

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра автоматизованих систем  
моніторингу навколишнього середовища

**Бакалаврська кваліфікаційна робота**

на тему: Розробка системи ідентифікації та моніторингу банківського клієнта

Виконав студент 2 (інт.) курсу  
групи К-25 (інт.)  
спеціальність 122 – комп'ютерні науки  
Дяченко Дмитро Анатолійович

Керівник к. т. н., доцент  
Великодний Станіслав Сергійович

Консультант \_\_\_\_\_  
\_\_\_\_\_

Рецензент к. т. н.,  
Онищенко Сергій Михайлович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ  
Факультет комп'ютерних наук, управління та адміністрування  
Кафедра автоматизованих систем моніторингу навколишнього середовища  
Рівень вищої освіти бакалавр  
спеціальність 122 – комп'ютерні науки

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри АСМНС

“13” квітня 2020 р.

**З А В Д А Н Н Я**  
**НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Дяченку Дмитру Анатолійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи ідентифікації та моніторингу банківського клієнта

керівник роботи Великодний Станіслав Сергійович, к. т. н., доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від «13» квітня 2020 р. № 37-С

2. Строк подання студентом роботи 30 травня 2020 р.

3. Вихідні дані до роботи: Web-застосунок, що реалізує ідентифікацію клієнта банку та функції підготовки даних для подальшого аналізу; окремий застосунок, що імітує роботу інформаційної системи банку; компоненти програмного забезпечення: Web-сервер – Apache; сервер баз даних – MySQL; мова програмування – PHP; для локальної розробки використовується емулятор пакета Apache-PHP-MySQL – додаток «Denwer»; складання структури бази даних за допомогою CASE-засобу ERWin; розроблений застосунок має бути крос-платформним

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

вступ; аналіз вихідних даних та вимог завдання на кваліфікаційну роботу; розробка структурних схем застосунків; розробка системи управління базою даних; вибір платформи проектування; розробка структури таблиць; розробка програмного засобу; розробка SQL-запитів для обробки таблиць; розробка інтерфейсу програмного засобу; висновки; додаток

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):  
графічний матеріал виконаний у вигляді презентації

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання “13” квітня 2020 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Оцінка виконання етапу	
			у %	за 4-х бальною шкалою
1	Отримання завдання на роботу	13.04.2020		
2	Аналіз вихідних даних	16.04.2020		
3	Виконання аналітичного розділу	19.04.2020		
4	Розробка структурних схем для застосунку	22.04.2020		
5	Розробка системи управління базами даних	04.05.2020		
6	Розробка структури таблиць	08.05.2020		
7	Проходження рубіжної атестації	11.05.2020		
8	Розробка програмного засобу	18.05.2020		
9	Розробка інтерфейсу програмного засобу	24.05.2020		
10	Тестування програмного забезпечення	28.05.2020		
11	Формування висновків до роботи	29.05.2020		
12	Перевірка роботи на плагіат	30.05.2020		
13	Складання графічної частини	02.06.2020		
14	Підготовка доповіді на захист	06.06.2020		
15	Проходження рецензування	08.06.2020		
	<b>Інтегральна оцінка виконання етапів календарного плану (як середня по етапам)</b>			

Студент

\_\_\_\_\_

(підпис)

Керівник роботи

\_\_\_\_\_

(підпис)

Дяченко Д. А.

(прізвище та ініціали)

Великодний С. С.

(прізвище та ініціали)

## ЗМІСТ

Перелік умовних позначок .....	5
Вступ.....	6
1 Аналіз вихідних даних та 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 структурних схем застосунків.....	15
2.1 Розробка заходів з автоматизації фінансового моніторингу у банку ....	15
2.2 Структурна схема.....	20
3 Розробка системи управління базою даних.....	25
3.1 Вибір платформи проектування .....	25
3.2 Розробка структури таблиць .....	28
4 Розробка програмного засобу .....	35
4.1 Розробка SQL-запитів для обробки таблиць БД.....	35
4.2 Розробка інтерфейсу програмного засобу.....	38
4.3 Розробка підсистеми інформаційного забезпечення застосунку .....	39
4.4 Аналіз вхідних даних за допомогою автоматизованої системи.....	43
4.5 Тестування програмного забезпечення.....	46
Висновки .....	48
Перелік джерел посилання .....	50
Додаток А Фрагменти програмного коду виконаного застосунка .....	53
А.1 Програмний код, який реалізує заголовок.....	53
А.2 Код головної сторінки програми .....	54
А.3 Програмний код для прикладу реалізації вікна введення даних до стоп-листа .....	55
А.4 Приклад файлу, що реалізує введення даних, які були попередньо сформовані за допомогою html-тега .....	55
А.5 Результат виконання запиту у сформований масив даних .....	56
А.6 Результуючий код занесення інформації про клієнта до журналу .....	58

## ПЕРЕЛІК УМОВНИХ ПОЗНАК

БД – база даних.

ЄДРПОУ – Єдиний державний реєстр підприємств та організацій України.

ІКБ – ідентифікація клієнта банку.

КК – кримінальний кодекс.

МФО – міжфіліальні обороти.

НБУ – Національний Банк України.

ОС – операційна система.

ПДВ – податок на додану вартість.

САПР – система автоматизованого проектування.

СУБД – система управління базами даних.

ЦБ – центральний банк.

CASE – Computer-Aided Software Engineering.

CGI – Common Gateway Interface.

PHP – Hypertext Preprocessor.

SQL – Structured Query Language.

## ВСТУП

Сьогодні однією з найбільш актуальних і обговорюваних тем є впровадження, оптимізація та реалізація (здійснення) системи фінансового моніторингу у всіх областях фінансової системи України. Основним нормативним документом є Положення про здійснення банками фінансового моніторингу Правління Національного банку України.

Під ідентифікацією фізичної особи розуміються дії співробітників банку, спрямовані на встановлення приналежності документів тій особі, які ця особа пред'являє в посвідчення своєї особистості і своїх правомочностей при встановленні клієнтських відносин з банком. Документи мають бути при цьому з розумною ретельністю перевірені на предмет їх дійсності і наявності всіх необхідних реквізитів. Вимога ідентифікації клієнта виключає законну можливість відкриття банківського рахунку та здійснення банківських операцій за пред'явленням ксерокопії документа, документа на чуже ім'я або документів з явними ознаками підробки.

У разі, коли клієнтом є юридична особа, то завдання співробітників банку значно ускладнюється. Ідентифікація юридичної особи полягає в перевірці та оцінці документів, представлених у посвідченні його правового статусу і підтверджують його державну реєстрацію, у встановленні власників цього підприємства, а також в діях, спрямованих на отримання достовірної інформації з метою про передбачуваний характер ділових відносин.

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

Тому пропонується тема дипломної роботи носить актуальний характер.

Мета дипломної роботи – спроектувати автоматизовану систему ідентифікації клієнта банку при виконанні ним фінансових операцій.

Об'єкт роботи – процес розробки системи «банк-клієнт».

Пояснювальна записка до бакалаврської кваліфікаційної роботи складається з: 58 сторінок; що містять, 4 розділи, 11 рисунків, 9 таблиць, 1 додаток на 6 сторінок та перелік джерел посилань з 18 найменувань.

# 1 АНАЛІЗ ВИХІДНИХ ДАНИХ ТА ВИМОГ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

## 1.1 Аналіз вихідних даних

В технічному завданні на дипломне проектування поставлена задача розробити заходи і програмне забезпечення, покликане забезпечити автоматизацію моніторингу клієнтів банку з точки зору відмивання доходів, отриманих незаконним шляхом і можливого фінансування терористичної діяльності.

Підзадачею проектування є розробка програмного засобу, призначеного для автоматизації ідентифікації клієнта банку. При цьому додаток, що розробляється у рамках проекту, має бути виконаний у вигляді Web-застосунку із використанням однієї з сучасних технологій програмування [1] <sup>1)</sup>.

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

Як керівна інформація виступає нормативна база України і постанови Національного банку України в частині фінансового моніторингу та забезпечення його роботи. Згідно з [2] <sup>2)</sup>, дана тема підходить під категорію «Розробка програми ведення обліково-звітної документації».

При цьому дана програма виконується за технологією «Клієнт-сервер». Використанням технології клієнт-сервер обумовлено тим, що використання даного програмного засобу можливо кількома співробітниками банку, які

---

<sup>1)</sup> [1] Великодний С. С. Методи реінжинірингу програмних систем. *Технологии приборостроения*. 2014. Спец. вып. С. 65–68.

<sup>2)</sup> [2] Великодний С. С. Проблема реінжиніринга видів обеспечення систем автоматизованого проектування. *Управляющие системы и машины*. 2014. № 1. С. 57–61, 76. Видання національної академії наук України.



працюють в рамках відокремленого підрозділу по фінансовому моніторингу. При цьому вони будуть використовувати загальне сховище даних [3] <sup>1)</sup>.

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**

Банк – фінансово-кредитний інститут, установа, яка провадить різноманітні види операцій з грошима і цінними паперами і що надає фінансові послуги уряду, підприємствам, організаціям, громадянам та іншим банкам [4] <sup>2)</sup>. Банки випускають, акумулюють, зберігають, надають у кредит, розміщують, купують і продають, обмінюють гроші і цінні папери, контролюють рух грошових коштів, обіг грошей і цінних паперів, надають послуги з платежів і грошових розрахунків; здійснюють посередницькі та трастові (довірчі) операції. Розрізняють два основні різновиди банків, що утворюють разом дворівневу систему:

- а) центральний банк (ЦБ) – головний державний банк країни, наділений особливими правами (в Україні – це Національний банк України НБУ)): ЦБ покликаний регулювати грошовий обіг у країні, здійснювати грошову емісію, регулювати кредит і валютний курс, контролювати діяльність комерційних банків, зберігати резерви і запаси грошових коштів золота, ЦБ прийнято також називати «Банком банків»;

---

<sup>1)</sup> [3] Velykodniy S. S., Burlachenko Zh. V., Zaitseva-Velykodna S. S. Method of presenting the assessment for reengineering of software systems with the project coefficients help. Інформаційні технології в моделюванні: матер. IV-ої всеукр. наук.-практ. конф. студ., аспірантів та мол. вчених, 21–22 березня 2019 р., м. Миколаїв. Миколаїв: МНУ ім. В. О. Сухомлинського, 2019. С. 10–11.

<sup>2)</sup> [4] Про затвердження Положення про здійснення банками фінансового моніторингу. Правління Національного банку України. Постанова №189 від 14.05.2003. Зареєстр. в Міністерстві юстиції 20.05.2003, № реєстр. 381/7702.

б) комерційні банки – найчастіше недержавні банки, що виконують широке коло банківських операцій, які обслуговують переважно підприємства, фірми, організації, установи і надають банківські послуги населенню; можуть бути як універсальними, тобто які виконують широке коло операцій, так і спеціалізованими (ощадні, іпотечні, інвестиційні, клірингові тощо).

Комерційний банк (далі Банк) – це кредитна установа, що надає банківські послуги для підприємств, організацій, громадян (розрахункові, платіжні операції, залучення вкладів, надання позик, а також операції на ринку цінних паперів). Грошові кошти для проведення операцій такий банк формує головним чином за рахунок заощаджень, залучених у вигляді внесків, міжбанківських кредитів, випуску власних акцій і облігацій. Комерційні банки функціонують під контролем з боку центрального банку (зокрема, НБУ) і можуть виконувати операції за дорученням державних фінансових органів, взаємодіяти з ними.

Легалізація доходів – це термін, який використовується переважно в правовій сфері. Чи означає надання правомірного вигляду володінню, користуванню земельними ділянками або розпорядженню грошовими коштами або іншим майном, одержаними внаслідок вчинення злочину, за винятком злочинів, передбачених відповідними статтями Кримінального Кодексу (КК). У тому числі фінансування тероризму – надання чи збір коштів або надання фінансових послуг з усвідомленням того, що вони призначені для фінансування організації, підготовки та здійснення хоча б одного із злочинів, передбачених відповідними статтями КК, або для забезпечення організованої групи, незаконного збройного формування або злочинного співтовариства (злочинної організації), створених або створюваних для здійснення хоча б одного з подібних злочинів.

Фінансовий моніторинг – це поняття, схоже з поняттям фінансового контролю, а також безпосередньо пов'язане з обов'язковими процедурами внутрішнього контролю в частині проведення фінансових операцій. Банк ор-

ганізовує внутрішній контроль з урахуванням рекомендацій по організації внутрішнього, контролю за ризиками банківської діяльності. Фінансовий моніторинг – це також діяльність організацій, що здійснюють операції з грошовими коштами або іншим майном, щодо виявлення операцій, що підлягають обов'язковому контролю, і інших операцій з грошовими коштами або іншим майном, пов'язаних з легалізацією (відмиванням) доходів, одержаних злочинним шляхом, і фінансуванням тероризму .

Система внутрішнього контролю організовується органами управління банку, уповноваженими установчими документами банку. З метою моніторингу за процесом функціонування системи внутрішнього контролю, виявлення і аналізу проблем [5]<sup>1)</sup>, пов'язаних з нею, а також розробки пропозицій щодо вдосконалення і 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 прийняті відповідно до законодавства України заходи.

Багато клієнтів банків – зайняті ділові люди, власники бізнесу і приватні підприємці, знати про тонкощі Банківського законодавства їм зовсім не обов'язково та й шляхи проведення тих чи інших операцій іноді бувають дуже неординарними на увазі ряду об'єктивних і суб'єктивних причин. Існує і зворотна проблема, коли співробітники Банку починають «консультувати» клієнтів про те, як грамотно уникнути комісій Банку, не потрапити під «зірке око» внутрішньобанківських служб з фінансового моніторингу тощо Це, здавалося б, дрібниці, але багатьох з них, та й несвоєчасних повідомлень до компетентних державних структур можна було б уникнути за введенням автоматизованих інформаційних систем.

Діяльність банку в частині фінансового моніторингу регламентується, в першу чергу, Положенням Національного банку України щодо здійснення

---

<sup>1)</sup> [5] Великодний С. С., Тимофєєва О. С., Зайцева-Великодна С. С., Нямцу К. Є. Порівняльний аналіз властивостей відкритого, вільного та комерційного програмного забезпечення. *Інформаційні технології та комп'ютерна інженерія*. 2018. № 1 (41). С. 21–27.

банками фінансового моніторингу [4] <sup>1)</sup>. Дане Положення передбачає розробку банками власних Правил здійснення фінансового моніторингу на підставі вимог НБУ.

На рис. 1 наведено основні ознаки легалізації доходів, отриманих незаконним шляхом [6] <sup>2)</sup>.

Дана схема досить наочно показує, в яких випадках слід бути особливо уважним при виконанні функцій ідентифікації клієнта банку.

На рис. 2 показані основні рекомендовані заходи з протидії відмиванню доходів, отриманих незаконним шляхом [6].

Наведена на рис. 1, 2 інформація узагальнює існуючий світовий досвід протидії різним видам незаконної діяльності по відмиванню доходів та фінансуванню терористичної діяльності.

На жаль, нормативна база недостатньо опрацьована в частині визначення, які операції підлягають фінансовому моніторингу. З іншого боку, в Україні дана інформація віднесена до компетенції окремих банків, отже, вона є закритою, не призначеної для широкого користування.

Таким чином, можна зробити наступний висновок: роль банків в діючій системі протидії відмиванню брудних грошей і фінансуванню тероризму настільки значна, що банки стали одним з головних складових елементів цієї системи. З іншого боку, участь у протидії відмиванню злочинних доходів і фінансуванню тероризму стало одним з найважливіших напрямків банківської діяльності.

Багато чого буде залежати від оцінок з боку уповноваженого органу, тому вкрай важливо, щоб у кожному банку була детально розроблена та діяла система спеціальних заходів, яка передбачена законодавством, що включає належну взаємодію із уповноваженим органом влади.

---

<sup>1)</sup> [4] Про затвердження Положення про здійснення банками фінансового моніторингу. Правління Національного банку України. Постанова №189 від 14.05.2003. Зареєстр. в Міністерстві юстиції 20.05.2003, № реєстр. 381/7702.

<sup>2)</sup> [6] Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербургский институт внешнеэкономических связей, экономики и права. Санкт-Петербург: Знание, 2014. 62 с.

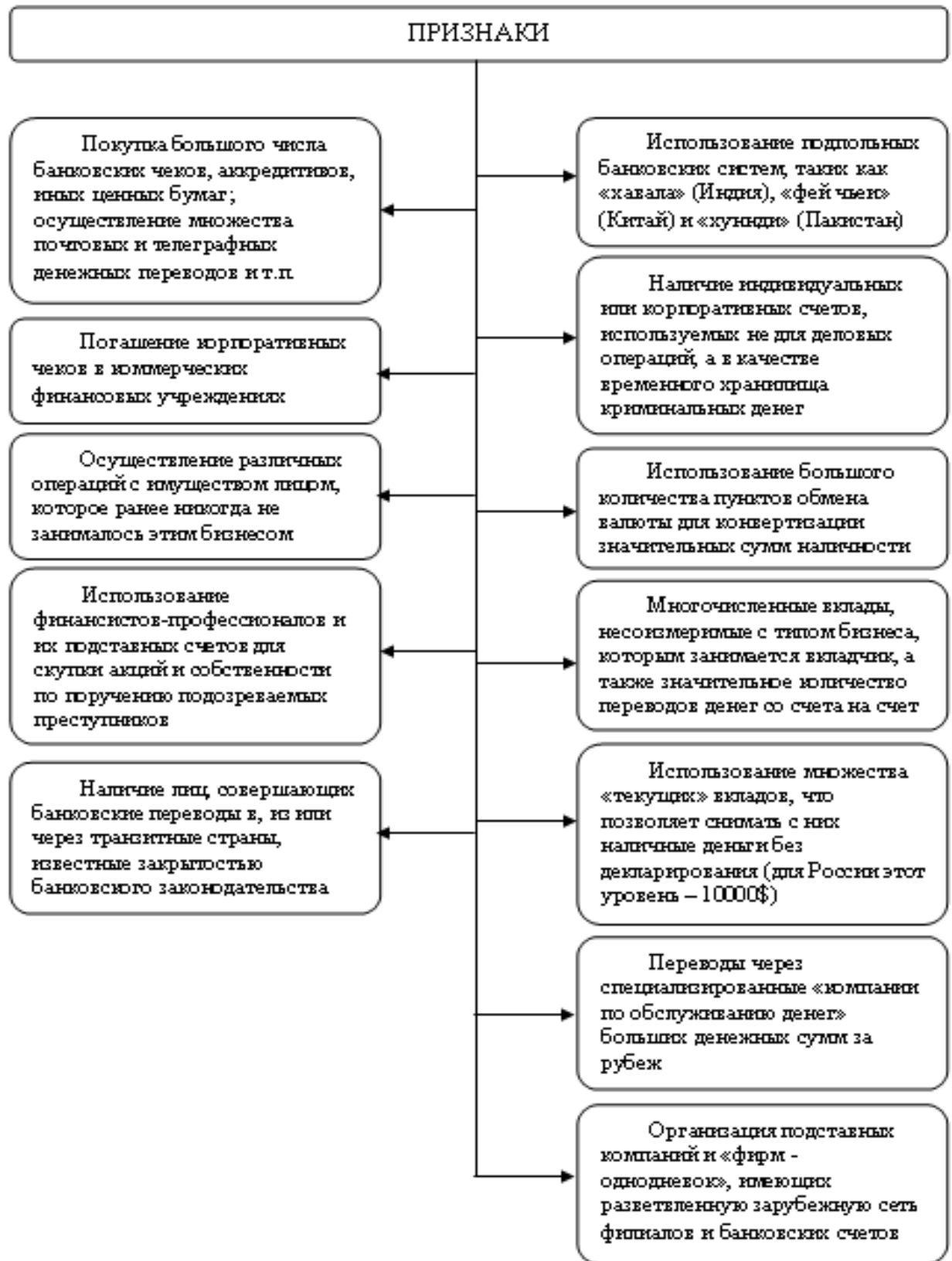


Рисунок 1 – Ознаки легалізації доходів [6] <sup>1)</sup>

<sup>1)</sup> [6] Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербургский институт внешнеэкономических связей, экономики и права. Санкт-Петербург: Знание, 2014. 62 с.

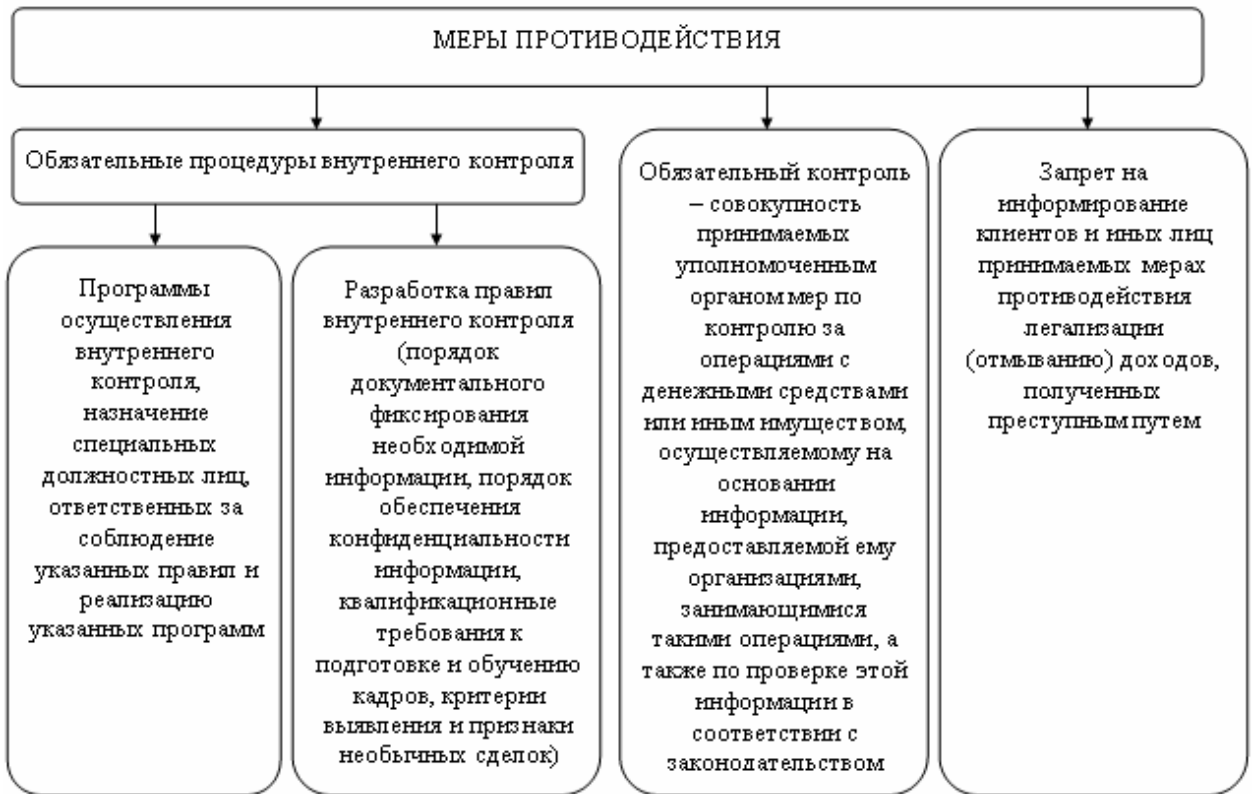


Рисунок 2 – Заходи із протидії відмиванню доходів, отриманих незаконним шляхом [6]<sup>1)</sup>

<sup>1)</sup> [6] Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербургский институт внешнеэкономических связей, экономики и права. Санкт-Петербург: Знание, 2014. 62 с.

## **2 РОЗРОБКА СТРУКТУРНИХ СХЕМ ЗАСТОСУНКІВ**

### **2.1 Розробка заходів з автоматизації фінансового моніторингу у банку**

При розробці таких заходів використовуємо тільки відкриті джерела інформації і керуємося:

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

В першу чергу необхідно систематизувати операції, ознаки яких свідчать про можливу неправову діяльність клієнта банку. Саме на підставі інформації про такі операції ми зможемо виробити заходи по ідентифікації клієнта на основі характеру проведених ним операцій.

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

- а) операції превентивного характеру – з метою запобігання можливих фінансових порушень клієнтами, в надійності яких існують сумніви;
- б) операції, сам склад яких є сумнівним, що вимагає більш ретельного контролю.

Резюмуючи вищевикладене, можна впорядкувати склад пунктів (полів) анкети новостворюваного або обслуговується клієнта банку з точки зору небезпеки можливого здійснення ним операцій з легалізації доходів, одержаних злочинним шляхом і фінансування терористичної діяльності. Дані відомості представлені у табл. 1.

У цій таблиці наведено найменування даних параметрів пошуку, які в подальшому будуть використовуватися при розробці програмного засобу.

На рис. 3 приведена діаграма, яка візуалізує взаємодію розглянутих даних з точки зору джерел інформації про клієнтів.

Таблиця 1 – Анкетні дані клієнта, що підлягають автоматизації фінансового моніторингу

№ з/п	Поле	Позначає:	Спосіб виявлення
1	cl_name	найменування суб'єкту господарювання	Чорний список юридичних осіб
2	promo	засновника	Чорний список фізичних осіб, чорний список юридичних осіб
3	authorised	уповноважену особу	Чорний список фізичних осіб
4	risk	Рівень ризику	Поле анкети (заповнюється експертом)
5	auth_capit	установчий фонд	Чисельне значення поля анкети, що заповнюється вручну на основі установчих документів; контролюється перевищення обсягу операцій за певний період
6	contrag	основні контрагенти	Чорний список фізичних осіб, чорний список юридичних осіб

Операції, склад яких є сумнівним, що вимагає більш ретельного контролю, також можуть бути об'єктом автоматизованого моніторингу, проведеного банком щодо клієнта. Ознаки такого роду операцій розглянуті в наявній літературі [7]<sup>1)</sup> і наведені на рис. 3. Однак, в такому вигляді ці відомості не-

<sup>1)</sup> [7] Невлюдов І. Ш., Тучин О. В., Карпов Г. В. Організаційно-технічні основи виробництва електронних апаратів: Підручник. Харків: ХНУРЕ, 2011. 182 с.



придатні для вирішення завдання автоматизації фінансового моніторингу в частині ідентифікації клієнтів банку.

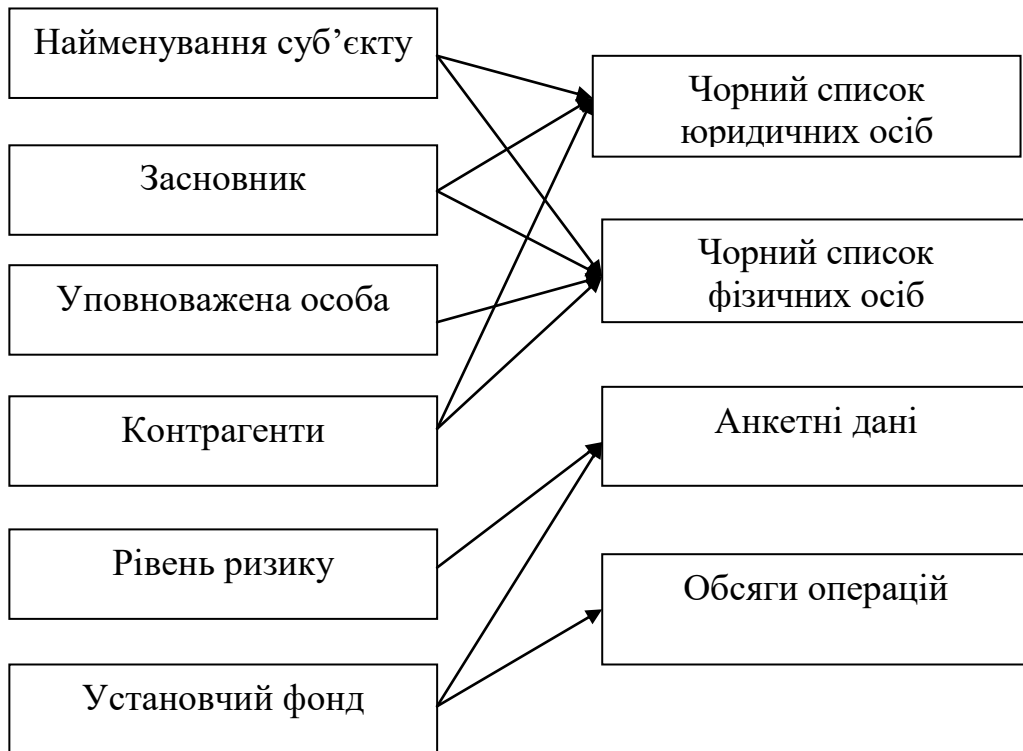


Рисунок 3 – Схема автоматизованого контролю анкетних даних клієнту

Наше завдання – трансформувати наявні дані для того, щоб їх було зручно використовувати.

Табл. 2 резюмує результати нашого дослідження, а рис. 4 показує схему взаємодії при виконанні автоматизованого моніторингу.

Особливістю даної схеми є те, що тут в якості вихідної інформації в будь-якому випадку використовується журнал (реєстр) платежів, що підлягають проводці.

Іншими словами, при виконанні банківських платежів служба фінансового моніторингу повинна «вклинитися» в проміжок часу, коли здійснюється проводка платежів, до того моменту, коли грошові кошти вже будуть перераховані на рахунок одержувача.

Таблиця 2 – Операції клієнта, що підлягають автоматизації фінансового моніторингу

№ з/п	Поле	Позначає:	Спосіб виявлення
1	pay_src	джерело платежу на суму більше 80 тис. грн.	Джерело платежу – не банківський рахунок, сума платежу – понад 80 тис. грн.
2	pay_purp	призначення платежу	Звірка призначення платежу з видами діяльності клієнта. Якщо платіж призначений для оплати електроенергії, комунальних послуг, сплати податків та інших обов'язкових платежів, то контроль не проводиться
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	Найменування одержувача	Зіставлення з чорним списком
5	pay_tgt_ac	Рахунок одержувача	Якщо гроші перераховуються не на рахунок (наприклад, грошові поштові перекази), то це об'єкт моніторингу, у разі, якщо загальна сума понад 25% від статутного фонду



Рисунок 4 – Оперативна інформація, яка використовується для фінансового моніторингу клієнта банку

Залежно від організації банку, така операція може виконуватись із накопиченими платежами, наприклад, кожні 2 години. Дане час може відрізнятися і не є предметом розгляду в нашому проекті. Важливо лише – як часто будуть запускати нашу програму в банку. Суть же роботи даної програми від цього не зміниться.

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

Тут найбільш важливим є:

- а) розробляється програма має використовувати деякі компоненти внутрішньобанківської інформаційної системи, зокрема, анкетні дані клієнта і журнал платежів протягом операційного дня;

- б) під час запуску програми в період перед черговою проводкою зазначених платежів;
- в) в разі, якщо та чи інша операція підлягає фінансовому моніторингу, розробляється програма повинна формувати власний тимчасовий реєстр, що є вихідною інформацією для служби фінансового моніторингу, що веде остаточний аналіз;
- г) розробляється програма покликана інформувати службу фінансового моніторингу, але не виконує функцію прийняття рішення – ця функція виконується відповідальним працівником банку, він же формує звітність, що надається в контролюючі та правоохоронні органи; складання остаточного звіту у завдання не входить.

## 2.2 Структурна схема

Розроблені заходи по автоматизації фінансового моніторингу є основою для розробки структурної схеми програми.

Для розробки структурної схеми необхідно сформулювати:

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

Надалі будемо вважати, що додаток буде мати модульну структуру, кожен модуль буде відповідати виконуваній функції.

Найменування програми – ІКБ (Ідентифікація Клієнта Банку). Ми застосували аббревіатуру, що цілком доречно для такого випадку.

Мета розробки програми – зниження трудомісткості і прискорення робіт по ідентифікації клієнта на предмет можливих дій по легалізації доходів, одержаних злочинним шляхом або фінансування терористичної діяльності.

Завдання, які вирішуються застосунком:

- а) автоматизація обробки інформації при відкритті рахунку клієнтом;
- б) контроль виконання разових операцій, що підлягають фінансовому моніторингу;
- в) контроль виконання повторюваних операцій, що підлягають фінансовому моніторингу;
- г) формування внутрішнього звіту про виконану роботу, припинення операцій, що підлягають фінансовому моніторингу.

Кожна із завдань повинна бути розглянута більш уважно з метою формування списку функцій, які 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. В першу чергу, це завдання має на меті позбавити відповідального працівника банку від тривалої і виснажливої процедури звірки анкетних даних клієнта з різними «чорними списками».

Таким чином, ми можемо скласти перелік функцій, які виконуються даним додатком. Цей перелік буде відповідати програмним модулям. Табл. 3 містить цей перелік із зазначенням позначень і найменувань програмних модулів, а також додатковими поясненнями [8], [9] <sup>1)</sup>.

На рис. 5 приведена структурна схема програмного засобу, вона являє собою послідовність операцій по перевірці: рахунків клієнтів; платежів.

Дана схема передбачає виконання операцій послідовно одна за одною. З головного модуля програми викликаються модулі для обробки рахунків (account) і для обробки платежів (payment). Процедури всередині кожного з модулів також виконуються послідовно.

<sup>1)</sup> [8] Семенец В. В., Кратц Дж, Невлюдов И. Ш., Палагин В. А. Технология межсоединений электронной аппаратуры: Учебник. Харьков: Компания СМІТ, 2005. 432 с.

[9] Новоселов С. П. Охоронні системи та системи відображення інформації: Навч. посібник. Харків: ХНУРЕ, 2007. 252 с.

Таблиця 3 – Функції застосунку

№ з/п	Функція	Відповідає за:	Опис
1	2	3	4
1	leg_check	контроль юридичних осіб	Контроль наявності найменування суб'єкта господарювання, засновників або контрагентів в стоп-аркуші (чорному списку)
2	phys_check	контроль фізичних осіб	Контроль наявності суб'єкта господарювання (фізичної особи), засновників, відповідальних працівників, контрагентів в стоп-аркуші
3	add check	перевірка додаткових даних	Перевірка рівня ризику і наявності рахунків клієнта в цьому ж банку
4	one-off_max	перевірка разового платежу	Перевірка готівкового платежу на суму 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 переводу з рахунку на рахунок однієї юридичної або фізичної особи
7	one-off_tgt	перевірка переказів	Перевірка платежу, при якому одержувачем є не банківський рахунок (на суму понад 25% від статутного фонду)

Кінець таблиці 3

1	2	3	4
8	mon_max	перевірка накопичення платежів	Перевірка платежів на предмет, коли сума платежів за календарний місяць перевищує статутний фонд
9	mon_tgt	перевірка накопичення перекладів	Перевірка платежів на предмет, коли одержувачем є не банківський рахунок, а сума платежів за календарний місяць перевищує 25% від статутного фонду
10	ac_probl_reg	реєстр проблемних рахунків	Ведення реєстру неблагонадійних рахунків для прийняття рішення про відкриття яких потрібна додаткова перевірка
11	op_probl_reg	реєстр проблемних операцій	Ведення реєстру операцій, що підлягають фінансовому моніторингу, маркування операцій, які слід призупинити

Структурна схема, що міститься на рис. 5, показує: підпорядкованість модулів, але не їх послідовність виконання. Послідовність виконання – це окремий предмет розгляду при розробці алгоритму роботи програми у відповідному розділі бакалаврської кваліфікаційної роботи.

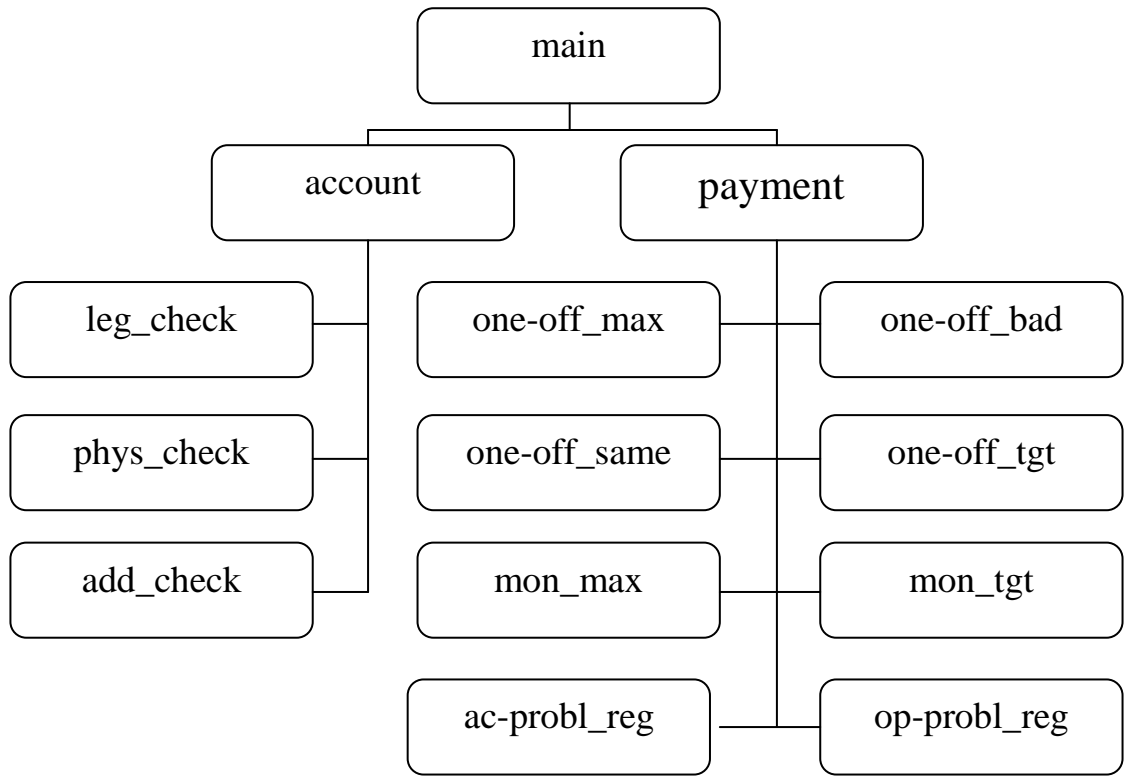


Рисунок 5 – Структурна схема застосунку



## 3 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ БАЗОЮ ДАНИХ

### 3.1 Вибір платформи проектування

Для проектування використовуємо безкоштовну платформу PHP-MYSQL.

Розробникам Web-застосунків немає необхідності говорити, що web-сторінки – це не тільки текст і картинки. Гідний уваги сайт повинен підтримувати деякий рівень інтерактивності з користувачем: пошук інформації, продаж продуктів, конференції тощо. Традиційно все це реалізувалося CGI-скриптами, написаними на Perl. Але CGI-скрипти дуже погано масштабуються. 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 самим схожий на ASP від Microsoft [10] <sup>1)</sup>.

Синтаксис PHP дуже схожий на синтаксис C або Perl. Люди, знайомі з програмуванням, дуже швидко зможуть почати писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях по виділенню / звільненню пам'яті. Програми, написані на PHP, досить легко читаються. Написаний PHP-код легко візуально прочитати і зрозуміти, на відміну від Perl-програм.

MySQL – компактний багатопотоковий сервер баз даних. MySQL характеризується великою швидкістю, стійкістю і легкістю у використанні [11] <sup>2)</sup>. MySQL був розроблений компанією ТсХ для внутрішніх потреб, які полягали у швидкій обробці дуже великих баз даних. Компанія стверджує, що викори-

---

<sup>1)</sup> [10] Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання поведінкової частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 2 (115). С. 117–126. (кат. «Б») DOI: 10.30929/1995-0519.2019.2.117-126.

<sup>2)</sup> [11] Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. *Вісник Кременчуцького національного університету ім. Михайла Остроградського*. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI: 10.30929/1995-0519.2019.3.130-139.

стовує MySQL з 1996 р. на сервері з більш ніж 40 баз даних (БД), які містять 10 тис. таблиць, з яких більш ніж 500 мають більше 7 мільйонів рядків.

MySQL є ідеальним рішенням для малих і середніх додатків [12]<sup>1)</sup>. Вихідні тексти сервера компілюються на безлічі платформ. Найбільш повно можливості сервера проявляються на Unix-серверах, де є підтримка багатопоточності, що дає значний приріст продуктивності. MySQL-сервер є безкоштовним для некомерційного використання. Інакше необхідно придбання ліцензії.

#### Можливості MySQL.

MySQL підтримує мову запитів SQL у стандарті ANSI 92, і крім цього має безліч розширень до цього стандарту, яких немає в жодній іншій системі управління базами даних (СУБД).

#### Короткий перелік можливостей MySQL:

- а) підтримується необмежена кількість користувачів, що одночасно працюють з БД;
- б) кількість рядків в таблицях може досягати 50 млн.;
- в) швидке виконання команд. Можливо MySQL найшвидший сервер з існуючих;
- г) проста і ефективна система безпеки.

Таким чином, ми бачимо, що даний сервер баз даних дуже популярний і дає значні можливості при розробці Web-додатків. Його ми і будемо використовувати у нашому проекті. Для розробки СУБД використовуємо результати попереднього проектування. Для цього визначимо перелік сховищ даних, які використовуються в роботі. До нього відносяться:

- а) стоп-лист (чорний список) юридичних осіб;
- б) стоп-лист фізичних осіб;
- в) анкета клієнта банку;
- г) журнал платежів, що підлягають обробці;

---

<sup>1)</sup> [12] Котеров Д. В. Самоучитель PHP5: Учеб. пособ. Санкт-Петербург: БХВ-Петербург, 2012. 580 с.

- д) журнал клієнтів, що підлягають моніторингу;
- е) журнал операцій, що підлягають моніторингу.

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

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

У цьому випадку вважаємо, що наше завдання – ведення даних, безпосередньо пов'язаних з виявленням клієнтів і операцій, що підлягають фінансовому моніторингу [6]<sup>1)</sup>.

До таких сховищ даних відносяться:

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

Перші два сховища даних мають однакове призначення. Вони зведені окремо тільки тому, що по відношенню до їх існують трохи розрізняються дані. Крім того, до них потрібно звертатися в різних випадках, тому немає сенсу виконувати подвійний обсяг пошуку. Так само ми точно знаємо, коли шукаємо фізичних, а коли юридичних осіб.

Іншим важливим фактором є те, що не вся інформація про цих осіб нас цікавить. Збільшення обсягу інформації про відповідному суб'єктові призведе до уповільнення роботи. Тому доцільно мінімізувати обсяг збережених дані.

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

---

<sup>1)</sup> [6] Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербургский институт внешнеэкономических связей, экономики и права. Санкт-Петербург: Знание, 2014. 62 с.

Дана інформація не є предметом проектування, тому вона буде обговорюватися на стадії розробки програмного коду при складанні SQL-запитів до відповідної БД.

### 3.2 Розробка структури таблиць

Наведені вище дані є вихідною інформацією для розробки структури таблиць, які формуються розробляється СУБД. Єдине, що дійсно вимагає певної мінімізації в нашому випадку – це систематизація даних про заснування занесення в журнал клієнтів або операцій, що підлягають моніторингу.

Справа в тому, що це може бути досить довгий запис, і оператору немає сенсу її вводити повністю. Особливо, якщо, як в нашому випадку, перелік таких підстав обмежений і може бути занесений в окремий реєстр. Даний реєстр може являти собою таблицю із зазначенням коду причини і її змісту.

Тому, у нашому проекті ми будемо будувати роботу саме за таким принципом. Для початку необхідно скласти перелік таблиць, які використовуються у роботі. Табл. 4 містить такий перелік.

Таблиця 4 – Перелік таблиць, що проектуються СУБД

№ з/п	Позначення	Найменування	Пояснення
1	2	3	4
1	cl_leg	Стоп-лист юридичних осіб	Інформація про юридичних осіб, що мають сумнівну репутацію з точки зору можливості легалізації доходів, отриманих злочинним шляхом, або фінансування тероризму

Кінець таблиці 4

1	2	3	4
2	sljph	Стоп-лист фізичних осіб	Інформація про фізичних осіб, що мають сумнівну репутацію з точки зору можливості легалізації доходів, отриманих злочинним шляхом, або фінансування тероризму
3	sljourn	Журнал по клієнтам	Інформація про клієнтів, які збираються відкрити або мають рахунок у банку, діяльність або анкетні дані про яких викликають сумнів
4	orjourn	Журнал за операціями	Інформація про операції, що підлягають фінансовому моніторингу
5	reas_reg	Реєстр причин	Внутрішній класифікатор підстав (причин), за якими зведення про клієнта або операцію повинні бути віднесені до сумнівних та занесені у відповідний журнал (реєстр)

Зараз необхідно скласти структуру кожної з таблиць. Табл. 5 – 9 містять перелік полів, що використовуються у спроектованій СУБД.

Таблиця 5 – 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	Юридичний адрес	char(30)	
	leg_country	Країна	char(15)	
	reas	Підстава	char(2)	+

Таблиця 6 – 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	Підстава	char(2)	+

Таблиця 7 – Структура таблиці cl\_ourn

Ключ (key)	Позначення	Найменування	Тип	Обов'язковість заповнення
*	cl_id	Ідентифікатор	int auto_increment	+
	cl_name	Прізвище, ім'я, по-батькові або найменування	char(40)	
	cl_ac_num	Номер рахунку	char(20)	
	cl_reg_num	Реєстраційний номер	char(15)	
	cl_date	Дата	date	+
	reas	Підстава	char(2)	+

Таблиця 8 – Структура таблиці op\_journ

Ключ (key)	Позначення	Найменування	Тип	Обов'язковість заповнення
1	2	3	4	5
*	op_id	Ідентифікатор операції	int	+
	op_date	Дата	date	+
	op_src_name	Відправник	char(40)	
	op_src_ac	Номер рахунку відправника	char(20)	
	op_src_mfo	МФО банку відправника	char(10)	
	op_tgt_name	Одержувач	char(40)	
	op_tgt_ac	Номер рахунку одержувача	char(20)	

Кінець таблиці 8

1	2	3	4	5
	op_tgt_mfo	МФО банку одержувача	char(10)	
	op_pay	Вид розрахунку	enum('cash', 'check', 'noncash')	+
	op_jpurp	Призначення	char(50)	
	op_sum	Сума	float(8,2)	+
	reas	Підстава	char(2)	+

Таблиця 9 – Структура таблиці reas\_reg

Ключ (key)	Позначення	Найменування	Тип	Обов'язковість заповнення
*	reas	Ідентифікатор причини (основи)	char (2)	+
	reas_name	Найменування причини (основи)	char(15)	+
	reas_explain	Детальний опис причини (підстави)	varchar(250)	

У таблиці зі стоп-листом додані поля ідентифікатора. Це обумовлено необхідністю створення первинного ключа таблиць. З огляду на те, що не завжди все поля будуть заповнені, ми потребуємо такого ідентифікатора. Іншою причиною є можливе схожість найменувань юридичних осіб або прізвищ фізичних осіб. Тоді наявність ідентифікатора також усуне можливу плутанину.



Також нам необхідно скласти перелік таблиць і полів, які не є об'єктом заповнення нашою програмою. У той же час, нам доведеться ними користуватися для організації нормальної роботи СУБД. У загальному 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 БД банку – досить буде уточнити імена БД, таблиць і полів, що використовуються в реально існуючих СУБД [13], [14] <sup>1)</sup>.

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

- а) анкети клієнтів;
- б) журнал (реєстр) платежів, що підлягають проводці.

Нехай таблиці з анкетними даними клієнта називається `leg_dossier`, `ph_dossier`, а таблиця з платежами називається `payment`. Складемо перелік полів таблиць і їх формати, які ми будемо використовувати в роботі. Цей перелік буде однозначно неповний, але це нас не повинно турбувати, тому що виконання запиту з вибіркою тільки деяких даних з таблиці – не проблема – засоби MySQL дозволяють робити це досить легко і швидко.

Як ми бачимо, дані, що використовуються в додатку, в основному збігаються із зовнішніми даними. Це тим більш дивно, що ми розробляємо систему моніторингу, завданням якої є спостереження за певними процесами, що відбуваються в банку. Наше завдання – виявити інформацію, що потребує підвищеної уваги і занести її до реєстру з відповідними поясненнями. На рис. 6 приведена структура БД, що використовується у проекті.

---

<sup>1)</sup> [13] 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. Пост. № 233 від 04.06.2013. Зареєстр. в Міністерстві юстиції 09.06.2013, № реєстр. 457/7778.

[14] Про затвердження «Правил зберігання, захисту, використання та розкриття банківської таємниці». Правління Національного банку України. Пост. №267 від 14.07.2006. Зареєстр. в Міністерстві юстиції 03.08.2006, № реєстр. 935/12809.

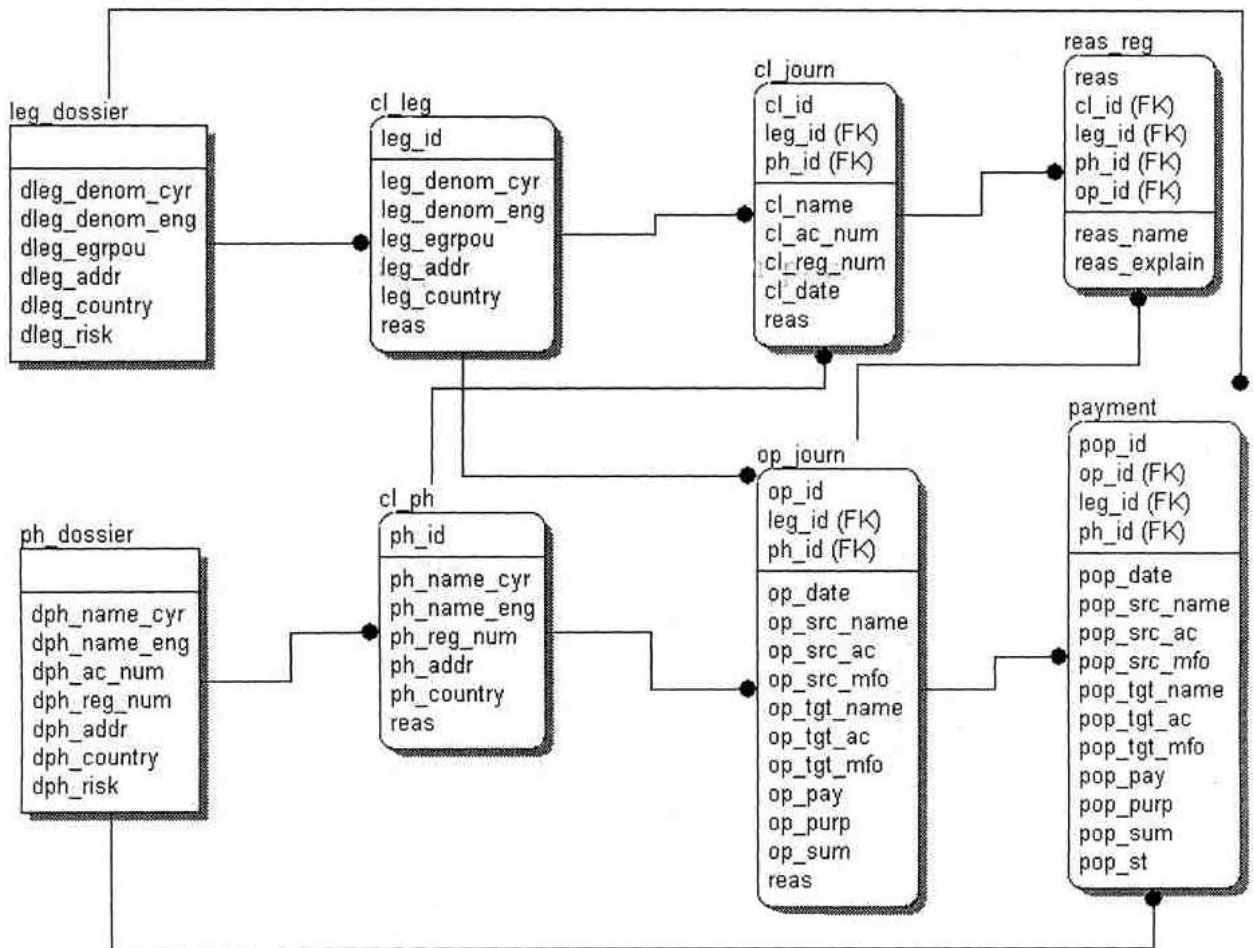


Рисунок 6 – Структура бази даних проекту

У структуру включені не тільки створювані у проекті таблиці, але також і зовнішні таблиці, що використовуються у роботі програми.

Подана структура згенерована за допомогою CASE-засобу ErWin [15] – [18]<sup>1)</sup> та буде використовуватися для розробки програмного коду роботи.

<sup>1)</sup> [14] Lavagno L., Martin G., Selic B.V. UML for Real: Design of Embedded Real-Time. Systems Kluwer Academic Publishers, 2003. 388 p.

[15] Иванов Д. Ю., Новиков Ф. А. Унифицированный язык моделирования UML. Учебное пособие. Санкт-Петербург: Изд-во Политехн. ун-та, 2010. 249 с.

[16] Hay D. C. UML and Data Modeling: A Reconciliation Technics Publications, 2011. 242 p.

[17] Бабич А. В. UML. Первое знакомство. Пособие для подготовки к сдаче теста УМО-100 (OMG Certified UML Professional Fundamental) (+CD-ROM). Санкт-Петербург: Бинум. Лаборатория базовых знаний, 2008. 176 с.

[18] Ларман К. Применение UML и шаблонов проектирования. 2-е изд.: пер. с англ. Москва: Издательский дом «Вильямс», 2004. 624 с.

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

### 4.1 Розробка SQL-запитів для обробки таблиць БД

Для роботи з БД використовуємо мову структурованих запитів (Structured Query Language – SQL). Як сервер баз даних застосовуємо MYSQL. Для локальної розробки програми використовуємо пакет Denwer. Даний пакет написаний на мові PERL і повністю емулює файлову систему UNIX, створюючи віртуальний ресурс на мережевому диску (якщо використовується операційна система (ОС) Windows). Це дозволяє здійснювати розробку додатків на локальній машині без установки і настройки Web і SQL-серверів.

Перший етап роботи: створення бази даних. Ім'я бази даних – 1KB (транслітерація найменування додатки, названого раніше, при розробці структурної схеми). Для цього спочатку виробляємо запуск `mysql` з командного рядка:

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

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

Структура запиту для створення БД:

```
create database ikd;
```

Наступним етапом є створення таблиць всередині бази даних. Для цього спочатку виконується команда: «`use ikd`»;

Після цього виконується запит на створення таблиць. Структура запиту є однотипною, тому не необхідності показувати кожен з них. Складемо тільки 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),
opjgtjnfo 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;

```

Даний запит створює таблицю з параметрами полів, записаними раніше у даному проекті.

Наступний основоположний 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, в стоп-аркуші, то потрібно спочатку прочитати його ім'я з таблиці ph\_dossier і привласнити його який-небудь змінної, наприклад, «name». Потім використовувати змінну «name» як фільтр запиту з таблиці cl\_ph (стоп-лист фізичних осіб). При цьому використовується запит типу:

```

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

```

В цьому випадку, якщо є таке ім'я або йому подібне в стоп-аркуші, то необхідно додаткова перевірка. В принципі, цей запит може бути спрощений:

```

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

```

Однак в цьому випадку система буде перевіряти імена тільки на їх повний збіг. Досить зробити помилку введення хоч на один символ (наприклад, пропуск в кінці імені), як такий запит нічого не дасть. Що вибрати конкрет-

но – питання вибору керівництва або відповідальних працівників банку. Ми будемо використовувати перший варіант як більш надійний, хоча ймовірність помилки при цьому все-таки є. У той же час ми повинні пам'ятати, що розробляється програма повинна інформувати відповідального працівника банку про можливі незаконні операції, отже, саме він і уточнить вже всі деталі вручну.

Наступний тип запити, який нам належить використовувати додавання записів в таблицю. Наприклад, якщо ми встановили що черговий рахунок, відкривається клієнтом, занесеним в 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 підсумку в таблицю БД буде додана відповідна запис. Якщо ми маємо в своєму розпорядженні неповним набором даних, то можна задати значення змінних за замовчуванням (наприклад, null) і використовувати їх для введення в таблицю полів, які в даному конкретному випадку не визначені.

Останній тип запити, який необхідно буде використовувати, це оновлення даних в існуючій таблиці. Наприклад, якщо ми хочемо змінити атрибут поля pop\_st таблиці payment на stop, що означає припинення виконання операції з ідентифікатором op\_jnumber, то ми повинні 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 управління БД.

## 4.2 Розробка інтерфейсу програмного засобу

Згідно з завданням на бакалаврську роботу, ми повинні розробити застосунок, заснований на Web-технології. Це означає, що ми повинні використовувати прийоми Інтернет-програмування для локальних комп'ютерних мереж, прикладом якої є мережа банку.

Для того, щоб почати проектування, потрібно розробити дизайн типової форми додатка. Незважаючи на те, що цей додаток не призначений для розміщення в Інтернет, проте певний дизайн все-таки потрібен. Головне – зручність і швидкість доступу до функцій, що реалізуються даними додатком.

З іншого боку, немає сенсу робити різні форми з різним дизайном, отже, ми повинні розробити типову форму для додатка. Вибираємо наступну структуру: вгорі маємо «шапку» форми з найменуванням застосунку і головним меню. Ця «шапка» може підключатися до будь-якої форми і не вимагає її постійного перезапису, тому ми говоримо про заголовок Веб-форми (додаток А.1).

Даний код, крім того, містить ще й головне меню. Воно реалізовано у табличній формі і досить зручно для рівномірного розподілу по всій ширині форми.

Нижня частина форми є таблицею, в якій ліва колонка однакова для всіх і містить повторення головного меню – список всіх базових операцій, які виконуються за допомогою цього додатка. У правій, основній частині даної таблиці буде міститися головна інформація, що стосується кожної конкретної підзадачі, розв'язуваної в рамках проекрованої інформаційно-аналітичної системи.

На головній сторінці програми в цій частині розташовується знову-таки перелік гіперпосилань, відповідних окремих функцій, що виконуються розробленим застосунком (додаток А.2).

Зовнішній вигляд головної сторінки додатка наведено на рис. 7. Надалі ми будемо розглядати тільки праву нижню частину форми, так як інша частина у всіх вікнах буде однаковою, причому для цього не пишеться якийсь великий програмний код. Підключення заголовка Веб-форми проводиться шляхом виконання команди:

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

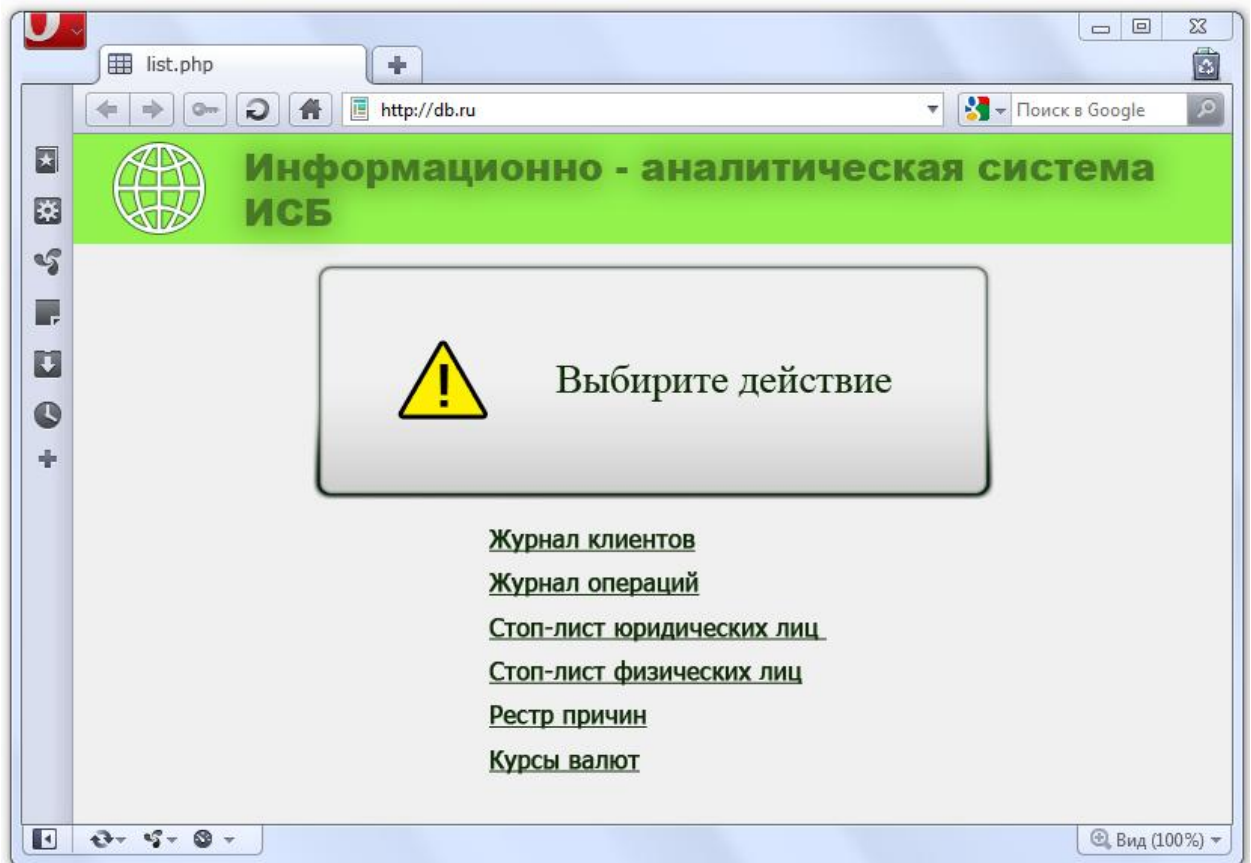


Рисунок 7 – Головна сторінка застосунку

### 4.3 Розробка підсистеми інформаційного забезпечення застосунку

В даному випадку під поняттям «підсистема інформаційного забезпечення» розуміємо ту частину програми, яка виконує функції введення і коригування даних, що використовуються в додатку. Таким даними є:

- а) стоп-лист юридичних осіб;

б) стоп-лист фізичних осіб;

в) реєстр причин необхідності проведення фінансового моніторингу;

Розробку програмного коду розглянемо на прикладі ведення стоп-листа юридичних осіб (раніше в проекті неодноразово застосовувався синонім даного терміну – «чорний список» юридичних осіб).

Для ведення даного списку необхідно не тільки супроводжувати таблицю з стоп-листом, але також в таблицю з реєстром можливих причин занесення в даний список. Це пов'язано з тим, що в стоп-лист поміщається тільки 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 до стоп-листа показано на рис. 8.

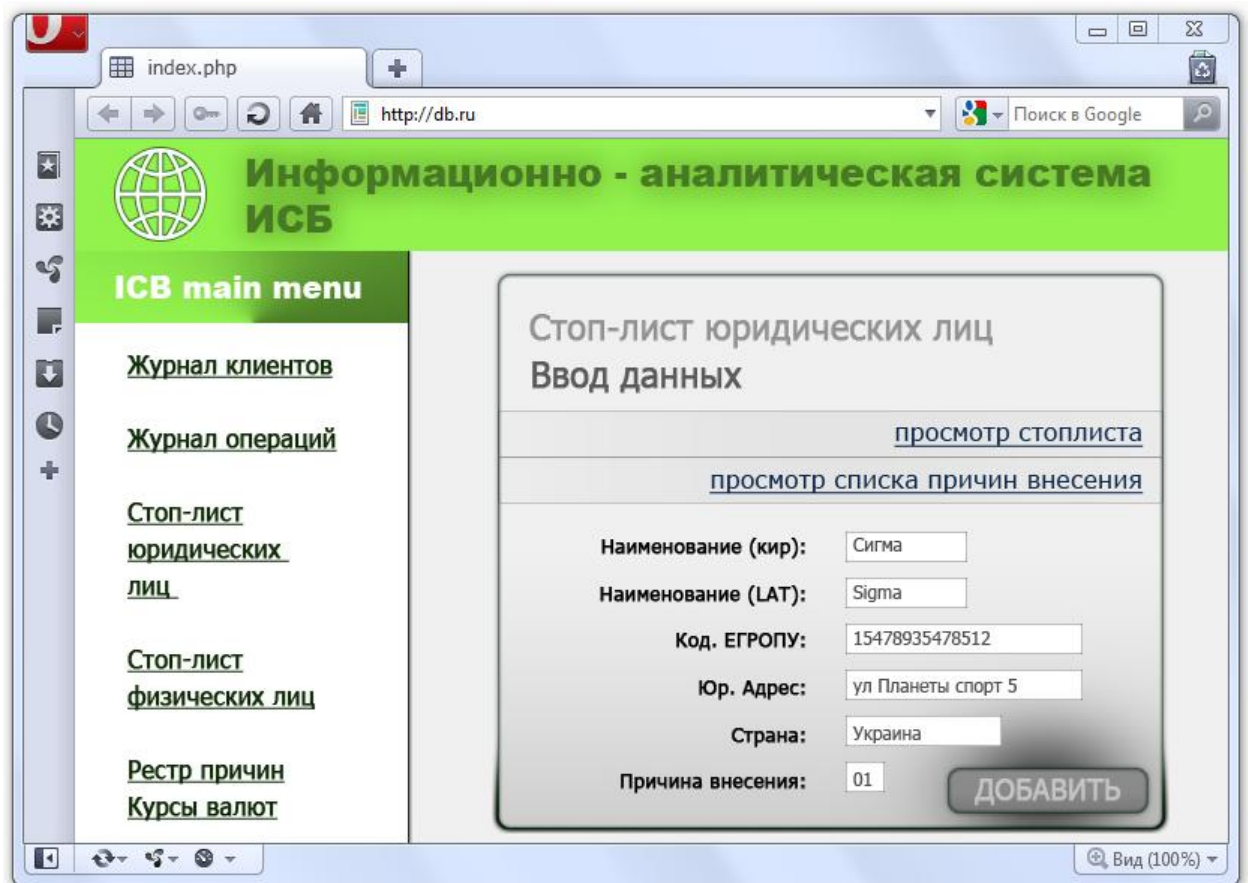


Рисунок 8 – Вікно введення даних до стоп-листа



Фрагмент програмного коду, який реалізує даний діалог, подано у додатку А.3.

Тут і далі ми будемо приводити тільки вміст основного осередку форми, для того, щоб уникнути повторення програмного коду, що зустрічається у всіх екранних формах. Фрагмент реєстру причин внесення в стоп-лист наведено на рис. 9. Тут здійснюється виведення даних у вигляді таблиці. Використовуючи відповідне посилання на сторінці з введенням інформації до стоп-листа, можна ознайомитися з кодами причин і використовувати їх для внесення до стоп-листа.

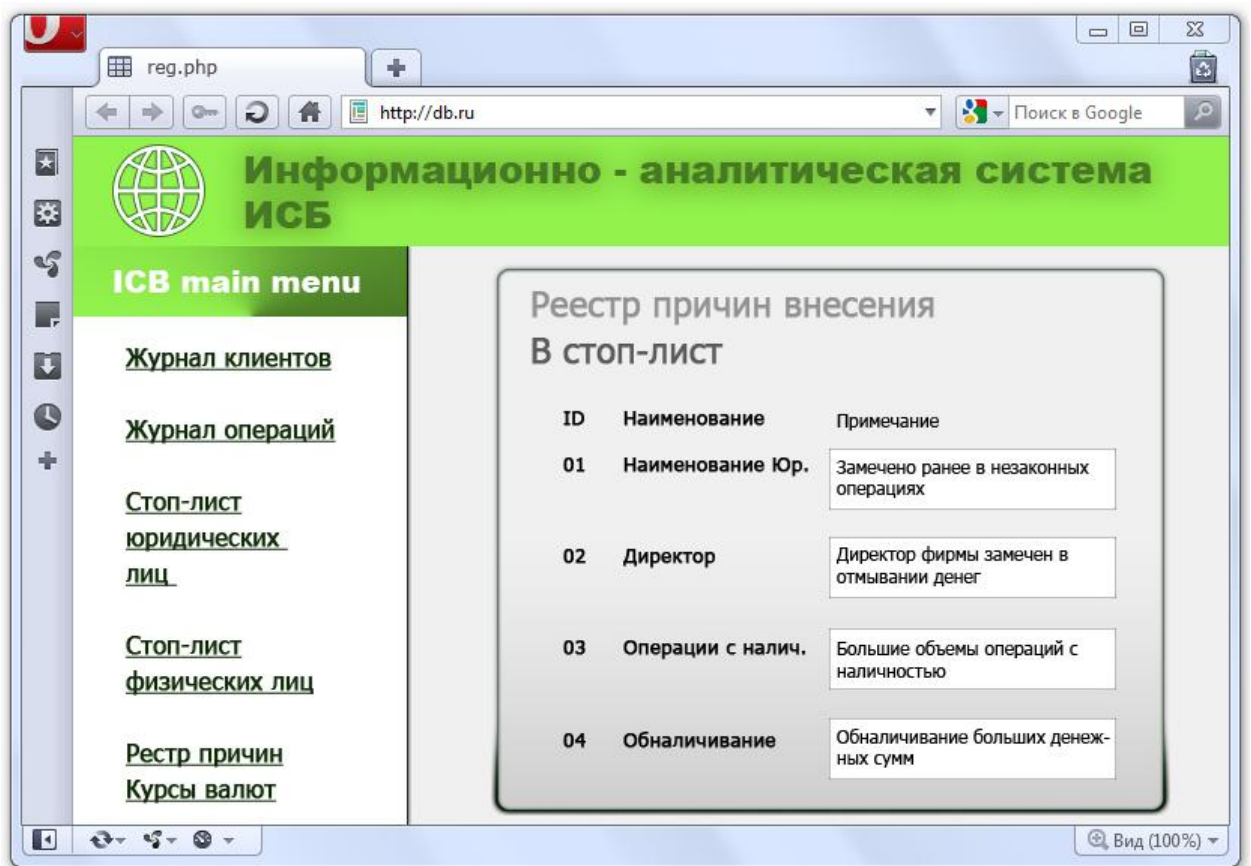


Рисунок 9 – Приклад перегляду реєстру причин внесення до стоп-листа

Тут ми вперше застосовуємо мову запитів для внесення змін у БД. Розглянемо це на прикладі файлу `reas_reg_add.php`, що реалізує введення даних,

які були попередньо сформовані за допомогою html-тега form у файлі reas\_reg.php (додаток А.4).

Тут все починається з блоку даних, які задають параметри підключення до БД. Ці дані можуть задаватися в окремому файлі і підключатися за допомогою команди include. В даному випадку для простоти і наочності ми це робимо прямо у файлі. Після цього слід ланцюжок команд, що дозволяють підключитися до БД. Це команди mysql\_connect і mysql\_select\_db. Після запуску цих команд ми готові працювати безпосередньо з таблицями БД.

Наступна ключова команда – mysql\_query. Вона дозволяє виконати sql-запит і привласнити результат його виконання будь-якої змінної. Згодом, застосовуючи команду mysql\_fetch\_row, ми перетворимо результат виконання запиту у масив даних, який потім може бути в циклі виведений в будь-яке місце, яке ми встановимо (додаток А.5). У нашому випадку все це виводиться до html-таблиці.

Тут і надалі буде використовуватися така послідовність дій при виконанні операцій обробки даних, тому більше ми не будемо акцентувати на цьому увагу. Приклад результату перегляду стоп-листа наведено на рис. 10.

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

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

З реєстру причин на підставі коду причини занесення до реєстру, витягується найменування даної причини і поміщається в відповідному полі екранної таблиці. Для цього служить змінна \$q2, яка використовується для створення другого запиту, причому виконання даного запиту поміщається всередину циклу порядкового зчитування записів зі стоп-листа, таким чином, що кожен раз, як чергова запис виводиться в таблицю, з реєстру причин витягується 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 детально розглядати.

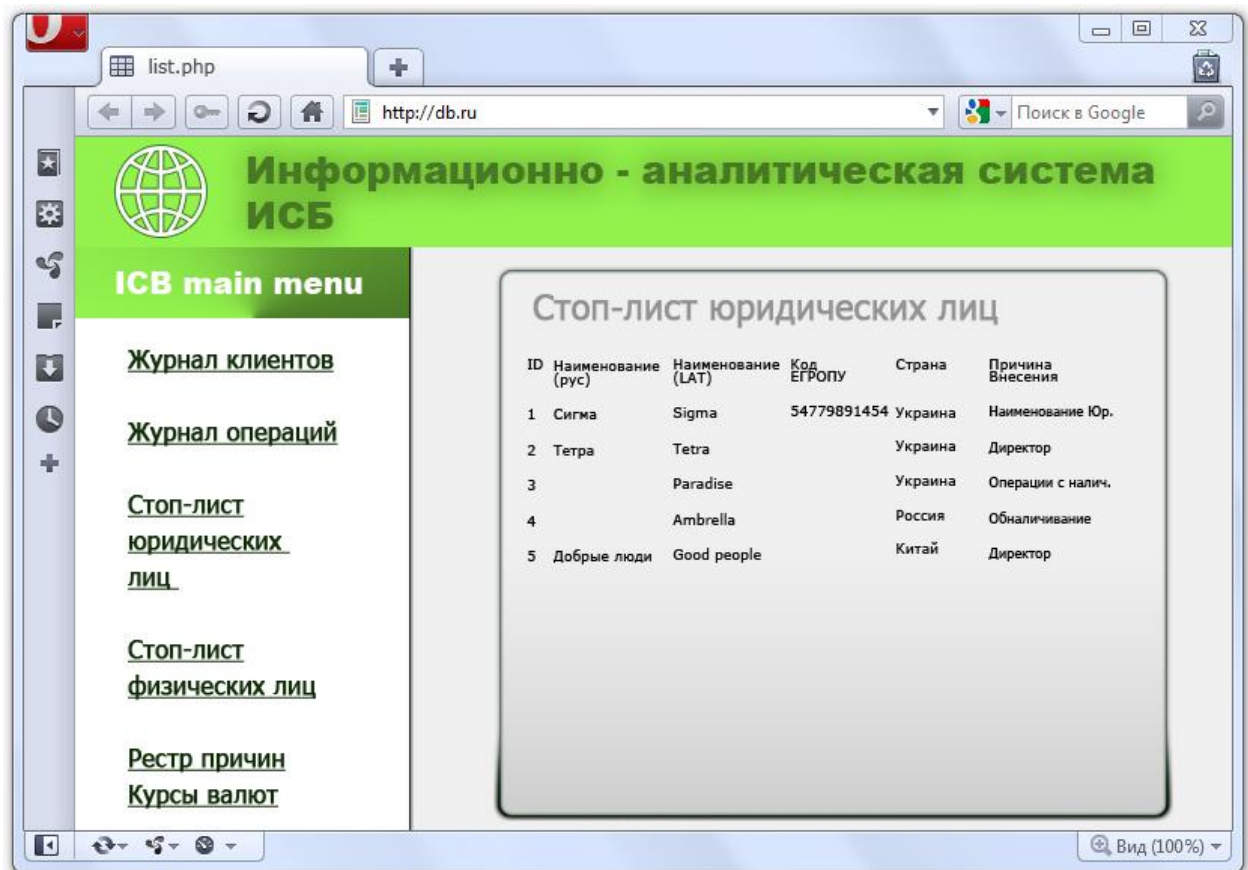


Рисунок 10 – Приклад результату перегляду стоп-листа юридичних осіб

#### 4.4 Аналіз вхідних даних за допомогою автоматизованої системи

Дуже важлива частина розробки – аналіз даних, отриманих з інших джерел – анкет клієнтів, які відкривають або мають рахунки в банку, журналу платежів на предмет виявлення можливих причин фінансового моніторингу. Тут ми розглянемо дані операції на прикладі аналізу анкетних даних клієнтів. Ми не будемо повністю повторювати код програмного засобу, так як зразки складання такого коду наведені раніше.

Наше завдання – алгоритм обробки даних і його реалізація у вигляді php-коду.

Для того, щоб виконати таку обробку, потрібно витягти дані з таблиці з анкетами клієнтів і виконати послідовну перевірку даних на предмет виявлення можливих причин внесення до реєстру підозрілих клієнтів.

Для цього ми використовуємо анкетні дані:

- а) dleg\_denom\_\_cyr: найменування (кирилиця);
- б) dleg\_denom\_eng: найменування (латиниця);
- в) dleg\_egrou: код ЄДРПОУ;
- г) dleg\_addr: юридична адреса;
- д) dleg\_country: країна.

Тут нас цікавлять найменування даного суб'єкта господарювання. Для того, щоб перевірити його надійність, ми повинні по черзі порівняти обидві назви з усіма записами у стоп-аркуші юридичних осіб.

Алгоритм роботи наступний:

- а) прочитати перший запис з таблиці анкет;
- б) прочитати перший запис з стоп-листа;
- в) порівняти попарно латинське і російське найменування аналізованого юридичної особи з тим, яке лічено з стоп-листа:
  - 1) якщо збіг відсутній, то перейти до п. г);
  - 2) якщо хоч одне з найменувань співпало, то занести дані юридичної особи у журнал клієнтів та змінити поле «ступінь ризику» анкети на high (висока);
- г) визначити такий запис із стоп-листа;
- д) якщо запис ще є, то перейти до п. в);
- е) прочитати такий запис з таблиці анкет;
- ж) якщо запис є, перейти до п. б);
- и) алгоритм завершений.

Даний алгоритм чітко показує два вкладених циклу:

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

Неважко припустити, що для цього достатньо використовувати запити «select» Назвемо запит даних з анкет «anc» і запит зі стоп-листа юридичних осіб – «lsl».

Код запиту з анкет наступний:

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

Результат виконання даного запиту – перелік змінних, зазначених раніше з префіксом «leg». Пошук збігів із записами стоп-листа організуємо за принципом фільтрації даних з 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%')");
```

Даний запит передбачає можливість контролю імен навіть при їх неповного збігу (досить, щоб збіглося початок). Якщо будуть знайдені клієнти, які потрапили під фільтр випадково, то відповідальний працівник банку це відстежить. При бажанні структуру запиту можна поміняти на перевірку повного збігу.

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

```
if (mysql_num_rows($lsl)>0)  
{  
}
```

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

Якщо кількість збігів більше нуля, то необхідно інформацію про клієнта занести до відповідного журналу (таблиця cl\_journ), також змінити атрибут поля «ступінь ризику» даного клієнта (додаток А.6).

В результаті буде змінений атрибут «ступінь ризику» в анкеті клієнта і внесено запис до журналу реєстрації підозрілих клієнтів, який буде потім використаний відповідальним працівником банку.

Дана частина програми не має свого інтерфейсу, так як вона виконується автоматично. Для перегляду результатів роботи програми необхідно скористатися типовий процедурою перегляду відповідного ресурсу БД.

#### **4.5 Тестування програмного забезпечення**

Вже зазначалося, що для забезпечення нормальної роботи програми використовується інформація, яка формується іншими службами банку. Для того, щоб перевірити правильність роботи розробленого додатка, необхідно розробити власну тестове програмне забезпечення.

Воно передбачає ведення трьох таблиць:

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

У БД ці ресурси представлені таблицями «leg\_dossier», «ph\_dossier» та «raument» відповідно. Приклад Веб форми по введенню таких даних наведено на рис. 11.

Як ми бачимо, ніякої відмінності від розглянутих раніше методів проектування немає. Єдина різниця полягає в тому, що тут 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 також зазначити, що в даному проекті розглядалася в першу чергу модель, використовувана для ідентифікації клієнтів та платежів в банку.

Що стосується питань безпеки програмного забезпечення, то це може розглядатися як наступна стадія роботи за умови впровадження застосунку. Справа в тому, що в будь-якому банку існують свої правила забезпечення

безпеки програмного забезпечення, і вони є секретною інформацією, тому у даному проєкті вони і не розглядалися.

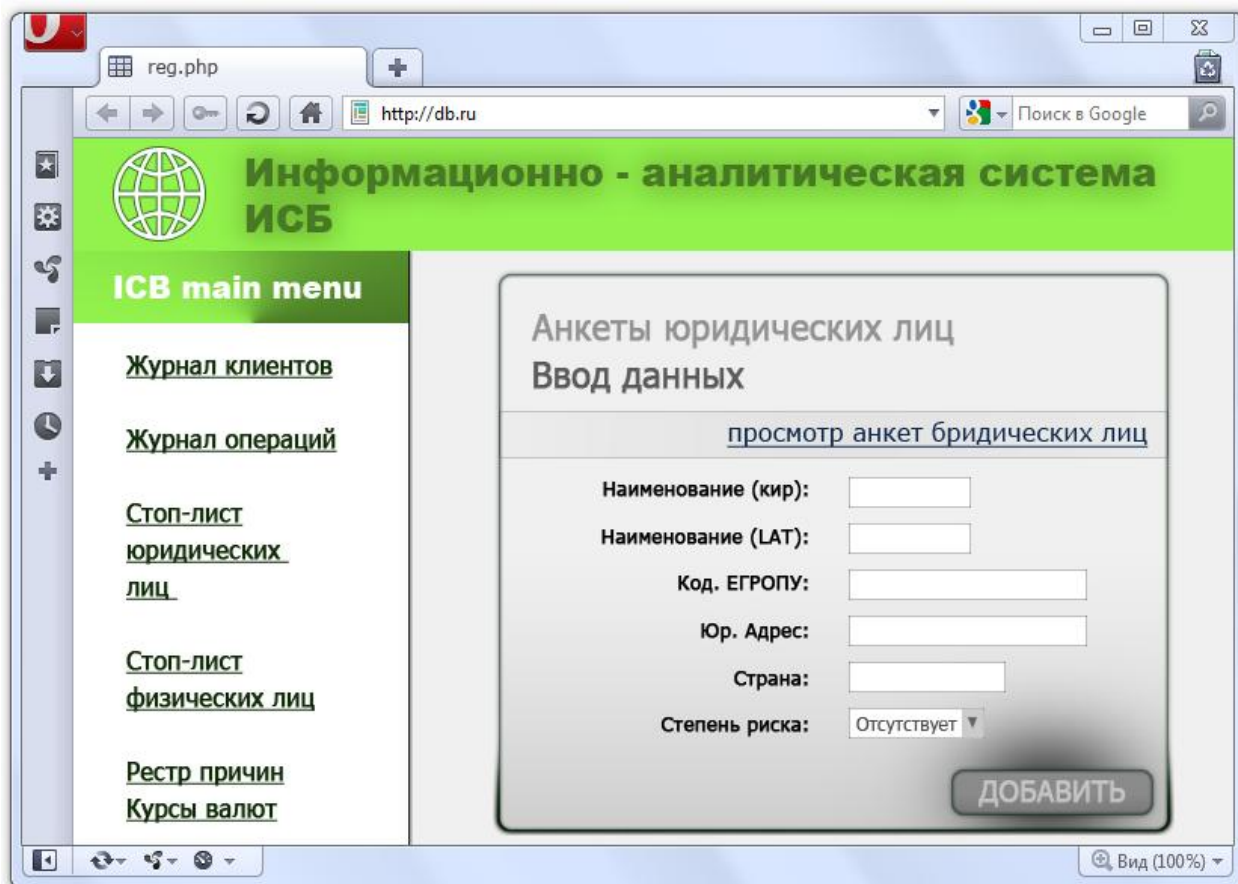


Рисунок 11 – Приклад форми уведення даних у тестовому режимі програмного забезпечення

## ВИСНОВКИ

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

- а) ідентифікація за анкетними даними;
- б) ідентифікація по проведеним операціям.

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

При виконанні проекту отримані наступні результати:

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

Для роботи застосунку обрано такі компоненти програмного забезпечення:

- а) Web-сервер – Apache;
- б) сервер баз даних – MySQL;
- в) мова програмування – PHP.



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

Для локальної розробки застосунку використаний емулятор пакета Apache-PHP-MySQL – додаток «Denwer».

Для складання структури бази даних використовується CASE-засіб ERWin.

Розроблений застосунок є крос-платформним та може працювати під різними операційними системами.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Великодний С. С. Методи реінжинірингу програмних систем. Технології приборостроєння. 2014. Спец. вып. С. 65–68.
2. Великодний С. С. Проблема реінжиніринга видів забезпечення систем автоматизованого проектування. Управляючі системи і машини. 2014. № 1. С. 57–61, 76. Видання національної академії наук України.
3. Velykodniy S. S., Burlachenko Zh. V., Zaitseva-Velykodna S. S. Method of presenting the assessment for reengineering of software systems with the project coefficients help. Інформаційні технології в моделюванні: матер. IV-ої всеукр. наук.-практ. конф. студ., аспірантів та мол. вчених, 21–22 березня 2019 р., м. Миколаїв. Миколаїв: МНУ ім. В. О. Сухомлинського, 2019. С. 10–11.
4. Про затвердження Положення про здійснення банками фінансового моніторингу. Правління Національного банку України. Постанова №189 від 14.05.2003. Зареєстр. в Міністерстві юстиції 20.05.2003, № реєстр. 381/7702.
5. Великодний С. С., Тимофєєва О. С., Зайцева-Великодна С. С., Нямцу К. Є. Порівняльний аналіз властивостей відкритого, вільного та комерційного програмного забезпечення. Інформаційні технології та комп'ютерна інженерія. 2018. № 1 (41). С. 21–27.
6. Литвиненко А. Н., Ковалева Е. Ю. Легализация (отмывание) доходов, полученных преступным путем. Учеб. пособ. в схемах. Санкт-Петербургский институт внешнеэкономических связей, экономики и права. Санкт-Петербург: Знание, 2014. 62 с.
7. Невлюдов І. Ш., Тучин О. В., Карпов Г. В. Організаційно-технічні основи виробництва електронних апаратів: Підручник. Харків: ХНУРЕ, 2011. 182 с.

8. Семенец В. В., Кратц Дж, Невлюдов И. Ш., Палагин В. А. Технология межсоединений электронной аппаратуры: Учебник. Харьков: Компания СМІТ, 2005. 432 с.
9. Новоселов С. П. Охоронні системи та системи відображення інформації: Навч. посібник. Харків: ХНУРЕ, 2007. 252 с.
10. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання поведінкової частини. Вісник Кременчуцького національного університету ім. Михайла Остроградського. 2019. Вип. 2 (115). С. 117–126. (кат. «Б») DOI: 10.30929/1995-0519.2019.2.117-126.
11. Великодний С. С., Бурлаченко Ж. В., Зайцева-Великодна С. С. Реінжиніринг графічних баз даних у середовищі відкритої системи автоматизованого проектування BRL-CAD. Моделювання структурної частини. Вісник Кременчуцького національного університету ім. Михайла Остроградського. 2019. Вип. 3 (116). С. 130–139. (кат. «Б») DOI: 10.30929/1995-0519.2019.3.130-139.
12. Котеров Д. В. Самоучитель PHP5: Учеб. пособ. Санкт-Петербург: БХВ-Петербург, 2012. 580 с.
13. 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. Пост. № 233 від 04.06.2013. Зареєстр. в Міністерстві юстиції 09.06.2013, № реєстр. 457/7778.
14. Про затвердження «Правил зберігання, захисту, використання та розкриття банківської таємниці». Правління Національного банку України. Пост. №267 від 14.07.2006. Зареєстр. в Міністерстві юстиції 03.08.2006, № реєстр. 935/12809.
15. Lavagno L., Martin G., Selic B.V. UML for Real: Design of Embedded Real-Time. Systems Kluwer Academic Publishers, 2003. 388 p.

16. Иванов Д. Ю., Новиков Ф. А. Унифицированный язык моделирования UML. Учебное пособие. Санкт-Петербург: Изд-во Политехн. ун-та, 2010. 249 с.
17. Hay D. C. UML and Data Modeling: A Reconciliation Technics Publications, 2011. 242 p.
18. Бабич А. В. UML. Первое знакомство. Пособие для подготовки к сдаче теста UMO-100 (OMG Certified UML Professional Fundamental) (+CD-ROM). Санкт-Петербург: Бином. Лаборатория базовых знаний, 2008. 176 с.
19. Ларман К. Применение UML и шаблонов проектирования. 2-е изд.: пер. с англ. Москва: Издательский дом «Вильямс», 2004. 624 с.

## ДОДАТОК А

### Фрагменти програмного коду виконаного застосунка

#### А.1 Програмний код, який реалізує заголовок

```

<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"х/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>

```

## А.2 Код головної сторінки програми

```

<html><body>
<?
include ('head.php');
?>
  <table width= 100% border=
    1 > <tr>
      <td width=20% height=500><font size="+2" color="blue">
        <br>
        <H1          ALIGN="CENTER"><FONT          FACE=GEORGIA
COLOR="GREEN" size="+2">
        echo "ICB Main Menu"
        </FONT>
        </H1>
        <A href="index.php">Главная</a> <br><br>
        <A href="cl_journ.php"> Журнал клиентов </a> <br><br>
        <A href = "op_journ.php ">Журнал операций </a> <br><br>
        <A href="cl__leg.php"> Стоп-лист<br>юридических<br>лиц
</a><br><br>
        <A href="cl_ph.php">
Стоп-
лист<br>физических<br>лиц</a><br><br>
        <a href="reas_reg.php">Реестр причин</a><br><br>
        href="currency.php">Курсы валют</a>< br >
        </td>
        <td width=67% border=0 bgcolor="lightgrey"
        align=center> <font size="+7" color="red">Выберите
действие</ font > <font size+=5 color="blue">
        <br><br>
        < A href = " cl_journ.php ">журнал клиентов </a> <br>
        <A href="op_journ.php">журнал операций </a> <br>
        <A href="cl_leg.php"> Стоп-лист юридических лиц</a>< br >
        <A href="cl_ph.php"> Стоп-лист физических лиц</a>< br >
        <a href="reas_jreg.php">Реестр причин</a>< br >
        <a href="currency.php">Курсы валют</a>< br >
        </font>
        </td>
      </tr><br>
    </font></td></tr></table>
</body></html>

```

### А.3 Программний код для прикладу реалізації вікна введення даних до стоп-листа

```

<font size="+7" color="red">Стоп-лист юридических лиц</ font
>
<font size="+5 color="blue">
<br><br>Ввод данных</font>
<br><font size="+2>
<a
href="cl_leg_view.php">Просмотр Стоп-
листа</a>< br >< br >
<a href="reas_view.php" > Просмотр списка причин
внесения</a><br>
<form action="cl_leg_add.php" meth-
od="post"> <table border=0>
<tr>
</tr>Наименование (кир)</td>
<td><input type="text" size=40 name="denom_cyr"></td>
<tr> Наименование (лат) </tr>
<td><input type="text"
size=40 name="denom_eng"></td></tr>
<td>Код ЕГРПОУ</tr>
<td><input type="text" size=15
name="egrpou"></td></tr>
<td>Юр. Адрес</td>
<td><input type="text" size=30
name="addr"></td></tr>
<td>Страна</td>
<td><input type="text" size=15
name="country"></td></tr>
<td>Причина внесения<M>
<td><input type="text"
size=2name="reas"></td></tr> </table>
<input type="submit"
value="добавить"> </form> </font>

```

### А.4 Приклад файлу, що реалізує введення даних, які були попередньо сформовані за допомогою html-тега

```

<td width=67% border=0 bgcolor="lightgrey"
align=center>
<font size="+3 color="red">Реестр причин внесения в стоп-лист
</font>
<?
$dbname='icb';
$hostname='icb';
$username='icb';

```

```

$password='icb';
$usertable="reas_reg
";
mysql_connect($hostname,$uname,$password) or DIE("Cannot
Create Connection");
mysql_select_db($dbname) or DIE(mysql_error());

mysql_query("insert into $usertable values
('$reas_id','$reas_name',
'$reas_explain')");
$q1 = mysql_query("select * from
$usertable");
echo "<table border=1>
<tr><td>ID</td>
<td>Наименование</td>
<td>Примечание</td>
</tr>";
if($q1)
{
while(list($reas_id,$reas_name,$reas_explain)=mysql_fetc
h_row($q1))
{
echo
"<tr><td>$reas_id</t
d>
<td>$reas_name</td>
<td>$reas_explain</t
d>
</tr>";
}
}
else
{
echo "<p><b>Error:
".mysql_error()."</b></p>"; exit();
}
?>
</font> </td></tr>

```

### A.5 Результат виконання запиту у сформований масив даних

```

<?
$dbname='icb';
$hostname='icb';

$username='icb';
$password='icb';
$usertable="cl_leg";
$usertable
l="reasjreg";

```



```

mysql__connect($hostname,$uname,$password) or DIE("Cannot
Create Connection");
mysql_select_db($dbname) or
DIE(mysql_error()); $ql = mysql_query("
select * from $usertable");
echo "<table border=1>
<tr><td>ID</td>
        <td width= 15%>Наименование (рус)</td>
        <td width= 5%>Наименование (анг)</td>
        <td width=15%>Код ЕГРПОУ</td >
        <td width=20%>Юридический адрес</td>
        <td width=10%>Страна</td>
        <td width=25%>Причина внесения</td >
    </tr>;
if($ql)
    {
        while(list($id,$denom_cyr,$denom_eng,$egrpou,$addr,$country,
        $reas)=mysql_fetch_row($ql))
            {
                $q2 = mysql_query("select reas_name from $usertable1
                where
                reas='$reas'");
                if($q2)
                    {
                        while(list($reas_name)=mysql_fetch_row(
                        $q2))
                            {
                                echo "<tr><td>$id</td>
                                <td>$denom_cyr</td>
                                <td>$denom_eng</td>
                                <td>$egrpou</td>
                                <td>$addr</td>
                                <td>$country</td>
                                <td>$reas_name</td>
                                </tr>";
                            }
                        }
                    else
                        {
                            echo "<p><b>Error:
                            ".mysql_error()."</b></p>"; exit();
                        }
                }
    }
else
    {
        echo "<p><b>Error: ".mysql_error().f""</b></p>";
        exit();
    }
?>

```

## A.6 Результуючий код занесення інформації про клієнта до журналу

```

$anc = mysql_query("select * from
leg_dossier"); if($anc)
{
    while(list($dleg_id,$dleg_denom_cyr,$dleg_denom_eng,$dleg_eg
rpou,$dle g_addr,$dleg_country,$dleg_reas)=mysql_fetch_row($anc))
        { $lsl = mysql_query(" select * from cl_leg where
leg_denom_cyr like
('dleg_denom_cyr%') or leg_denom_eng like ('dleg_denom_eng%')");
        if($lsl)
        {
            if (mysql_num_rows($lsl)>0)
            {
                while (list($leg_id,
$leg_denom_cyr,$leg_denom_eng,
$leg_egpou,$leg_addr,$leg_country,$leg_reas)=mysql_fetch_row($
anc))
                {
                    mysql_query("update leg_dossier set
dleg_risk='high' where dleg_id='$id'");
                    mysql_query("insert into cljourn (cl_id,cl_name, cl_date, reas)
values(0,'$leg_denom_cyr','$date','$leg_reas')");
                }
            }
        }
    else
    {
        echo , ,<p><b>Error:
".mysql_error()."</b></p>"; exit();
    }
}
else
{
    echo "<p><b>Error: ".mysql_error()."</p>";
    exit();
}
}

```