленні власників цього підприємства, а також у діях, спрямованих на отримання достовірної інформації з мети та передбачуваному характері ділових відносин.

Ідентифікація клієнта як найважливіша складова фінансового моніторингу – це складне і дуже трудомістке завдання, яке вимагає не тільки знання правової бази, а й певного досвіду та інтуїції. При постійно зростаючих об'єктах та умовах жорсткої конкуренції необхідно робити таку роботу швидко і з мінімальною ймовірністю помилки. І найбільш перспективний метод вирішення такого завдання – автоматизація деяких видів робіт з ідентифікації клієнта та моніторингу руху його коштів на рахунках.

Тому запропонована тема кваліфікаційної роботи бакалавра має актуальний характер.

Мета кваліфікаційної роботи – спроектувати систему аналізу та ідентифікації клієнта банку при виконанні ним фінансових операцій.

Об'єкт роботи – система «банк-клієнт».

1 АНАЛІЗ ВИХІДНИХ ДАНИХ І ВИМОГ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз вихідних даних

У технічному завданні (ТЗ) на кваліфікаційну роботу поставлено завдання розробити, заходи та програмне забезпечення, покликане забезпечити автоматизацію моніторингу клієнтів банку з погляду відмивання доходів, отриманих незаконним шляхом та можливого фінансування терористичної діяльності.

Під завданням проектування є розробка програмного засобу, призначеного для автоматизації ідентифікації клієнта банку.

При цьому додаток, що розробляється в рамках проекту, має бути виконаний у вигляді WEB-додатка з використанням однієї з сучасних технологій програмування. Яка платформа буде використовуватися при цьому, в завданні не вказано. Також у завданні не вказано платформу бази даних – БД (SQL сервер), який буде використовуватись для зберігання даних про клієнта. Тому це питання залишається на розсуд проектувальника.

Додатковим обмеженням на розробку є те, що моніторинг рахунків здійснюється в рамках окремого підрозділу банку. Це означає, що заходи з автоматизації фінансового моніторингу, що розробляються, матимуть локальне використання без виходу в загальну мережу банку. Це спрощує процедуру поділу прав доступу та практично не вимагає використання додаткових спеціальних заходів безпеки програмного забезпечення.

Як керівна інформація виступає нормативна база України та постанови Національного банку України в частині фінансового моніторингу та забезпечення його роботи.

При цьому програма виконується за технологією «Клієнт-сервер». Використання технології клієнт-сервер обумовлено тим, що використання даного програмного засобу можливе декількома співробітниками банку, які пра-

цюють у рамках відокремленого підрозділу з фінансового моніторингу. При цьому вони використовуватимуть спільне сховище даних.

Робота такого підрозділу нерозривно пов'язана з іншими службами банку, тому відділ фінансового моніторингу повинен буде використовувати інформацію, що знаходиться в обороті всередині банку, наприклад, інформацію про платежі до їх завершення.

1.2 Аналіз заходів фінансового моніторингу вкладів та операцій у банках

Фінансовий моніторинг – це поняття, схоже на поняття фінансового контролю, а також безпосередньо пов'язане з обов'язковими процедурами внутрішнього контролю в частині проведення фінансових операцій. Банк організує внутрішній контроль з урахуванням рекомендацій щодо організації внутрішнього контролю за ризиками банківської діяльності.

Система внутрішнього контролю організується органами управління банку, уповноваженими установчими документами банку. З метою моніторингу за процесом функціонування системи внутрішнього контролю, виявлення та аналізу проблем, пов'язаних з нею, а також розробки пропозицій щодо вдосконалення та підвищення ефективності її діяльності в банках створюється служба внутрішнього контролю. Важливо також, що до одного із заходів протидії легалізації (відмиванню) коштів та доходів, отриманих злочинним шляхом, відносять заборону на інформування клієнтів та інших осіб про заходи, що вживаються відповідно до законодавства України.

На рис. 1.1 наведено основні ознаки легалізації доходів, одержаних незаконним шляхом [1].

Ця схема досить наочно показує, у разі слід особливо уважним і під час функцій ідентифікації клієнта банку.

На рис. 1.2 показано основні рекомендовані заходи щодо протидії відмиванню доходів, отриманих незаконним шляхом.

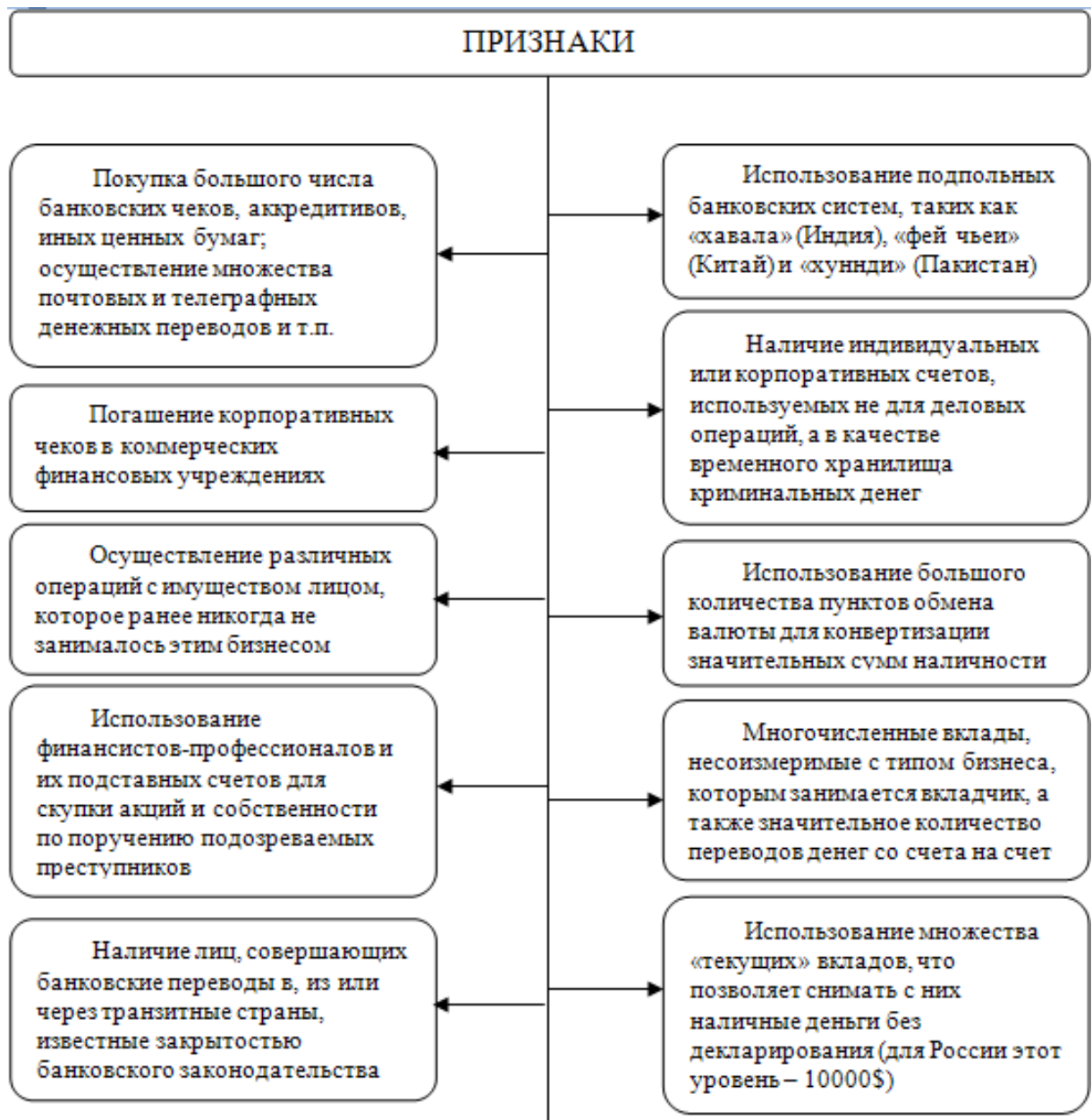


Рисунок 1.1 – Ознаки реалізації доходів

Наведена на рис. 1.1 та 1.2 інформація узагальнює існуючий світовий досвід протидії різним видам незаконної діяльності з відмивання доходів та фінансування терористичної діяльності.

Для того, щоб перейти до виконання безпосередньо проектних робіт, необхідно розглянути заходи щодо здійснення фінансового моніторингу стосовно українського законодавства.

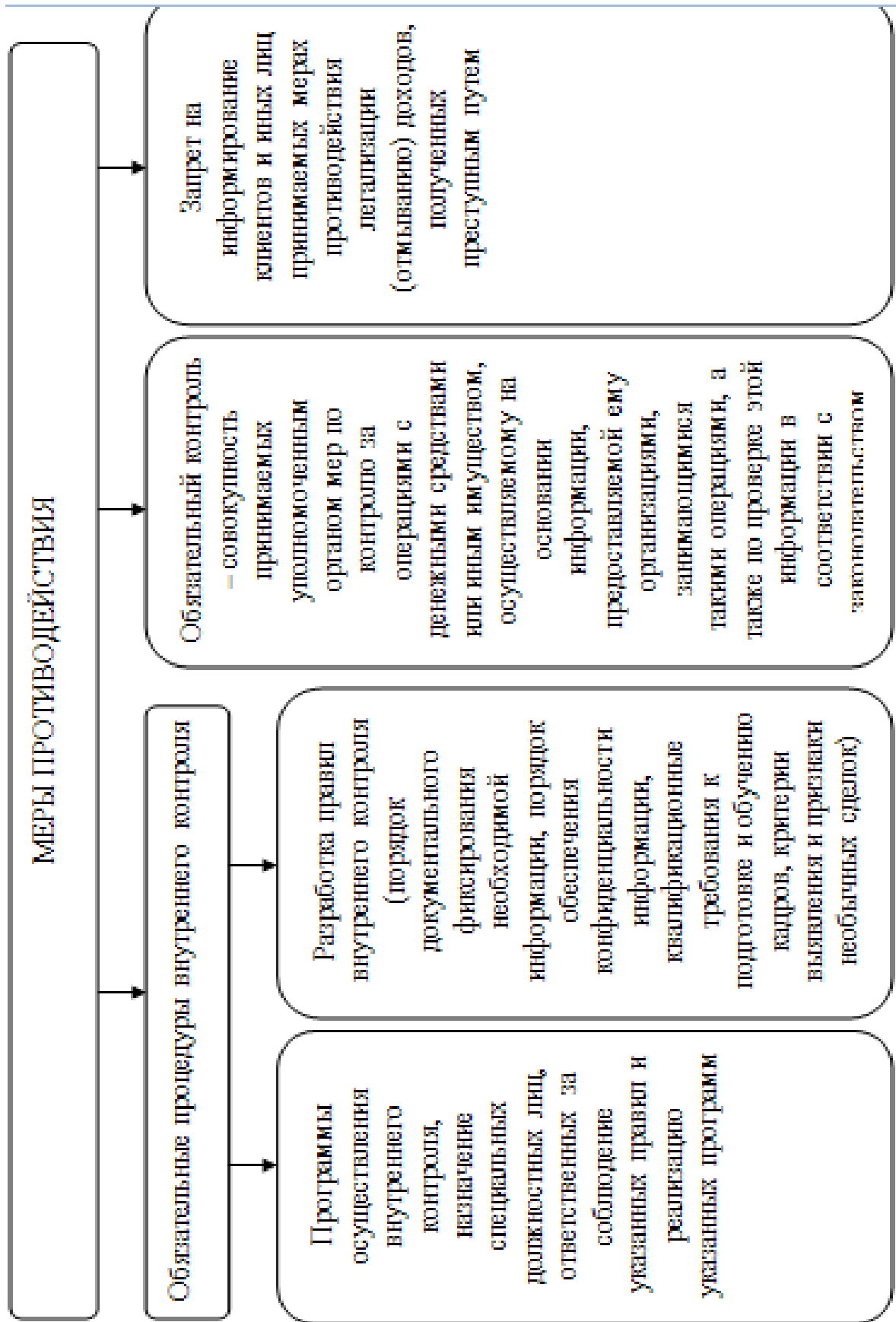


Рисунок 1.2 – Заходи протидії легалізації (відмиванню) доходів, отриманих злочинним шляхом

Діяльність банку щодо фінансового моніторингу регламентується насамперед, Положенням НБУ про здійснення банками фінансового моніторингу [1].

Це Положення передбачає розробку банками власних Правил здійснення фінансового моніторингу на підставі вимог НБУ.

Складовою частиною таких правил є Програма здійснення фінансового моніторингу, яка містить у загальному випадку:

- а) ознаки операцій, що підлягають фінансовому моніторингу;
- б) ознаки операцій, що підлягають внутрішньому фінансовому моніторингу;
- в) порядок виявлення фінансових операцій, які:
 - 1) підлягають фінансовому моніторингу;
 - 2) підлягають внутрішньому фінансовому моніторингу;
 - 3) можуть бути пов'язані, стосуватися або призначатися для фінансування тероризму;
- г) порядок проведення заходів щодо з'ясування (уточнення) суті та цілей операції клієнта, що підлягає фінансовому моніторингу;
- д) порядок підготовки та подання відповідному державному органу інформації про такі операції.

У разі виявлення операції, що підлягає фінансовому моніторингу, відділ фінансового моніторингу має забезпечити:

- а) реєстрацію такої операції;
- б) вжити заходів щодо з'ясування сутності та цілей такої операції;
- в) вжити належних заходів щодо запобігання виконання такої операції;
- г) за наявності підстав – надати відповідну інформацію контролюючого органу.

У межах цього дипломного проекту необхідно розробити власні заходи задля досягнення поставленої мети.

Для цього керуватимемося інформацією, отриманою з джерел інших держав [2] – [4].

2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ДОДАТКА

2.1 Розробка заходів з автоматизації фінансового моніторингу у банку

При розробці таких заходів використовуємо лише відкриті джерела інформації та керуємося:

- а) вимогами українського законодавства у галузі фінансового моніторингу;
- б) світовим досвідом у цій галузі.

Насамперед необхідно систематизувати операції, ознаки яких свідчать про можливу неправову діяльність клієнта банку. Саме на підставі інформації про такі операції ми зможемо виробити заходи з ідентифікації клієнта на основі характеру операцій, які він проводить.

Розглянемо ще раз і детальніше основні ознаки операцій, що підлягають фінансовому моніторингу. Ці операції можна розвинути на дві групи:

- а) операції превентивного характеру - з метою запобігання можливим фінансовим порушенням клієнтами, у надійності яких існують сумніви;
- б) операції, сам склад яких є сумнівним, потребує більш ретельного контролю.

Дані відомостей представлені в табл. 2.1.

У цій таблиці наведено найменування даних параметрів пошуку, які надалі використовуватимуться розробки програмного засобу.

На рис. 2.1 наведено діаграму, яка візуалізує взаємодію розглянутих даних з погляду джерел інформації про клієнтів.

Операції, склад яких є сумнівним, потребує більш ретельного контролю, також може бути об'єктом автоматизованого моніторингу, проведеного банком щодо клієнта.

Таблиця 2.1 – Анкетні дані клієнта, що підлягають автоматизації фінансового моніторингу

№ з/п	Позначення	Найменування	Спосіб виявлення
1	cl_name	Найменування суб'єкта господарювання	Чорний список юридичних осіб
2	promo	Засновник	Чорний список фізичних осіб, чорний список юридичних осіб
3	authorised	Уповноважена особа	Чорний список фізичних осіб
4	risk	Рівень ризику	Поле анкети (заповнюється експертом)
5	auth_capit	Статутний фонд	Чисельне значення поля анкети, що заповнюється вручну з урахуванням установчих документів. Контролюється перевищення обсягу операцій за певний період
6	contrag	Основні контрагенти	Чорний список фізичних осіб, чорний список юридичних осіб

Ознаки такого роду операцій розглянуті в літературі [4] і наведені на рис. 1.1. Однак, у такому вигляді ці відомості непридатні для вирішення завдання автоматизації фінансового моніторингу щодо ідентифікації клієнтів банку. Наше завдання – трансформувати наявні дані для того, щоб їх було зручно використати.

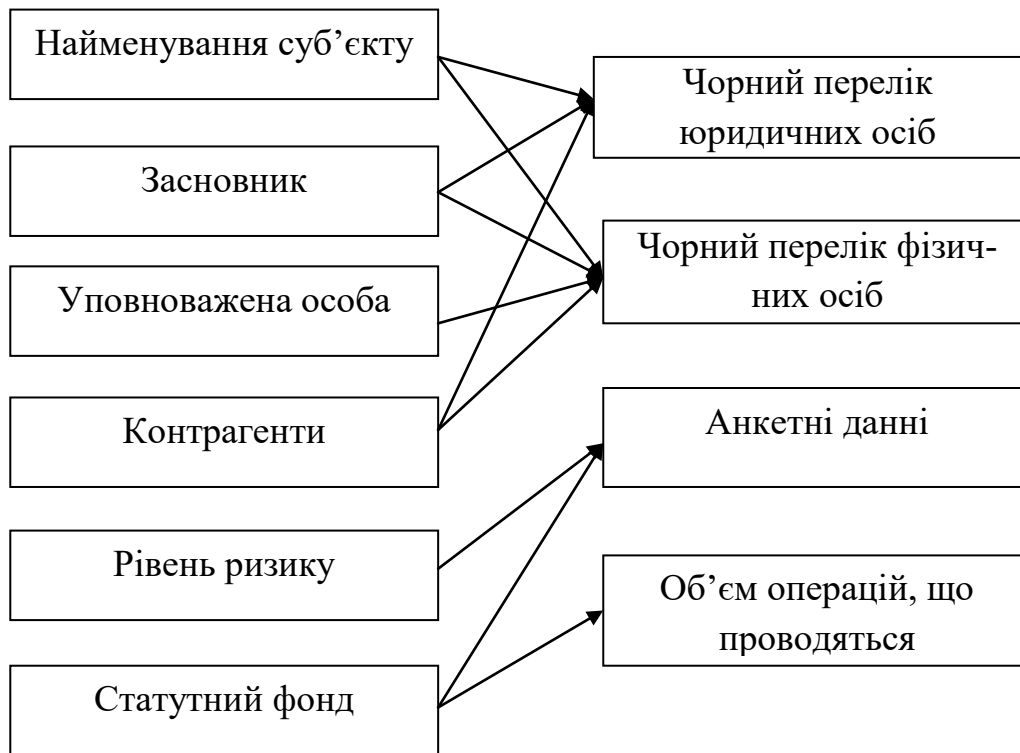


Рисунок 2.1 – Схема автоматизованого контролю анкетних даних клієнта

Насамперед – це операція, регламентована Положенням НБУ про здійснення банками фінансового моніторингу. Ця ознака свідчить, що обов'язковому моніторингу підлягають операції, що виконуються на суму понад 280 тис. грн. без відкриття рахунку у банку. Фактично це виконання операцій з використанням готівкових коштів.

Така операція може бути виявлена шляхом контролю двох параметрів кожного платежу:

- а) джерело платежу – готівковий розрахунок;
- б) сума платежу – понад 280 тис. грн. або в іноземній валюті у тому ж еквіваленті.

Ці відомості можна досить впевнено контролювати за допомогою сканування всіх платежів у базі даних банку, що підлягають проведенню.

При виявленні відповідних платежів змінюється їх статус (наприклад, на «зупинений») з наступним моніторингом відповідальними працівниками банку (у нашому випадку – працівниками відділу фінансового моніторингу).

Розглянемо такі параметри даного контролю:

- а) тип операції – будь-яка, крім безготівкового платежу;
- б) контрольований період – календарний місяць;
- в) обсяг операцій за контрольований період – 25% від статутного фонду.

Ці цифри можуть бути іншими. Очевидно, що конкретні значення повинні бути секретом, оскільки інакше нечистим на руку клієнтам буде легше уникнути небажаного для них підвищення уваги з боку служби фінансового моніторингу банку.

Табл. 2.2 резюмує результати нашого дослідження, а рис. 2.2 показує схему взаємодії під час виконання автоматизованого моніторингу.

Таблиця 2.2 – Операції клієнта, що підлягають автоматизації фінансового моніторингу

№ п/п	Позначення	Найменування	Засіб виявлення
1	2	3	4
1	pay_src	Джерело платежу на суму понад 280 тис. грн.	Джерело платежу – не банківський рахунок, сума платежу – понад 280 тис. грн.
2	pay_purp	Призначення платежу	Звіряє призначення платежу з видами діяльності клієнта. Якщо платіж призначено оплати електроенергії, комунальних послуг, сплати податків та інших обов'язкових платежів, то контроль не проводиться

Кінець табл. 2.2

1	2	3	4
3	pay_tot	Загальний обсяг фінансових операцій за контрольований період	Якщо сума перевищує розмір статутного фонду (auth_capit), то моніторинг необхідний
4	pay_tgt_name	Назва отримувача	Зіставлення з чорним списком
5	pay_tgt_ac	Рахунок отримувача	Якщо гроші перераховуються 1 не на рахунок (наприклад, грошові поштові перекази), то це об'єкт моніторингу, якщо загальна сума більше 25% від статутного фонду

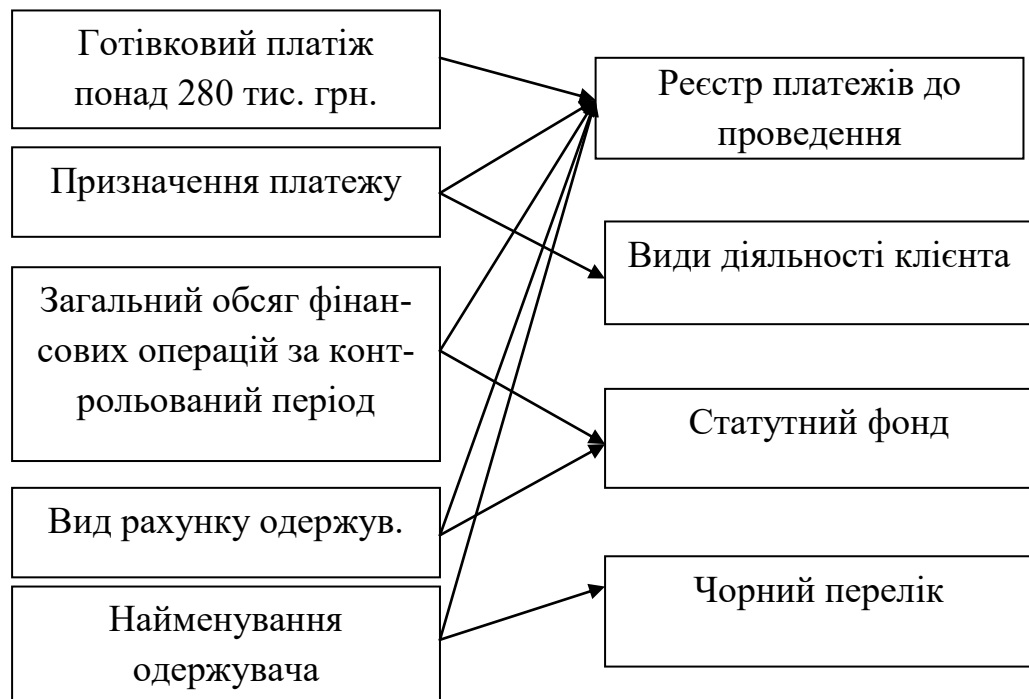


Рисунок 2.2 – Оперативна інформація, що використовується для фінансового моніторингу клієнта банку

Особливістю даної схеми і те, що у ролі вихідної інформації у разі використовується журнал (реєстр) платежів, які підлягають проведенню.

Іншими словами, при виконанні банківських платежів служба фінансового моніторингу повинна "вклинитися" у проміжок часу, коли здійснюється проведення платежів, до того моменту, коли кошти вже будуть перераховані на рахунок отримувача. Залежно від організації банку, така операція може проводитись із накопиченими платежами, наприклад, кожні 2 години. Цей час може відрізнятися і є предметом розгляду нашому проекте. Важливо лише – як часто запускатимуть нашу програму в банку. Суть роботи цієї програми від цього не зміниться.

Таким чином, можна зробити невеликий висновок – ми класифікували інформацію та її місцезнаходження щодо можливості автоматизації фінансового моніторингу в банку з метою запобігання легалізації доходів, отриманих злочинним шляхом, та з метою фінансування можливої терористичної діяльності.

Тут найважливішим є:

а) додаток, що розробляється, повинен використовувати деякі компоненти внутрішньобанківської інформаційної системи, зокрема, анкетні дані клієнта та журнал платежів протягом операційного дня;

б) програма запускається в період перед черговим проведенням зазначених платежів;

в) у випадку, якщо та чи інша операція підлягає фінансовому моніторингу, програма, що розробляється, повинна формувати власний тимчасовий реєстр, що є вихідною інформацією для служби фінансового моніторингу, що веде остаточний аналіз;

г) програма, що розробляється, покликана інформувати службу фінансового моніторингу, але не виконує функцію прийняття рішення - ця функція виконується відповідальним працівником банку. Він же формує звітність, що надається до контролюючих та правоохоронних органів. Складання остаточного звіту в завдання програми, що розробляється, не входить.

2.2 Розробка структурної схеми

Розроблені заходи з автоматизації фінансового моніторингу є основою розробки структурної схеми докладання.

Для розробки структурної схеми необхідно сформулювати:

- а) найменування програми;
- б) мета розробки програми;
- в) завдання, що вирішуються цим додатком;
- г) функції, виконання яких необхідне для вирішення завдань програми;
- д) перелік вхідної та вихідної інформації даної програми, перелік вхідної та вихідної інформації для кожної з функцій.

Надалі вважатимемо, що програма матиме модульну структуру, кожен модуль відповідатиме виконуваній функції.

Найменування програми – ІКБ (Ідентифікація Клієнта Банку).

Мета розробки програми – зниження трудомісткості та прискорення робіт з ідентифікацій клієнта на предмет можливих дій щодо легалізації доходів, отриманих злочинним шляхом та (або) фінансування терористичної діяльності.

Завдання, які вирішуються додатком:

- а) автоматизація обробки інформації під час відкриття рахунку клієнтом;
- б) контроль за виконанням разових операцій, що підлягають фінансовому моніторингу;
- в) контроль виконання повторюваних¹ операцій, що підлягають фінансовому моніторингу;
- г) формування внутрішнього звіту з виконаної роботи, зупинення операцій, що підлягають фінансовому моніторингу.

Кожне із завдань має бути розглянуте уважніше з метою формування списку функцій, що виконуються додатком.

Автоматизація обробки інформації при відкритті рахунку клієнтом дозволить на етапі відкриття рахунку, використовуючи анкетні дані клієнта, забезпечити підтримку прийняття рішення відповідальним працівником банку про відкриття рахунку клієнта або відмову в такому відкритті [5], [6]. Насамперед, це завдання має на меті позбавити відповідального працівника банку від тривалої та стомлюючої процедури звіряння анкетних даних клієнта з різними "чорними списками". Такими списками є:

а) список неблагонадійних юридичних осіб у частині перевірки:

- 1) суб'єктів господарювання;
- 2) засновників;

б) список неблагонадійних фізичних осіб у частині перевірки:

- 1) суб'єктів господарювання;
- 2) засновників;
- 3) уповноважених представників;
- 4) контрагентів;

в) рівень ризику;

г) наявність рахунків, вже відкритих клієнтом у цьому банку.

Контроль виконання разових операцій, що підлягають фінансовому моніторингу:

а) платіж за готівку у розмірі понад 50 тис. грн.;

б) платіж у сумі, перевищує статутний фонд;

в) платіж на рахунки неблагонадійних одержувачів;

г) переказ з рахунку на рахунок одного юридичного чи фізичного особи;

д) одержувачем є не банківський рахунок при переказі грошей на суму понад 25% від статутного фонду.

Контроль виконання повторюваних операцій, що підлягають фінансовому моніторингу:

а) сума платежів за календарний місяць перевищує статутний фонд;

б) одержувачем не є банківський рахунок, сума платежів за календарний місяць перевищує 25% від статутного фонду.

Формування внутрішнього звіту з виконаної роботи, зупинення операцій, що підлягають фінансовому моніторингу - ведення таблиць бази даних, в яких:

- а) будуть вноситись неблагонадійні рахунки для ухвалення рішення про відкриття яких потрібна додаткова перевірка;
- б) вказуватимуться операції, що підлягають фінансовому моніторингу;
- в) зміна у реєстрі платежів стану операції на «зупинена» для подальшого уточнення відповідальним працівником банку.

Таким чином, ми можемо скласти перелік функцій, що виконуються цією програмою. Цей перелік відповідатиме програмним модулям. Табл. 2.3 містить цей перелік із зазначенням позначень та найменувань програмних модулів, а також додатковими поясненнями [7] – [9].

Таблиця 2.3 – Функції застосунку

№ п/п	Позначення	Найменування	Опис
1	2	3	4
1	leg_check	Контроль юридичних осіб	Контроль наявності найменування суб'єкта господарювання, учредителей или контрагентов в стоп-листе (черном списке)
2	phys_ckeck	Контроль фізичних осіб	Контроль наявності суб'єкта господарювання (фізичної особи), засновників, відповідальних працівників, контрагентів у стоп-листі

Продовження табл. 2.3

1	2	3	4
3	add check	Перевірка додаткових даних	Перевірка рівня ризику та наявності рахунків клієнта у цьому ж банку
4	one-off_max	Перевірка разового платежу	Перевірка платежу за готівку у розмірі понад 50 тис. грн.; платежу на суму, що перевищує статутний фонд
5	one-off_bad	Перевірка одержувача	Перевірка платежу на рахунки неблагонадійних одержувачів
6	one-off_same	Перевірка платежу самому собі	Перевірка переказу з рахунку на рахунок однієї юридичної чи фізичної особи
7	one-off_tgt	Перевірка переводів	Перевірка платежу, за якого одержувачем є не банківський рахунок при переказі грошей на суму понад 25% від статутного фонду
8	mon_max	Перевірка накопичення платежів	Перевірка платежів на предмет, коли сума платежів за календарний місяць перевищує статутний фонд
9	mon_tgt	Перевірка накопичення переводів	Перевірка платежів на предмет коли одержувачем є не банківський рахунок, а сума платежів за календарний місяць перевищує 25% від статутного фонду

Кінець табл. 2.3

1	2	3	4
10	ac_probl_reg	Реєстр проблемних рахунків	Ведення реєстру неблагонадійних рахунків для ухвалення рішення про відкриття яких потрібна додаткова перевірка
11	op_probl_reg	Реєстр проблемних операцій	Ведення реєстру операцій, що підлягають фінансовому моніторингу, маркуванню операцій, які слід призупинити

На рис. 2.3 наведена структурна схема програмного засобу, воно є послідовністю операцій з перевірки:

- а) рахунків клієнтів;
- б) платежів.

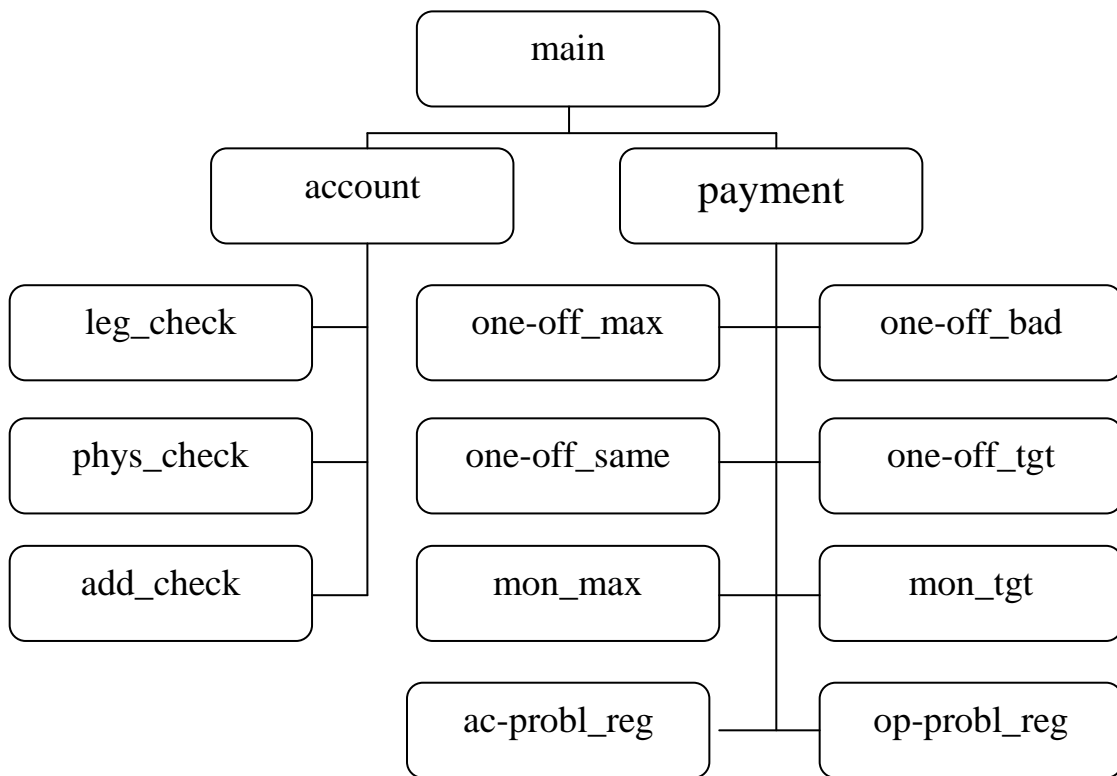


Рисунок 2.3 – Структурна схема додатка

Схема передбачає виконання операцій послідовно одна одною. З головного модуля програми викликаються модулі для обробки рахунків (account) та обробки платежів (payment). Процедури всередині кожного модуля також виконуються послідовно.

Ця структурна схема показує підпорядкованість модулів, але з їх послідовність виконання. Послідовність виконання – це предмет розгляду розробки алгоритму роботи програми.

3 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ

3.1 Вибір платформи проектування

Для проектування використовуємо безкоштовну платформу PHP MySQL.

Розробникам Web-додатків немає необхідності казати, що web-сторінки – це текст і картинки. Цікавий сайт повинен підтримувати певний рівень інтерактивності з користувачем: пошук інформації, продаж продуктів, конференції тощо [10].

Зазвичай усе це реалізувалося CGI-скриптами, написаними на Perl. Але CGI-скрипти дуже погано масштабуються [11]. Кожен новий виклик CGI вимагає від ядра породження нового процесу, а це займає процесорний час і витрачає оперативну пам'ять. PHP пропонує інший варіант – він працює як частина Web-сервера, і цим схожий на ASP від Microsoft.

Синтаксис PHP дуже подібний до синтаксису C або Perl. Люди, які знайомі з програмуванням, дуже швидко зможуть почати писати програми на PHP [12]. У цій мові немає строгої типізації даних і немає необхідності в діях виділення / звільнення пам'яті. Програми, написані на PHP, досить легко читаються. Написаний PHP-код легко візуально прочитати та зрозуміти, на відміну від Perl-програм.

MySQL – компактний багатопоточний сервер баз даних. MySQL характеризується великою швидкістю, стійкістю та легкістю у використанні [13].

Наведемо стислий перелік можливостей MySQL:

- а) підтримується необмежену кількість користувачів, що одночасно працюють з базою даних;
- б) кількість рядків у таблицях може сягати 50 млн.;
- в) швидке виконання команд – можливо MySQL найшвидший сервер із існуючих;
- г) проста та ефективна система безпеки.

Таким чином, ми бачимо, що даний сервер баз даних дуже популярний і дає значні можливості розробки Web додатків. Його ми й використовуватимемо у нашому проекті.

Для розробки системи управління базою даних (СУБД) використовуємо результати попереднього проектування, тому визначимо перелік сховищ даних. До нього відносяться:

- а) стоп-лист (чорний список) юридичних;
- б) стоп-лист фізичних осіб;
- в) анкета клієнта банку;
- г) журнал платежів, що підлягають обробці;
- д) журнал клієнтів, які підлягають моніторингу;
- е) журнал операцій, що підлягають моніторингу.

Тут можна чітко виділити дві групи сховищ даних, які використовуються в програмі, що розробляється:

- а) сховища, що формуються у банку іншими службами або людьми;
- б) сховища, які формуються працівниками банку, що виконують функцію моніторингу.

У цьому випадку вважаємо, що наше завдання – ведення даних, безпосередньо пов'язаних із виявленням клієнтів та операцій, що підлягають фінансовому моніторингу.

До таких сховищ даних належать:

- а) стоп-лист (чорний список) юридичних;
- б) стоп-лист фізичних осіб;
- в) журнал клієнтів, які підлягають моніторингу;
- г) журнал операцій, що підлягають моніторингу.

Перші два сховища даних мають однакове призначення. Вони зведені окремо тільки тому, що щодо них існують дещо різні дані. Крім того, до них потрібно звертатися у різних випадках, тому немає сенсу виконувати подвійний обсяг пошуку. Також ми точно знаємо, коли шукаємо фізичних, а коли юридичних осіб.

У табл. 3.1 наведено перелік даних, що використовуються для ідентифікації підозрілого клієнта – юридичної особи.

Таблиця 3.1 – Дані для стоп-аркуша юридичних осіб

№ п/п	Позначення	Найменування	Пояснення
1	leg_denom_cyr	Найменування (кирилиця)	Повне або скорочене найменування юридичної особи російською або українською мовою
2	leg_denom_eng	Найменування (латиниця)	Повне або скорочене найменування юридичної особи англійською або іншою мовою, яка використовує латинський шрифт
3	leg_egrpou	Код ЄДРПОУ	Код згідно з єдиним державним реєстром підприємств та організацій України або інший ідентифікатор для юридичних осіб нерезидентів України
4	leg__addr	Юридична адреса	Юридична адреса цієї юридичної особи
5	leg_country	Країна	Держава, представником якої є цей суб'єкт
6	leg_reas	Заснування	Заснування в стоп-лист, короткий опис видів діяльності, що призвело до такого рішення

Як видно з таблиці, перелік невеликий, але достатній для того, щоб ідентифікувати, що дана юридична особа може становити небезпеку з точки зо-

ру легалізації доходів, отриманих злочинним шляхом, або фінансування тероризму.

Інші відомості про таку юридичну особу, які зазвичай заповнюються при її створенні, нас не цікавлять, оскільки навряд чи можна використовувати. Додатково слід сказати про країну власності тієї чи іншої суб'єкта.

Справа в тому, що в різних країнах одне й те саме найменування може використовуватися різними суб'єктами. Тому, ідентифікація як за найменуванням, а й у країні власності має сенс.

Аналогічно формуємо перелік даних, які нас цікавлять щодо фізичних осіб, які мають бути занесені в стоп-лист. Ці відомості наведено у табл. 3.2.

Таблиця 3.2 – Дані для стоп-листа фізичних осіб

№ п/п	Позначення	Найменування	Пояснення
1	2	3	4
1	ph_name_cyr	Прізвище ім'я по-батькові (кирилиця)	Прізвище, ім'я по-батькові (якщо є) фізичної особи російською або українською мовою
2	ph_name_eng	Прізвище, ім'я по-батькові (латиниця)	Прізвище, ім'я по-батькові (якщо є) фізичної особи англійською або іншою мовою, що використовує латинський шрифт
3	phj_eg_num	Реєстраційний номер	Реєстраційний номер фізичної особи або інший ідентифікатор для фізичних осіб – нерезидентів України
4	ph_addr	Адреса	Адреса цієї фізичної особи

Кінець табл. 3.2

1	2	3	4
5	ph_country	Країна	Держава, представником якої є цей суб'єкт
6	phjreas	Підстава	Заснування занесення в стоп-лист, короткий опис видів діяльності, що призвело до такого рішення

Як бачимо з табл. 3.1 та табл. 3.2, список даних однаковий, різниця в малих деталях.

Якщо порівняти з даними, які в принципі заповнюються на відповідну юридичну та фізичну особу, то тут різниця суттєва. Проте, для нас головне – та інформація, за якою ми можемо ідентифікувати цю особу.

Наступне, що нам необхідно визначити склад інформації, яка повинна використовуватися для формування журналів клієнтів та операцій, що підлягають моніторингу. Таких журналів два та інформація в них може дещо відрізнятися. Причина цього – різні джерела та цілі. Табл. 3.3 містить відомості про клієнтів, які підлягають моніторингу.

Щодо операцій, що підлягають моніторингу, то відомості про них містяться у табл. 3.4.

Тут ми бачимо деяку відмінність від того, що нам потрібно знати про клієнтів. Ця відмінність обумовлена, як мовилося раніше, різною природою інформації, що використовується для автоматизації фінансового моніторингу. Тут вже присутня інформація не тільки про джерело (відправника), але і про мету (одержувача), призначення платежу та його суму.

Обидві наведені таблиці дають неповну інформацію про клієнта та платежі, яка повинна надаватися компетентним органам у разі, якщо вона підтвердиться.

Таблиця 3.3 – Журнал відомостей про клієнтів, що підлягають моніторингу

№ з/п	Позначення	Найменування	Пояснення
1	cl_name	Прізвище ім'я по-батькові або найменування	Прізвище, ім'я по батькові (якщо є) або найменування клієнта, що підлягає моніторингу при відкритті рахунку
2	cl ac num	Номер рахунку	Номер рахунку клієнта, якщо такий вже є у цьому банку
3	cl_reg_num	Реєстраційний	Реєстраційний номер фізичної особи (для фізичних осіб – не підприємців ідентифікаційний код) або інший ідентифікатор для фізичних осіб – нерезидентів України
4	cl_date	Дата	Дата занесення в журнал
5	cljreas	Підстава	Підстава, на якій даний клієнт занесений до журналу

Розробка програми, яка б генерувала відповідні форми надання компетентним органам, завдання цього проекту не входить. Тут наше завдання – надати необхідну інформацію для того, щоб використати її для пошуку повної інформації в інформаційній системі банку.

Такою інформацією є:

- а) найменування суб'єкта;
- б) ідентифікаційні дані суб'єкта (якщо є);
- в) банківські рахунки суб'єкта (якщо є);
- г) ідентифікатор банківської операції, що викликала певні сумніви.

Таблиця 3.4 – Журнал відомостей про операції, що підлягають моніторингу

№ з/п	Позначення	Найменування	Пояснення
1	op_id	Ідентифікатор	Ідентифікатор фінансової операції у рамках загальної ідентифікації даного банку
2	op_date	Дата	Дата проведення операції, щодо якої виникли сумніви
3	op_srcjame	Відправник	Найменування або ім'я джерела-відправника грошових коштів
4	op_src_ac	Номер рахунку відправника	Номер рахунку відправника
5	op_src_mfo	МФО банку відправителя	МФО банку відправителя
6	op_tgt_name	Одержувач	Найменування або ім'я мети – одержувача коштів
7	op_tgt_ac	Номер рахунку отримувача	Номер рахунку отримувача
8	op_tgt_jnfo	МФО банку одержувача	МФО банку одержувача
9	op_jay	Вид розрахунку	Вид розрахунку, що використовується під час виконання платежу (готівковий, безготівковий, чек)
10	op_purp	Призначення	Призначення платежу
11	op_sum	Сума	Сума платежу
12	op__reas	Реєстрування	Занесення до журналу

3.2 Розробка структури таблиць

Наведені вище дані є вихідною інформацією розробки структури таблиць, формованих СУБД. В основному, ці дані можуть повністю використовуватися. Дійсно, дублювання відомостей, наприклад, про клієнта, насправді відсутнє.

Справа в тому, що ми працюємо:

- а) з клієнтами, які мають рахунки банку;
- б) з клієнтами, які виконують операції готівкою або чеками.

Деякі клієнти відомі лише з англійськими іменами, деякі – лише з російськими тощо. У різних випадках у нас є різна інформація. Універсальним засобом мінімізації зберігання даних є нормалізація таблиць [7]. Проте, як відомо, все гаразд, тому ми зупинимося саме на такій структурі.

Для початку необхідно скласти перелік таблиць, які у нашій роботі. Табл. 3.5 містить такий перелік.

Таблиця 3.5 – Перелік таблиць, що проектуються СУБД

№ з/п	Обозначение	Наименование	Пояснение
1	2	3	4
1	cl_leg	Стоп-лист юридичних осіб	Інформація про юридичних осіб, які мають сумнівну репутацію з точки зору можливості легалізації доходів, одержаних злочинним шляхом, або фінансування тероризму
2	cl_jph	Стоп-лист фізичних осіб	Інформація про фізичних осіб, які мають сумнівну репутацію з точки зору можливості легалізації доходів, отриманих злочинним шляхом, або фінансування тероризму

Кінець табл. 3.5

1	2	3	4
3	cljourn	Журнал з клієнтів	Інформація про клієнтів, які збираються відкрити або мають рахунок у банку, діяльність або анкетні дані про які викликають сумнів
4	orjourn	Журнал з операцій	Інформація про операції, що підлягають фінансовому моніторингу

Тепер необхідно скласти структуру кожної таблиці. Табл. 3.6 –3.10 містять перелік полів, що використовуються у проєктованій СУБД.

Таблиця 3.6 – Структура таблиці cl_leg

Ключ (key)	Позначення	Найменування	Тип	Обов'язковість заповнення
*	leg_id	Ідентифікатор	int auto_increment	+
	leg_denom_cyr	Найменування (кирилиця)	char(40)	
	leg_denom_eng	Найменування (латиниця)	char(40)	
	leg_egrpou	Код ЄДРПОУ	char(15)	
	leg_addr	Юридична адреса	char(30)	
	leg_country	Країна	char(15)	
	reas	Підстава	char(2)	+

У таблиці зі стоп-листом додано поля ідентифікатора. Це зумовлено необхідністю створення первинного ключа таблиць. Враховуючи те, що не завжди всі поля будуть заповнені, ми потребуємо такого ідентифікатора.

Іншою причиною є можлива схожість найменувань юридичних осіб або прізвищ фізичних осіб. Тоді наявність ідентифікатора також усунути можливу плутанину.

Таблиця 3.7 – Структура таблиці cl_ph

Ключ (key)	Позначення	Найменування	Тип	Обов'язковість заповнення
*	ph_id	Ідентифікатор	int auto_increment	+
	ph_name_cyr	Прізвище ім'я по-батькові (кирилиця)	char(40)	
	ph_name_eng	Прізвище, ім'я по-батькові (латиниця)	char(40)	
	ph_reg_num	Реєстраційний номер	char(15)	+
	ph_addr	Адреса	char(30)	
	ph_country	Країна	char(15)	
	reas	Пдстава	char(2)	+

Тут нам необхідно скласти перелік таблиць і полів, які не є об'єктом заповнення нашою програмою. Водночас нам доведеться ними користуватися для організації нормальної роботи СУБД.

У випадку дана інформація не підлягає розголошенню, тому ми складемо свій перелік, який будемо умовно вважати правильним. Даний перелік дозволить швидко адаптувати розроблену програму до будь-якої бази даних банку – достатньо буде уточнити імена баз даних, таблиць та полів, що використовуються у реально існуючих СУБД [8], [9].

Як уже говорилося раніше, у нас є дві таблиці, інформація з яких нам потрібна:

- а) анкети клієнтів;

б) журнал (реєстр) платежів, які підлягають проведенню.

Нехай таблиця з анкетними даними клієнта називається `leg_dossier`, `ph_dossier`, а таблиця з платежами називається `payment`. Необхідно скласти перелік полів таблиць та їх формати, які ми використовуватимемо у роботі.

Цей перелік буде однозначно неповний, але це не повинно нас турбувати, оскільки виконання запиту із вибіркою лише деяких даних із таблиці нам не проблема, оскільки засоби `mysql` дозволяють робити це досить легко та швидко.

Як бачимо: дані, що використовуються у системі, збігаються із зовнішніми даними. Це не дивно, адже ми розробляємо систему моніторингу, завданням якої є спостереження за певними процесами, які відбуваються у банку. Наше завдання – виявити інформацію, що вимагає підвищеної уваги та занести її до реєстру з відповідними поясненнями.

На рис. 3.1 наведено структуру бази даних, що використовується у проєкті. У цю структуру включені як створювані у проєкті таблиці, так й зовнішні таблиці, використовувані у роботі застосунку.

Дану структуру згенеровано за допомогою системи автоматизованого проєктування (САПР) `ErWin` та використовуватиметься вона для розробки програмного коду застосунку.

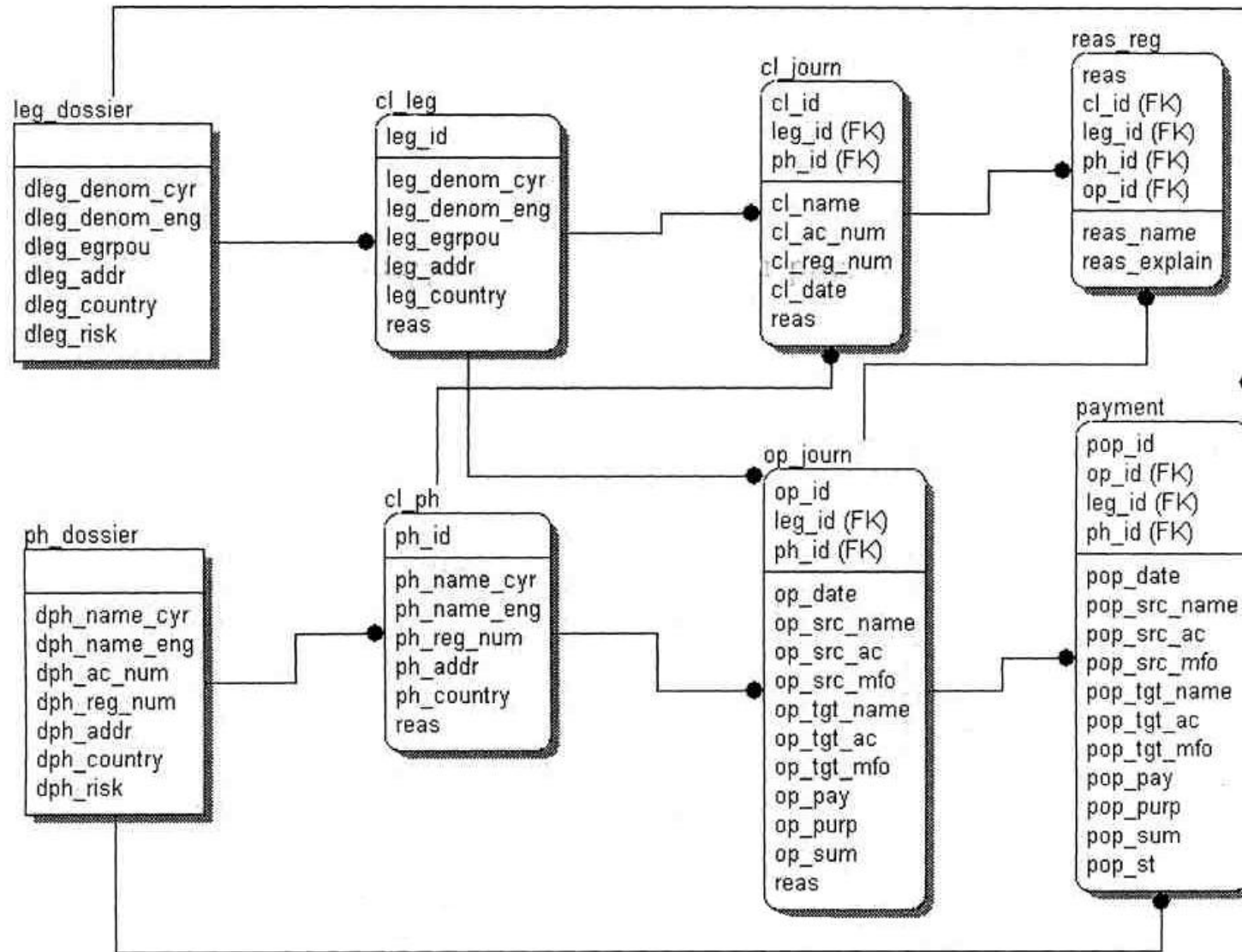


Рисунок 3.1 – Структура бази даних

4 РОЗРОБКА ПРОГРАМНОГО ЗАСОБИ

4.1 Розробка SQL запитів до обробки таблиць БД

Для роботи з базою даних використовуємо мову структурованих запитів (Structured Query Language – SQL).

Як сервер баз даних застосовуємо MYSQL. Для локальної розробки програми використовується пакет Denwer. Цей пакет написаний мовою PERL і повністю емулює файлову систему UNIX, створюючи віртуальний ресурс на мережному диску (якщо використовується операційна система Windows). Це дозволяє здійснювати розробку додатків на локальній машині без встановлення та налаштування Web та SQL серверів.

Перший етап роботи: створення бази даних. Ім'я бази даних: ІКВ – транслітерація найменування програми – ІКБ – ідентифікація клієнта банку.

Для цього спочатку робимо запуск `mysql` з командного рядка:

```
/usr/local/mysql4/bin/mysql -h localhost -n valentine -p
```

Після цього система запросить пароль користувача `valentine` та при успішному введенні пароля з'явиться відповідне запрошення програми `mysql`.

Структура запити для створення бази даних:

```
create database ikd;
```

Наступним етапом є створення таблиць усередині бази даних. Для цього спочатку виконується команда:

```
use ikd;
```

Після цього виконується запит створення таблиць. Структура запити є однотипною, тому не потрібно показувати кожен з них. Складемо лише один запит на створення таблиці, наприклад `op_journ`:

```
create table op_journ (
```

```

op_id int not null auto_increment,
op_date date not null,
op_src_name char(40),
op_src_ac char(20),
op_src_mfo char( 10),
op_tgt_name char(40),
op_tgt_ac char(20),
opjgtjinfo char(10),
op_pay enum('cash', 'check' , 'noncash') not null default
'noncash',
op_purp char(50),
op_sum float(8,2) not null default 0.00,
reas char(2) not null,
primary key (op_id))
type MyISAM;

```

Цей запит створює таблицю з параметрами полів, писаними раніше у цьому проекті.

Наступний основний тип запитів – вибірка з таблиці певного набору даних. Для цього використовується запит SELECT. Наприклад, якщо нам необхідно перевірити, чи знаходиться клієнт, який відкриває рахунок з банку, у стоп-листі, то потрібно спочатку прочитати його ім'я з таблиці ph_dossier і привласнити його будь-якій змінній, наприклад, name.

Потім використовувати змінну name як фільтр запиту з таблиці cl_ph (стоп-аркуш фізичних осіб). При цьому використовується запит типу:

```

select * from cljph where ph_name_cyr like ('$name%') or
ph_name_eng like ('$name%');

```

У цьому випадку, якщо є таке ім'я або йому подібне до стоп-листу, то потрібна додаткова перевірка. В принципі, цей запит може бути спрощений:

```

select * from cl_ph where ph_name_cyr=$name or
ph_name_eng=$name;

```

Однак у цьому випадку система перевірятиме імена лише на їх повний збіг. Досить зробити помилку введення хоч на один символ (наприклад, пробіл наприкінці імені), як такий запит нічого не дасть. Що вибрати конкретно – питання вибору керівництва чи відповідальних працівників банку. Ми будемо використовувати перший варіант як надійніший, хоча ймовірність по-

милки при цьому все-таки є. У той же час ми повинні пам'ятати, що програма, що розробляється, повинна інформувати відповідального працівника банку про можливі незаконні операції, отже, саме він і уточнить уже всі деталі вручну.

Наступний тип запиту, який ми маємо використовувати - додавання записів в таблицю. Наприклад, якщо ми встановили, що черговий рахунок відкривається клієнтом, занесеним у міжбанківський стоп-лист, то потрібно використовувати запит типу:

```
insert into cl_journ
values(0,$cl_name,$cl_ac_num,$cl_reg_num, $cl_date,$reas);
```

У результаті таблицю БД буде додано відповідний запис. Якщо ми маємо неповний набір даних, то можна задати значення змінних за замовчуванням (наприклад, null) і використовувати їх для введення в таблицю полів, які в даному випадку не визначені.

Останній тип запиту, який необхідно використовувати, це оновлення даних у існуючій таблиці. Наприклад, якщо ми хочемо змінити атрибут поля pop_st таблиці payment на stop, що означає припинення виконання операції з ідентифікатором op_jnumber, ми маємо використовувати такий запит:

```
update payment set pop_st='stop' where pop_id=$op_number;
```

Ми розглянули всі типові SQL-запити, які можуть бути використані при розробці СУБД ІКБ. Наступним етапом є розробка програмного коду системи керування базою даних.

4.2 Розробка інтерфейсу програмного засобу

Відповідно до завдання на дипломне проектування, ми повинні розробити програму, засновану на Web-технології. Це означає, що ми маємо вико-

ристовувати прийоми Інтернет-програмування для локальних комп'ютерних мереж, прикладом якої є мережа банку.

Для того, щоб розпочати проектування, потрібно розробити дизайн типової форми програми. Незважаючи на те, що ця програма не призначена для розміщення в Інтернеті, проте певний дизайн все-таки потрібен. Головне – зручність і швидкість доступу до функцій, що реалізується цим додатком.

З іншого боку, немає сенсу робити різні форми з різним дизайном, отже, ми повинні розробити типову форму для програми. Вибираємо таку структуру: вгорі маємо "шапку" форми з найменуванням програми та головним меню. Ця "шапка" може підключатися до будь-якої форми та не вимагає її постійного перезапису. Тому ми говоримо про заголовок веб-форми.

Програмний код, що реалізує цей заголовок, наступний:

```
<html><body>
<table border=0 bgcolor="lightblue" width= 100%>
<tr><td width= 10%>
<A HREF=" " ><IMG HEIGHT=85 WIDTH=100% ALT="LOGO"
SRC="logo.png"x/IMG></A></td>
<td width=90%>
<H1 ALIGN="CENTER"><FONT FACE=GEORGIA COLOR="RED"
size="+20">
<?
echo "Информационно-аналитическая система ИКБ "
?>
</FONT></H 1 ></tdx/tr></table>
<table border=0 bgcolor="RED" width=100%>
<tr>
<tdwidth=100%>
</td>
</tr>
</table>
<tableborder=0bgcolor="lightgrey"width=100%>
<tr>
<tdwidth=12%align=center>
<font size="+1" color="blue">
<A href="Мех.рпр">Главная </A>
</td>
<td width=12% align=center>
<font size="+1" color="blue">
<A href="cl_journ">Журнал клиентов </A>
</td>
<td width= 12% align=center>
<font size="+1" color="blue">
<A href="op_journ.php">Журнал операций </a>
</td>
<td width= 12% align=center>
<font size="+1" color="blue">
```

```

<A href="cl_leg.php"> Стоп лист юр. лиц</a>
</td>
<td width=12% align=center>
<font size="+1" color="blue">
<A href="cl_ph.php"> Стоп лист физ. лиц</a>
</td>
<td width=12% align=center>
<font size="+1" color="blue">
<A href="reas_reg.php"> Реестр причин</a>
</td>
<td width=12% align=center>
<font size="+1" color="blue">
<A href="currency.php"> Курсы валют</a>
</td>
</tr>
</table>
</body>
</html>

```

Цей код містить ще й головне меню. Воно реалізовано у табличній формі та досить зручно для рівномірного розподілу по всій ширині форми (рис. 4.1).

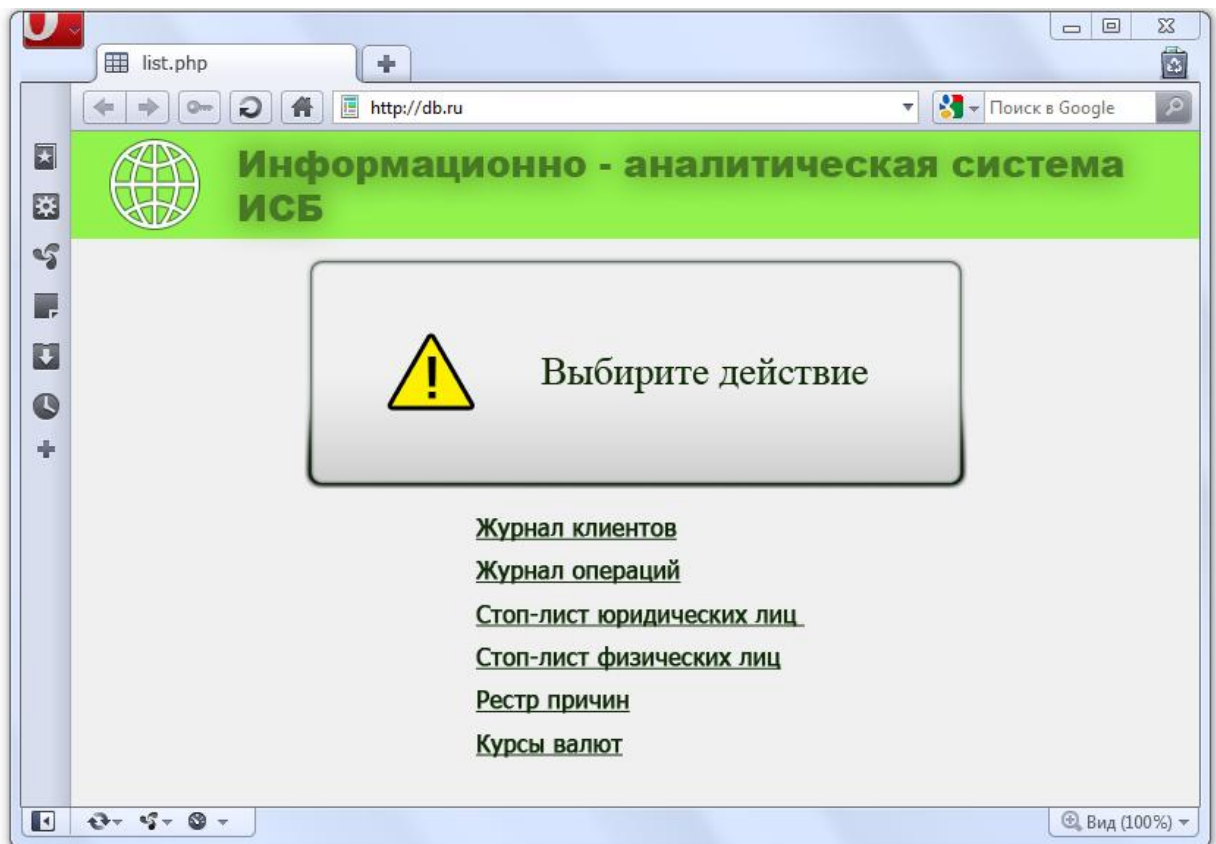


Рисунок 4.1 – Форма головного меню

Нижня частина форми є таблицю, у якій ліва колонка однакова всім і містить повторення головного меню – список всіх базових операцій, виконуваних з допомогою даного застосунку. У правій, основній частині даної таблиці буде поміщатися головна інформація, що стосується кожної конкретної підзавдання, що розв'язується в рамках інформаційно-аналітичної системи (ІАС), що проектується [10].

На головній сторінці програми в цій частині розташовується знову ж таки перелік гіперпосилань, що відповідають окремим функціям, що виконується програмою, що розробляється.

Надалі ми розглядатимемо лише праву нижню частину форми, тому що решта у всіх вікнах буде однакова, причому для цього не пишеться якийсь великий програмний код. Підключення заголовка Веб-форми здійснюється шляхом виконання команди:

```
<? include ('head.php'); ?>
```

4.3 Розробка підсистеми інформаційного забезпечення програми

В даному випадку під поняттям "підсистема інформаційного забезпечення" розуміємо ту частину програми, яка виконує функції введення та коригування даних, що використовуються у додатку. Такими даними є:

- а) стоп-лист юридичних;
- б) стоп-лист фізичних осіб;
- в) реєстр причин необхідності проведення фінансового моніторингу;

Розробку програмного коду розглянемо з прикладу ведення стоп-листа юридичних (раніше у проекті неодноразово застосовувався синонім даного терміна – " чорний список " юридичних осіб).

Для ведення даного списку необхідно не тільки супроводжувати таблицю зі стоп-листом, але також таблицю з реєстром можливих причин занесення до цього списку. Це пов'язано з тим, що в стоп-лист міститься лише умовний ідентифікатор причини. У той же час оператор повинен мати можливість

переглянути список причин для того, щоб правильно ввести ідентифікатор в стоп-лист. Приклад вікна введення даних у стоп-аркуш показано на рис. 4.2.

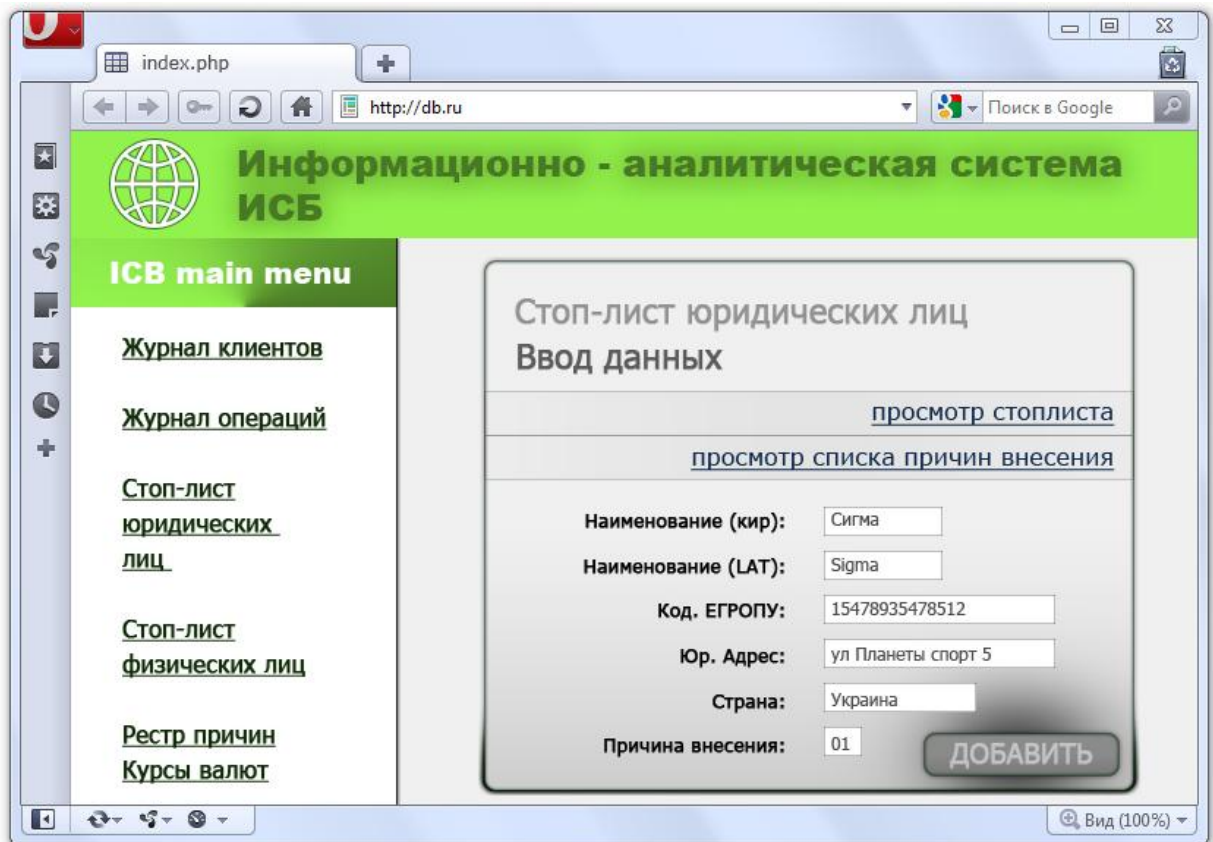


Рисунок 4.2 – Вікно уведення даних до стоп-аркуша

Тут і далі ми будемо наводити тільки вміст основного осередку форми, щоб уникнути повторення програмного коду, що є в усіх екранних формах.

Фрагмент реєстру причин внесення до стоп-листу наведено на рис. 4.3.

Тут здійснюється виведення даних як таблиці. Використовуючи відповідне посилання на сторінці з введенням інформації в стоп-лист, можна ознайомитися з кодами причин та використовувати їх для внесення до стоп-листу.

Тут ми вперше використовуємо мову запитів для внесення змін до бази даних. Розглянемо це з прикладу файлу `reas_reg_add.php`, реалізує введення даних, які були попередньо сформовані з допомогою `html`-тегу «`form`» у файлі `reas_reg.php`.

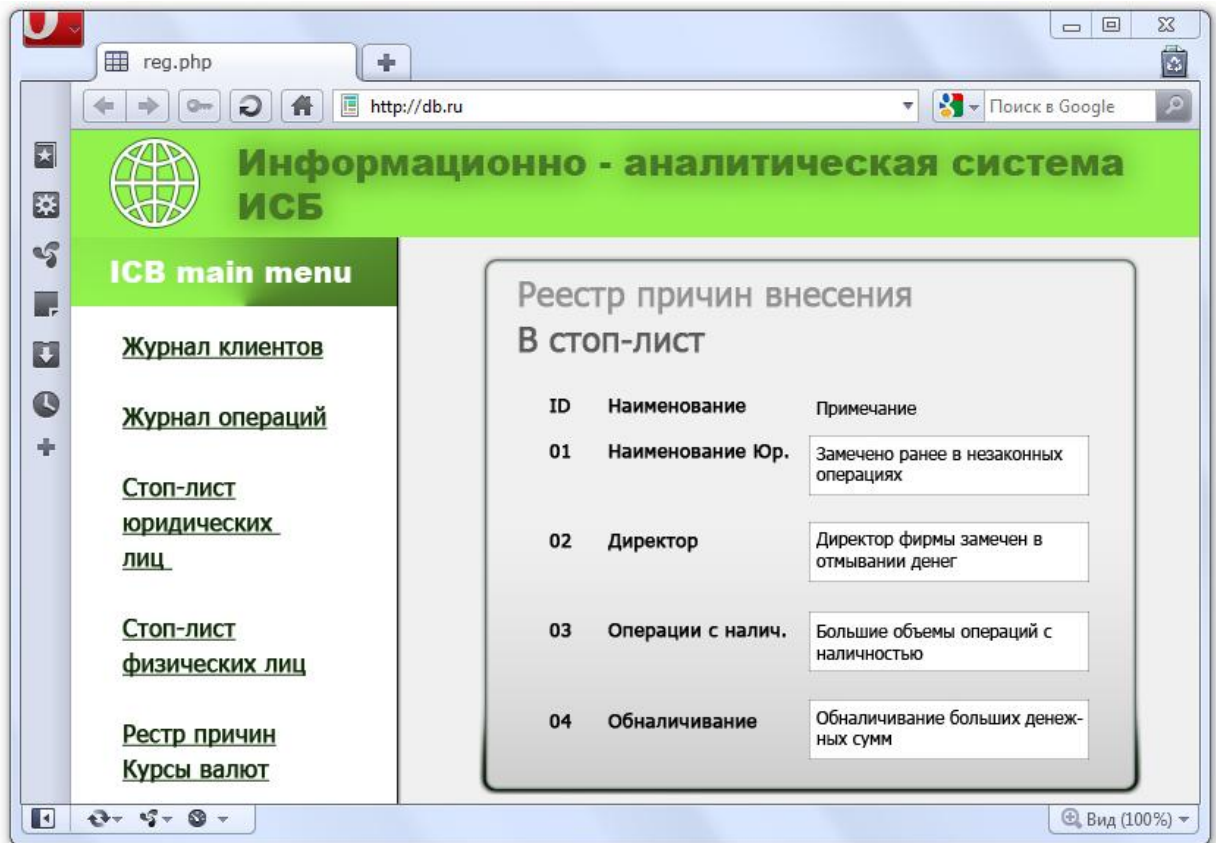


Рисунок 4.3 – Приклад перегляду реєстру причин внесення до стоп-аркушу

Тут все починається з блоку даних, що задає параметри підключення до бази даних. Ці дані можуть задаватися в окремому файлі та підключатися за допомогою команди «include». В даному випадку для простоти та наочності ми це робимо прямо у файлі.

Після цього слід ланцюжок команд, що дозволяють підключитися до бази даних. Це команди `mysql_connect` та `mysql_select_db`. Після запуску цих команд, ми готові працювати безпосередньо з таблицями бази даних.

Наступна ключова команда – `mysql_query`. Вона дозволяє виконати SQL запит і присвоїти результат його виконання будь-якої змінної.

Потім, застосовуючи команду «`mysql_fetch_row`», ми перетворимо результат виконання запиту в масив даних, який потім може бути в циклі рядково виведений у будь-яке місце, яке ми поставимо. У нашому випадку все це виводиться у html таблицю.

Тут і надалі буде використовуватися така послідовність дій при виконанні операцій обробки даних, тому більше ми не акцентуватимемо на цьому увагу.

Приклад результату перегляду стоп-листа наведено на рис. 4.4.

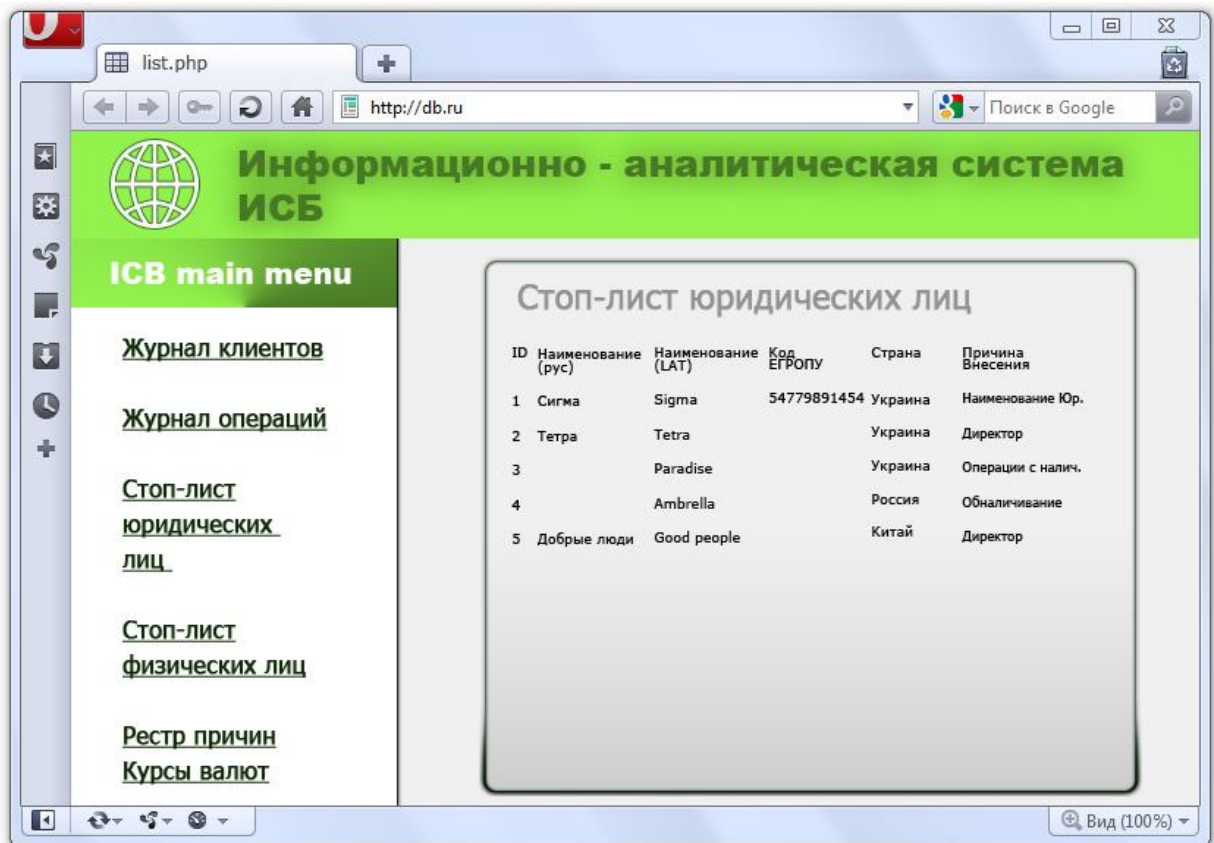


Рисунок 4.4 – Приклад результату перегляду стоп-аркуша юридичних осіб

Тут застосований ще один хід, який дозволив використовувати результати нормалізації таблиць – в одну екранну таблицю заносяться дані із двох таблиць бази даних:

- а) стоп-лист юридичних;
- б) реєстр причин.

З реєстру причин на підставі коду причини занесення до Реєстру, витягується найменування даної причини і поміщається у відповідну комірку екранної таблиці. Для цього служить змінна «\$q2», яка використовується для створення другого запиту, причому виконання даного запиту поміщається

всередину циклу рядкового зчитування записів зі стоп-листа, таким чином, що кожен раз, як черговий запис виводиться в таблицю, з реєстру причин вилучається найменування причини на підставі її коду і поміщається в останню колонку екранної таблиці. Всі інші операції, що виконуються за допомогою програми, що розробляється, будуються за аналогічним принципом, тому немає сенсу їх докладно розглядати.

4.4 Аналіз вхідних даних за допомогою автоматизованої системи

Дуже важлива частина програми, що розробляється – аналіз даних, що отримуються з інших джерел – анкет клієнтів, що відкривають або мають рахунки в банку, журналу платежів, на предмет виявлення можливих причин фінансового моніторингу [11].

Тут ми розглянемо дані операції з прикладу аналізу анкетних даних клієнтів. Ми не будемо повністю повторювати код програмного засобу, оскільки зразки складання такого коду наведено раніше.

Наше завдання – алгоритм обробки даних та його реалізація у вигляді php-коду. Для того, щоб виконати таку обробку, потрібно витягти дані з таблиці з анкетами клієнтів та виконати почергову перевірку даних щодо виявлення можливих причин внесення до Реєстру підозрілих клієнтів [12].

Для цього ми використовуємо анкетні дані:

- а) `dleg_denom__cyr` – найменування (кирилиця);
- б) `dleg_denom_eng` – найменування (латиниця);
- в) `dleg_egrpou` – Код ЄДРПОУ;
- г) `dleg_addr` – юридична адреса;
- д) `dleg_country` – к країна.

Тут нас цікавлять найменування (кириличне та латинське) даного суб'єкта господарювання. Для того, щоб перевірити його надійність, ми маємо по черзі порівняти обидва найменування з усіма записами у стоп-листі юридичних осіб.

Алгоритм роботи наступний:

- а) прочитати перший запис із таблиці анкет;
- б) прочитати перший запис зі стоп-листа;
- в) порівняти попарно латинське та російське найменування аналізованої юридичної особи з тим, яка рахується зі стоп-листа:
 - 1) якщо збіг відсутній, перейти до пункту г;
 - 2) якщо хоч одне з найменувань співпало, то занести цю юридичну особу в журнал клієнтів та змінити поле "ступінь ризику" анкети на high (висока);
- г) рахувати наступний запис зі стоп-листа;
- д) якщо запис ще є, перейти до пункту;
- е) прочитати наступний запис із таблиці анкет;
- ж) якщо запис є, перейти до пункту б;
- и) алгоритм завершено.

Цей алгоритм чітко показує два вкладені цикли:

- а) читання даних із таблиці анкет;
- б) читання даних із стоп-листа юридичних.

Неважко припустити, що для цього достатньо використати запити select. Назвемо запит даних з анкет "anc" та запит зі стоп-листа юридичних осіб – "lsl".

Код запиту з анкет наступний:

```
$anc = mysql_query("select * from leg_dossier");
```

Результат виконання цього запиту – перелік змінних, зазначених раніше із префіксом leg.

Пошук збігів із записами стоп-листа організуємо за принципом фільтрації даних із таблиці стоп-листа. Запит має вигляд:

```
$lsl = mysql_query("select * from cl_leg where leg_denom_cyr like ('dleg_denom__cyr%') or leg_denom_eng like ('dleg_denom_eng%')");
```

Цей запит передбачає можливість контролю імен навіть при їх неповному збігу (достатньо, щоб збігся початок). Якщо буде знайдено клієнтів, які потрапили під фільтр випадково, то відповідальний працівник банку це відстежить. За бажання структуру запиту можна змінити на перевірку повного збігу.

Наступний момент – як ми можемо перевірити наявність таких збігів. Для цього потрібно просто порахувати кількість рядків, отриманих у результаті запиту:

```
if (mysql_num_rows($ls1)>0)
{...
}
```

Дана конструкція дозволить оминати необхідність виконання двох запитів (для підрахунку рядків збігів, а потім – для вибірки цих рядків).

Якщо кількість збігів більше нуля, необхідно інформацію про клієнта занести у відповідний журнал (таблиця `cljourn`), також змінити атрибут поля "ступінь ризику" даного клієнта.

В результаті буде змінено атрибут "ступінь ризику" в анкеті клієнта та внесено запис до журналу реєстрації підозрілих клієнтів, який потім буде використаний відповідальним працівником банку.

Ця частина програми не має свого інтерфейсу, тому що вона виконується автоматично.

Для перегляду результатів роботи програми необхідно скористатися типовою процедурою перегляду ресурсу бази даних. Як скласти таку процедуру, описувалося раніше.

4.5 Тестування програмного забезпечення

Вже наголошувалося, що для забезпечення нормальної роботи програми використовується інформація, що формується іншими службами банку.

Щоб перевірити правильність роботи розробленого додатка, необхідно розробити власне тестове програмне забезпечення.

Воно передбачає ведення трьох таблиць:

- а) анкет юридичних;
- б) анкет фізичних осіб;
- в) журналу платежів.

У базі даних ці ресурси представлені таблицями «leg_dossier», «ph_dossier» та «payment» відповідно.

Приклад Інтернет форми з введення таких даних наведено на рис. 4.5.

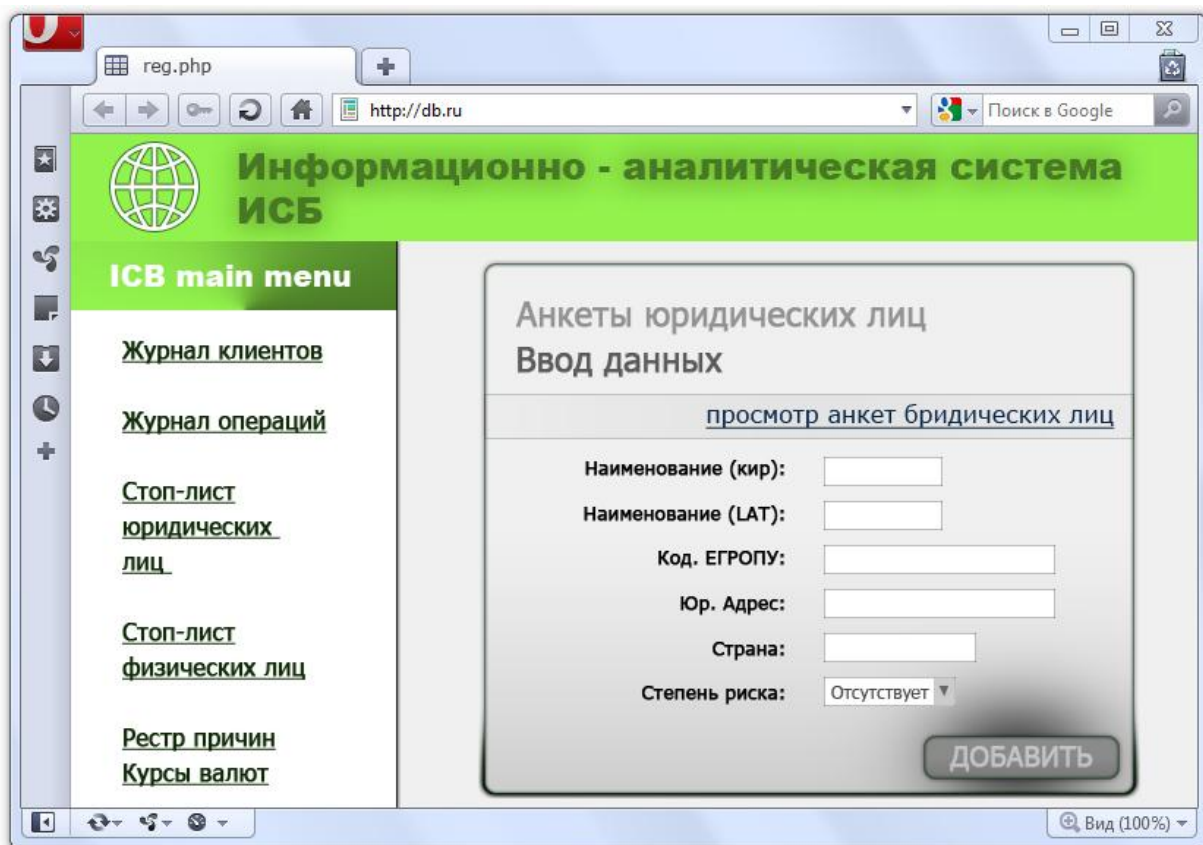


Рисунок 4.5 – Приклад форми введення даних у тестовому ПЗ

Як бачимо, жодної відмінності від розглянутих раніше методів проектування немає. Єдина різниця полягає в тому, що для запуску використовується не файл «index.php», як в основному додатку, а файл «/test.php», який знаходиться в тому ж каталозі, але повинен запускатися вручну.

Слід зазначити, що у цьому проекті розглядалася насамперед модель, використовувана для ідентифікації клієнтів та платежів у банку.

Що стосується питань безпеки програмного забезпечення, це може розглядатися як наступна стадія роботи за умови впровадження додатка. Справа в тому, що в будь-якому банку існують свої правила безпеки програмного забезпечення, і вони є секретною інформацією – тому у даному проекті вони і не розглядалися.

ВИСНОВКИ

Виконання кваліфікаційної роботи бакалавра показало, що ідентифікація клієнта – це дуже ємна операція, яка виконується за двома основними напрямками:

- а) ідентифікація за анкетними даними;
- б) ідентифікація по операціях, що проводяться.

При цьому кожен із напрямків є відносно самостійним і вимагає окремого опрацювання.

При виконанні проекту отримано такі результати:

- а) на основі аналізу нормативної бази України та досвіду інших країн сформовано перелік основних ознак, за якими можливе проведення процедур автоматизованого моніторингу клієнтів банку;
- б) складено схеми взаємодії між підрозділом фінансового моніторингу та загальною інформаційною системою банку;
- в) розроблена загальна структурна схема додатка, що включає операції з контролю анкетних даних клієнтів банку та операцій;
- г) розроблено структурну схему бази даних, що використовується в роботі, сформовано перелік даних, що формуються зовнішніми підрозділами банку;
- д) у рамках специфікованої підзадачі ідентифікації клієнта банку розроблено Web-додаток, що реалізує функції підготовки даних для подальшого аналізу;
- е) з метою забезпечення можливості автономного тестування програми розроблено окремий додаток, що імітує роботу інформаційної системи банку в межах вхідних даних, що нас цікавлять.

Для складання структури бази даних використано CASE засіб ERWin.

Для роботи програми вибрано такі компоненти програмного забезпечення:

- а) Web-сервер – APACHE;

- б) сервер баз даних – MySQL;
- в) мова програмування – PHP.
- г) для локальної розробки програми використаний емулятор пакету APACHE-PHP-MYSQL – віртуальний хостинг Denwer.

Цей комплекс програмного забезпечення, маючи необхідну продуктивність та функціональність, побудований з використанням вільного програмного забезпечення без необхідності його ліцензування.

Розроблений додаток є мультиплатформним і може працювати під різними операційними системами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про затвердження Положення про здійснення банками фінансового моніторингу/ Правління Національного банку України. Постанова №189 від 14.05.2003. Зареєстровано в Міністерстві юстиції 20.05.2003, № реєстрації 381/7702.
2. Velykodniy S. S. Analysis and synthesis of the results of complex experimental research on reengineering of open CAD systems. *Applied Aspects of Information Technology*. 2019. Vol. 2. No 3. P. 186–205. (кат. «Б») DOI: 10.15276/aait.03.2019.2.
3. Velykodniy S., Burlachenko Zh., Zaitseva-Velykodna S. Modelling the behavioural component of the emergent parallel processes of working with graph databases using Petri net-tools. *International Journal of Parallel, Emergent and Distributed Systems*. (Scopus) 2021. Vol. 36. Iss. 6. P. 498–515. DOI: <https://doi.org/10.1080/17445760.2021.1934836>. Taylor & Francis Group, England & Wales. London.
4. Velykodniy S. Reengineering of open software system of 3D modeling BRL-CAD. *Innovative Technologies and Scientific Solutions for Industries*. 2019. No 3 (9), P. 62–71. (кат. «Б») DOI: <https://doi.org/10.30837/2522-9818.2019.9.062>.
5. Марков М. Проблемы финансового мониторинга в банковской сфере. URL: [http:// finansy.com](http://finansy.com).
6. Про затвердження складу реквізитів та структури файлів інформаційного обміну між спеціально уповноваженим органом виконавчої влади з питань фінансового моніторингу та банками (філіями)/ Правління Національного банку України. Постанова №233 від 04.06.2003. Зареєстровано в Міністерстві юстиції 09.06.2003, № реєстрації 457/7778.
7. Про затвердження Правил зберігання, захисту, використання та розкриття банківської таємниці/ Правління Національного банку України.

8. Постанова №267 від 14.07.2006. Зареєстровано в Міністерстві юстиції 03.08.2006, № реєстрації 935/12809.
9. Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербург: Знание, 2004. 162 с.
10. Котеров Д. В. Самоучитель PHP 4: Учеб. пособ. Санкт-Петербург: БХВ-Петербург, 2012. 580 с.
11. Великодний С. С. Проблема реинжиниринга видов обеспечения систем автоматизированного проектирования. *Управляющие системы и машины*. 2014. № 1. С. 57–61, 76.
12. Великодний С. С. Методологические основы реинжиниринга систем автоматизированного проектирования. *Управляющие системы и машины*. 2014. № 2. С. 39–43.
13. Petukhin D., Velykodniy S., Kozlovskaya V. Modeling the space of possible states of the lesson schedule in higher education institutions. International Scientific and Practical Conference "Intellectual Systems and Information Technologies", 13-19 sep. 2021, Odesa, Ukraine. PP. 230–237.
14. Котлер Дж. Маркетинговые исследования. Москва: Финансы и статистика, 1991. 220 с.
15. Великодний С. С. Моделі та методи проактивного управління проектами із розвитку програмних систем і продуктів. Монографія. Одеса: Одеський державний екологічний університет, 2021. 322 с. ISBN 978-966-186-182-3. (URL: <http://eprints.library.odeku.edu.ua/id/eprint/9595/>)