

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра комп'ютерних наук

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему **«Інформаційна технологія моніторингу банківських послуг для фізичних осіб»**

Виконав: студент 2 курсу,
групи 2КН-19м
спеціальності 122 «Комп'ютерні науки»
Іржавський О.Ю

Керівник: к.т.н., ст. викл. Озеранський В.С.

Рецензент: д.т.н., проф. Романюк О.Н.

Вінниця
2020

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
д.т.н., проф.. Яровий А.А.

(підпис)

“ _____ ” _____ 2020 року

ЗАВДАННЯ

на магістерську кваліфікаційну роботу на здобуття кваліфікації магістра наук зі спеціальності: 122 – «Комп'ютерні науки»

08-22.МКР.019.19.000.ПЗ

Магістранта групи 2КН-19м Іржавського Олексія Юрійовича

Тема магістерської кваліфікаційної роботи: «Інформаційна технологія моніторингу банківських послуг для фізичних осіб»

Вхідні дані: 1) кількість банківських послуг – до 10; 2) кількість банків для аналізу - не менше 10; 3) обсяг валют - до 10000000грн., об'єктно-орієнтована мова програмування.

Короткий зміст частин магістерської кваліфікаційної роботи:

1. Графічна: Граф-схема алгоритму роботи програмного забезпечення моніторингу банківських послуг для фізичних осіб, діаграма класів програмного забезпечення, початкові вікна програмного забезпечення, робочі вікна програмного забезпечення керування.

2. Текстова (пояснювальна записка): Вступ, аналіз предметної області моніторингу банківських послуг для фізичних осіб, розробка інформаційної технології моніторингу банківських послуг для фізичних осіб, програмна реалізація моніторингу банківських послуг для фізичних осіб, економічна частина, висновки, перелік використаних джерел, додатки.

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз сучасного рівня розвитку інформаційних технологій моніторингу банківських послуг для фізичних осіб Постановка задач дослідження			Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Аналіз методів створення моніторингу банківських послуг для фізичних осіб та керування обліковими записами			Математичні моделі, розділ 2
3	Розробка алгоритму моніторингу банківських послуг для фізичних осіб			розділ 3
4	Підготовка економічної частини			розділ 4
5	Апробація та/або впровадження результатів дослідження			тези доповідей
6	Оформлення пояснювальної записки, графічного матеріалу та презентації			Пояснювальна записка, графічний матеріал, презентація

Консультанти з окремих розділів магістерської кваліфікаційної роботи

1. Науковий керівник _____ канд. техн. наук, ст. викл. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

“ _____ ” _____ 20__ р.

В.С.Озеранський

ініціали та прізвище

2. Економічна частина _____ канд. екон. наук, доцент кафедри ЕПВМ
(підпис) наук. ступінь, вчене звання (посада)

“ _____ ” _____ 20__ р.

Бальзан М.В.

ініціали та прізвище

Дата попереднього захисту роботи “ _____ ” _____ 20__ р.

Рецензент _____ доктор техн. наук, професор, зав. каф. ПЗ
(підпис) наук. ступінь, вчене звання (посада)

Романюк О.Н.

ініціали та прізвище

Завдання видав науковий керівник _____ канд. техн. наук, ст. викл. кафедри КН
(підпис) наук. ступінь, вчене звання (посада)

В.С.Озеранський

ініціали та прізвище

“ _____ ” _____ 20__ р.

Завдання отримав магістрант _____ О.Ю.Іржавський
(підпис) ініціали та прізвище

“ _____ ” _____ 20__ р.

АННОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена удосконаленню інформаційної технології моніторингу банківських послуг для фізичних осіб. Запропонована технологія реалізує функції пошуку послуг за критеріями опису, зв'язок з банком, ознайомлення з інформацією про послуги.

В роботі було проведено аналіз існуючого програмного забезпечення моніторингу банківських послуг для фізичних осіб, визначено їх переваги та недоліки.

Програмна реалізація була здійснена для браузерів з використанням мови програмування PHP, база даних розроблялася за допомогою утиліти MySQL.

Даний програмний продукт дозволить користувачу здійснювати пошук банківських послуг, які його цікавлять, без реклами та інших факторів, які можуть відволікати його, використовуючи модифікований алгоритм та зручний інтерфейс.

ABSTRACT

This master's qualification work is devoted to the improvement of information technology for monitoring banking services for individuals. The proposed technology implements the functions of searching for services according to the criteria of description, communication with the bank, acquaintance with information about services. The paper analyzes the existing software for monitoring banking services for individuals, identifies their advantages and disadvantages.

The software implementation was carried out for browsers using the PHP programming language, the database was developed using the MySQL utility.

This software product will allow the user to search for banking services that interest him, without advertising and other factors that may distract him, using a modified algorithm and user-friendly interface.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ ТЕХНОЛОГІЙ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ	11
1.1 Суть технічної проблеми моніторингу банківських послуг для фізичних осіб ...	11
1.2 Аналіз програм-аналогів моніторингу банківських послуг для фізичних осіб ...	12
1.3 Висновок про необхідність розробки нового програмного продукту	16
1.4 Формулювання вимог до нової програми.....	17
1.5 Висновок	17
2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ.....	19
2.1 Розробка структури інформаційної технології моніторингу банківських послуг для фізичних осіб.....	19
2.2 Побудова діаграми прецедентів.....	20
2.3 Побудова діаграми діяльності моніторингу банківських послуг	22
2.4 Побудова IDEF0 діаграм.....	24
2.5 Побудова діаграми потоку даних моніторингу банківських послуг	27
2.6 Побудова схеми бази даних	28
2.7 Побудова діаграми класів.....	29
2.8 Розробка алгоритму функціонування інформаційної технології моніторингу банківських послуг	30
2.9 Висновок	30
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ	32
3.1 Вибір мови та середовища програмування.....	32
3.2 Вибір технологій для роботи з базою даних системи моніторингу банківських послуг для фізичних осіб	35
3.3 Побудова діаграми компонентів.....	36

3.4	Опис використаних бібліотек та модулів системи моніторингу банківських послуг для фізичних осіб.....	38
3.5	Розробка та опис програмних модулів системи моніторингу банківських послуг для фізичних осіб.....	39
3.6	Розробка інтерфейсу користувача системи моніторингу банківських послуг для фізичних осіб	42
3.8	Опис альтернативних підходів, використаних для розробки системи моніторингу банківських послуг для фізичних осіб	45
3.9	Тестування системи моніторингу банківських послуг для фізичних осіб та аналіз результатів її роботи	47
3.10	Висновок	49
4	ЕКОНОМІЧНА ЧАСТИНА.....	51
4.1	Оцінювання комерційного потенціалу розробки.....	51
4.2	Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи.....	54
4.3	Прогнозування комерційних ефектів від реалізації результатів розробки.....	58
4.4	Розрахунок ефективності вкладених інвестицій та період їх окупності.....	59
4.5	Висновок.....	62
	ВИСНОВКИ.....	63
	ЛІТЕРАТУРА.....	64
	Додаток А. Інструкція користувача.....	66
	Додаток Б. Лістинг програми.....	68
	Додаток В. Графічна частина	90

ВСТУП

Актуальність теми дослідження. На сьогоднішній день відбувається швидкий розвиток науки і техніки у всій областях людської діяльності. При цьому особлива увага приділяється банківській галузі. Автоматизація та комп'ютеризація процесів банківських послуг дозволяє забезпечити значну достовірність підбору послуг та спростити роботу працівникам.

Ще п'ять років тому, щоб підібрати собі якийсь банківський продукт, будь то кредитна карта або депозит, споживачеві необхідно було відвідати відділення банку – отримати інформацію з різного роду рекламних та інформаційних матеріалів, поспілкуватися з менеджером. Зараз цю проблему вирішують сайти самих же банків або тематичні сайти, де зібрана і зручно згрупована інформація яка цікавить вас банківським продуктам.

На сайті банку необхідно залишити персональну інформацію про себе та потреби – написати про те, як вас звать, в якому місті ви проживаєте, яка сума кредиту вам потрібна тощо.

Але час іде, і навіть неповороткі банки воліють міняти надання своїх же послуг. За останній рік більшість роздрібних великих банків підготували гарну інфраструктуру по віддаленій роботі з потенційними та існуючими клієнтами. Таким чином, клієнт можете вибрати і замовити банківський продукт, не виходячи зі свого будинку або офісу – достатньо зайти на сайт банку, в якому потрібно щось замовити, знайти цікавий продукт, наприклад кредитну картку, і замовити її онлайн.

Добре якщо клієнт знає, який банк йому необхідний, яка пропозиція найвигідніша. Правда це буває досить рідко і споживачеві доводиться «бігати» по сайтах різних банків, порівнювати їх послуги, знаходити ці найзаповітніші сторіночки, де можна замовити онлайн банківську послугу. Дану проблему повина вирішити наша система.

Системи онлайн підбору на цей момент також не мають достатньої ефективності, а саме – не забезпечують потрібної швидкості та високої достовірності підбору.

Отже, задача банківського моніторингу носить важливий прикладний характер, і для її розв'язання відсутні достатньо ефективні класичні методи, що спричинює

можливість застосування засобів штучного інтелекту для досягнення кращих показників достовірності та швидкодії моніторингу послуг.

Тема магістерської кваліфікаційної роботи є актуальною, оскільки завжди існує потреба у точному підборі послуг, а використання при цьому значень показників кредитування дозволить визначити відхилення від нормальних показників.

Використання методів штучного інтелекту для розв'язання поставленої задачі є актуальним через те, що надає великі можливості для модифікацій і налаштування різних методів під цю задачу, а отже дозволяє таким чином досягати кращого, порівняно з традиційними існуючими методами, результату.

Таким чином, застосування для розв'язання задачі сучасних методів штучного інтелекту забезпечує подальше впровадження інформаційних технологій у банківську галузь, а також є актуальною темою дослідження нових методів використання штучного інтелекту.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження магістерської кваліфікаційної роботи є розширення функціональних можливостей програмного забезпечення для моніторингу банківських послуг для фізичних осіб.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати існуючі техноогії моніторингу банкіських послуг для фізичних осіб;
- удосконалити інформаційну технологію моніторингу банкіських послуг для фізичних осіб;
- розробити алгоритм моніторингу банкіських послуг для фізичних осіб;
- за розробленим алгоритмом здійснити програмну реалізацію інформаційної технології моніторингу банківських послуг для фізичних осіб;
- здійснити тестування розробленого програмного забезпечення та аналіз результатів його роботи.

Об'єкт дослідження – це процес моніторингу банківських послуг з використанням інформаційних технологій.

Предмет дослідження – це програмні засоби моніторингу банківських послуг.

Методи дослідження. У роботі використано такі методи наукових досліджень: метод системного аналізу для аналізу структури інформаційної системи; теорія нейронних мереж для реалізації інформаційної технології до банківського моніторингу; методи математичної статистики; методи об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в такому:

– удосконалено інформаційну технологію моніторингу банківських послуг для фізичних осіб, яка відрізняється від існуючих аналізом парсингованих даних про банківські послуги, що дозволяє розширити функціональні можливості системи моніторингу банківських послуг для фізичних осіб;

Практичне значення одержаних результатів полягає у такому:

1. Розроблено алгоритм аналізу парсингованих даних для знаходження вигіднішої пропозиції для фізичної особи.
2. Здійснено програмну реалізацію інформаційної технології моніторингу банківських послуг для фізичних осіб.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Результати даної магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить аналіз наявних технологій моніторингу банківських послуг для фізичних осіб [1].

Апробація результатів роботи. Результати досліджень було апробовано на 12-й міжнародній науково-практичній конференції «Інтернет-Освіта-Наука-2020»

Публікації. За основними результатами досліджень опубліковано одну публікацію [1].

1 АНАЛІЗ ТЕХНОЛОГІЙ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ

1.1 Суть технічної проблеми моніторингу банківських послуг для фізичних осіб

Ще п'ять років тому, щоб підібрати собі якийсь банківський продукт, будь то кредитна карта або депозит, споживачеві необхідно було відвідати відділення банку – отримати інформацію з різного роду рекламних та інформаційних матеріалів, поспілкуватися з менеджером. Зараз цю проблему вирішують сайти самих же банків або тематичні сайти, де зібрана і зручно згрупована інформація яка цікавить вас банківським продуктам.

На сайті банку необхідно залишити персональну інформацію про себе і ваших потребах – написати про те, як вас звать, в якому місті ви проживаєте, яка сума кредиту вам потрібна тощо.

Але час іде, і навіть неповороткі банки воліють міняти надання своїх же послуг. За останній рік більшість роздрібних великих банків підготували гарну інфраструктуру по віддаленій роботі з потенційними та існуючими клієнтами. Таким чином, ви можете вибрати і замовити банківський продукт, не виходячи зі свого будинку або офісу – достатньо зайти на сайт банку, в якому ви хочете щось замовити, знайти цікавий продукт, наприклад кредитну картку, і замовити її онлайн.

Добре якщо можна знати наперед, який банк буде необхідний, яка пропозиція найвигідніша і т.д. Правда, це буває досить рідко і споживачеві доводиться «бігати» по сайтах різних банків, порівнювати їх послуги, знаходити ці найзаповітніші сторіночки, де можна замовити онлайн банківську послугу. Дану проблему повина вирішити нова розробка.

1.2 Аналіз програм-аналогів моніторингу банківських послуг для фізичних осіб

На сьогоднішній день існує багато засобів для моніторингу інформації. Одним з таких інструментів є API. Прикладний програмний інтерфейс (англ. Application Programming Interface, API) — набір визначень взаємодії різнотипного програмного забезпечення. API — це зазвичай (але не обов'язково) метод абстракції між низькорівневим та високорівневим програмним забезпеченням.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA). Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними [3].

XML – це спроба вирішити перераховані проблеми шляхом створення простої мови розмітки, що описує довільні структуровані дані. Точніше кажучи, це метамова, на якій пишуться спеціалізовані мови, що описують дані певної структури.

Такі мови називаються XML-словниками. На відміну від HTML, XML не містить ніяких вказівок на те, як описані в XML-документі дані повинні відображатися. Спосіб відображення даних для різних пристроїв задається мовою опису стилів XSL, який грає для XML приблизно ту ж роль, що CSS дл HTML [4].

Інша принципова її відмінність від HTML полягає в тому, що XML може містити будь-які теги, які вважатимуть за потрібне використовувати творці XML-словника.

Наведемо список лише декількох спеціалізованих мов на базі XML, які сьогодні знаходяться в різних стадіях розробки робочими групами W3C: MathML – мова математичних формул; SMIL – мова інтеграції і синхронізації мультимедійних засобів; SVG – мова двовимірної векторної графіки; RDF – мова метаописів ресурсів; XHTML – переформулювання HTML в термінах XML.

Процес обробки XML-документа полягає в наступному. Його текст аналізується спеціальною програмою, яка називається XML-процесором. XML-процесор нічого не знає про семантику даних у документі, він тільки виробляє синтаксичний розбір (parsing) тексту документа і перевіряє його правильність з точки зору правил XML. Якщо документ правильно оформлений (well-formed), то результати розбору тексту передаються XML-процесором прикладній програмі, яка виконує їх змістовну обробку, якщо ж документ оформлений невірно, тобто містить синтаксичні помилки, то XML-процесор повинен повідомити про них користувачеві. HTML не висловлює змісту документів [2].

JSON — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу (завдяки процесу, що називають серіалізацією).

JSON знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML (використовується в AJAX) під час асинхронної передачі структурованої інформації між клієнтом та сервером.

На даний момент, в мережі інтернету існує декілька аналогів до мого програмного продукту.

Взагалі користувач може без допомоги нашого програмного забезпечення самостійно знаходити офіційний веб-сайт банків для замовлення банківської послуги,

але так як банків є досить багато, це буде не продуктивно перевіряти кожний банк на наявність потрібної нам послуги.

Зручний сервіс – «Фінансовий супермаркет», який дозволяє вам в одному місці підібрати, порівняти і головне – замовити банківські послуги різних банків.

Інформація, представлена в даному сервісі, є актуальною і, головне, достовірною – без жодних банківських маркетингових вивертів. Ви можете не хвилюватися, що замовляєте «кота в мішку». Тобто, кредитну картку під 0,001% річних, а насправді під 48% річних, адже сервіс вже зробив всю роботу за вас – порахував всі витрати на отримання тієї чи іншої кредитної картки, і звів все в єдиний показник – загальний дохід по депозиту або загальну переплату по кредиту (рисунок 1.1).

Перевагою даного сервісу є надання можливості порівняння де-яких банківських послуг (отримання кредиту, можливість вкладу грошей і т.д.). Недоліком сервісу є неможливість вибрати країну в якій користувач хоче замовити банківську послугу та користувацький інтерфейс лише на російській мові.

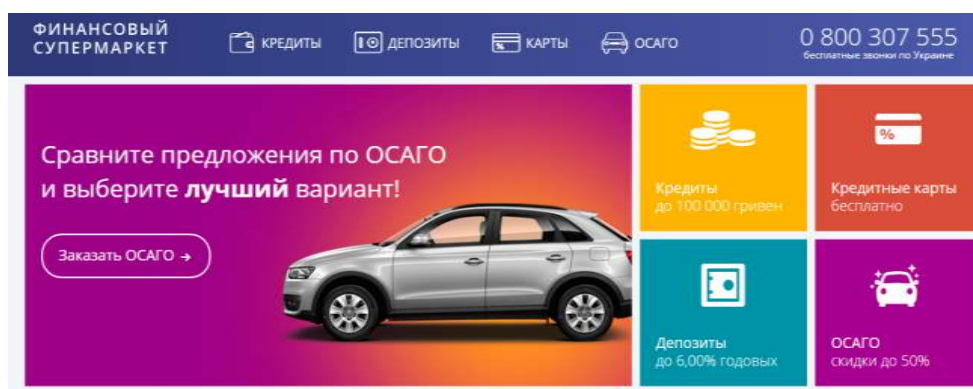


Рисунок 1.1 – Функціонал сайту «Фінансовий супермаркет»

Ще один сервіс для здійснення банківських послуг онлайн є сервіс «dvbank», який містить широкий вибір послуг, зручний інтерфейс (рисунок 1.2).

Недоліком є обмеження лише в російській мові та неможливість вибору за критерієм «тип послуги».

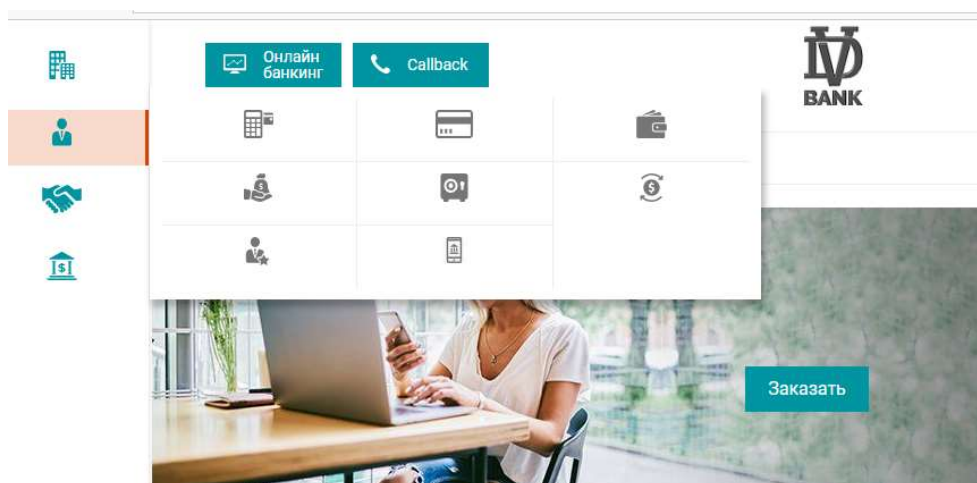


Рисунок 1.2 – Функціонал сайту «dvbank»

Порівняльний аналіз програм-аналогів з нашим програмний забезпеченням наведено в таблиці 1.1.

Таблиця 1.1 - Порівняльний аналіз програм-аналогів з новим програмний забезпеченням

	Наше програмне забезпечення	Фінансовий супермаркет	dvbank
Кількість країн	5	1	1
Тип послуги	є	є	немає
Інтерфейс на українській мові	є	немає	немає
Можливість порівняння банківської послуги	немає	є	немає

Отже ми можемо бачити що нове програмне забезпечення має певні переваги над програмами-аналогами.

1.3 Висновок про необхідність розробки нового програмного продукту

Для того щоб збільшити ефективність діяльності, та зменшити на неї витрати часу, нам потрібно при пошуку банківських послуг додати ще такі критерії, з допомогою яких ми зможемо отримати саме ту інформацію, яка і цікавить користувача.

Одним з таких критеріїв є тип послуги. Це може бути кредитування, інвестиції, страхування та інші.

Іншим, не менш важливим критерієм є країна.

Після вибору критеріїв система повинна відфільтрувати всі банки та надати можливість замовлення послуги перейшовши на сайт банку. На сайті банку необхідно буде залишити персональну інформацію про себе і ваших потребах – написати про те, як вас звуть, в якому місті ви проживаєте, яка сума кредиту вам потрібна тощо. Візит в банк вже не займе багато часу, і швидше за все, буде необхідний тільки один раз, власне для підписання договору на надання банківської послуги.

Головна задача - це виділення зі всіх пропозицій саме ту частину, яка задовольняє всім нашим критеріям. Механізми моніторингу дозволяють нам це зробити з допомогою критеріїв які були описані вище.

Якщо користувачеві буде дійсно цікава вибрана послуга – нам вдасться зменшити пошукові витрати його часу на 80%. Для фізичних осіб які користуються банківськими послугами – це дозволить збільшити ефективність своєї діяльності в декілька разів.

Таким чином, створивши систему яка дозволить парсити банківські послуги за критеріями та з допомогою інструментів, що були описані в попередньому розділі, ми можемо бути впевнені, що вони будуть показуватися аудиторії яка дійсно зацікавлена в цих послугах, що пропонує наша система. Дана система буде цікава лише фізичним особам, так як лише вони можуть здійснювати банківські послуги онлайн.

В подальшому є перспектива його реалізації на ринку інтернет маркетингу з метою отримання прибутку.

Переваги даної системи очевидні:

- Економія вашого часу;
- Великий асортимент банківських продуктів

1.4 Формулювання вимог до нової програми

Сформулювавши способи вирішення проблеми, ми розуміємо потрібний нам функціонал та можемо скласти список вимог до нашого програмного засобу.

Вимоги до функціональних характеристик:

- Парсинг банківських послуг за критеріями:
 - Тип послуги
 - Країна
- Пошук пропозицій за ключовими словами;
- Стійкість до помилок у вхідних даних.

Вимоги до програмного забезпечення:

- Програмний засіб повинен використовувати API - це інтерфейс, який дозволяє отримувати інформацію з бази даних за допомогою http-запитів до спеціального серверу. Синтаксис запитів і тип повертаються ними даних строго визначені на стороні самого сервісу
 - Робота за допомогою веб-інтерфейсу;
 - Видача результатів в форматі XML.

1.5 Висновок

В даному розділі було проаналізовано суть технічної проблеми моніторингу банківських послуг для фізичних осіб. Виконано порівняльний аналіз програм-аналогів моніторингу банківських послуг для фізичних осіб, розглянуто їх основні переваги та недоліки. Обґрунтовано необхідність розробки нового програмного продукту для моніторингу банківських послуг для фізичних осіб. Сформульовано вимоги до нової програми моніторингу банківських послуг для фізичних осіб.

2 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ

2.1 Розробка структури інформаційної технології моніторингу банківських послуг для фізичних осіб

На вхід подаються дані для підбору потрібних послуг. Після цього перевіряється введені данні клієнта на відповідність. Далі відбувається процес підбору банківських послуг за вказаними даними. Після цього вирахування найкращих варіантів та виведення результату Структура інформаційної технології моніторингу банківських послуг для фізичних осіб зображена на рисунку 2.1

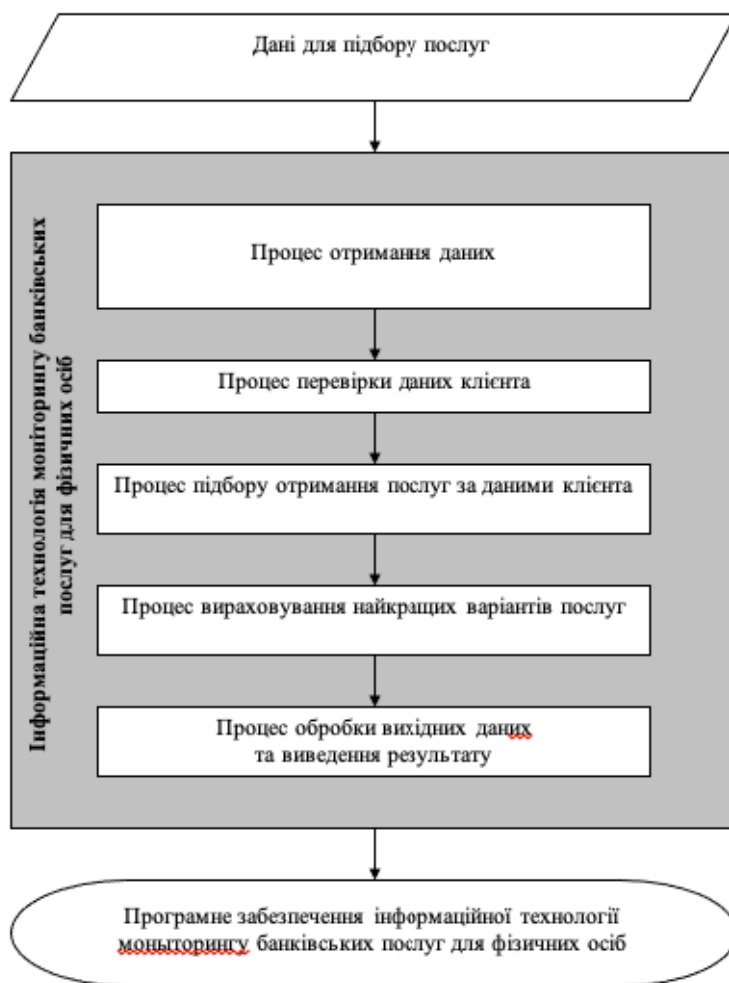


Рисунок 2.1 – Структура інформаційної технології моніторингу банківських послуг для фізичних осіб

2.2 Побудова діаграми прецедентів

Діаграма прецедентів в UML - це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також перекладається як діаграма варіантів використання.

Прецедентом (use-case) називається опис множини послідовностей дій (включаючи варіанти), що виконуються системою для того, щоб актор міг отримати певний результат. Графічно прецедент зображується у вигляді еліпса.

Такі діаграми описують функціональність, яка буде надаватись користувачам системи, котра проектується. Представляються шляхом використання прецедентів та акторів, а також відношень між ними. Набір усіх прецедентів діаграми фактично визначає функціональні вимоги, за допомогою яких може бути сформульоване технічне завдання.

Суб'єкт — це система, що розглядається, і до якої відносяться прецеденти. Користувачів та будь-які інші системи, що можуть взаємодіяти із суб'єктом, представлено як акторів. Актори завжди представляють сутності, що знаходяться за межами системи. Поведінка суб'єкта описується одним або більше прецедентами, що визначаються відповідно до потреб акторів.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

- асоціації (англ. association relationship)
- включення (англ. include relationship)
- розширення (англ. extend relationship)
- узагальнення (англ. generalization relationship)

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML чудово зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки.

Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

На рисунку 2.2 зображена діаграма прецедентів моніторингу банківських послуг.

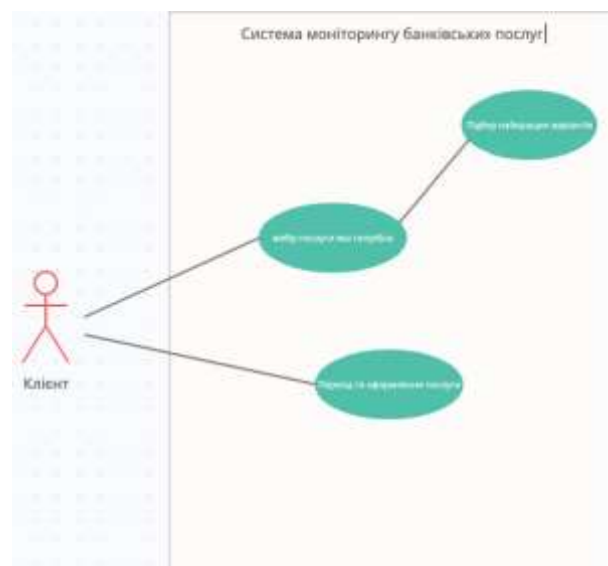


Рисунок 2.2 – Діаграма прецедентів моніторингу банківських послуг

Користувач може взаємодіяти з системою наступним чином:

- вибір послуги яка потрібна:
- перехід та оформлення потрібної послуги

2.3 Побудова діаграми діяльності моніторингу банківських послуг

Діаграма діяльності (англ. activity diagram) — UML-діаграма, яка є візуальним представленням графу діяльностей, який є одним з різновидів графу станів скінченного автомату, вершинами якого є певні дії або діяльності, а переходи відбуваються тільки по завершенню дій. Діаграми діяльності використовуються при моделюванні бізнес-процесів, технологічних процесів та послідовних або паралельних обчислень.

При побудові діаграми діяльності використовуються обмежені типи фігур, які з'єднані стрілками:

- чорний кружок, який позначає старт (початковий стан) процесу;
- прямокутник із округленими кінцями, який позначає дію;
- ромб, який позначає рішення;
- горизонтальна риска, яка позначає початок чи кінець паралельних активностей;
- чорний кружок в колі, який позначає кінець (кінцевий стан) процесу.

На рисунку 2.3 зображена діаграма діяльності технології моніторингу банківських послуг.

В цій діаграмі описані дії, які виконує система, коли користувач робить запит на інформацію. Спочатку система отримує доступ до веб-даних після чого считує дані, якщо вони не збігаються з існуючими в базі, то вона вилучає отриманні дані. Далі відбувається перевірка коректності отримання даних. Якщо дані отримані не коректні, то відбувається повторення вилучення даних поки вся інформація не буде коректно вилучена.

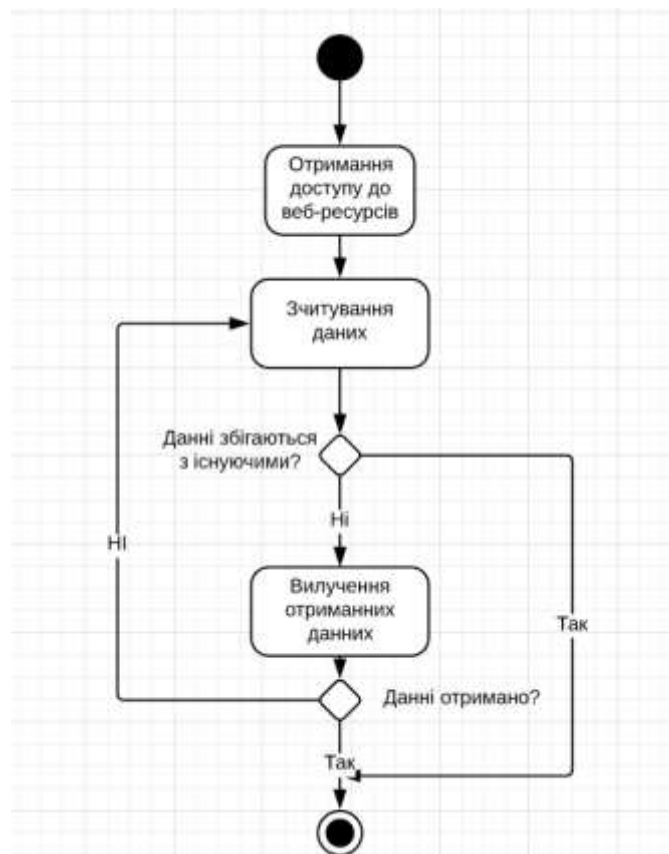


Рисунок 2.3 – Діаграма діяльності моніторингу банківських послуг

Аналіз парсингованих даних про банківські послуги для вибору найвигіднішого варіанту для користувача відбувається таким чином.

1. Після заходу клієнт вносить дані за якими потрібно підібрати послуги
2. Враховуючи дані від користувача та дані отримані від банків підбирається найкращі пропозиції в залежності від послуги за такими критеріями:

1. Кредит

- a. Процент який нараховується поверх тіла кредиту
- b. Можливість дострокового погашення
- c. Дострокове погашення з оплатою процентів тільки за час користування

2. Депозит

- a. Процент який нараховується за період
- b. Можливість зняття вкладу в певний період
- c. Нарахування проценту при достроковому знятті

3. Після успішного підбору клієнт отримує результат та може перейти на потрібний йому сервіс для оформлення

Даграма діяльності аналізу парсингованих даних для вибору вигіднішого варіанту представлена на рисунку 2.4.

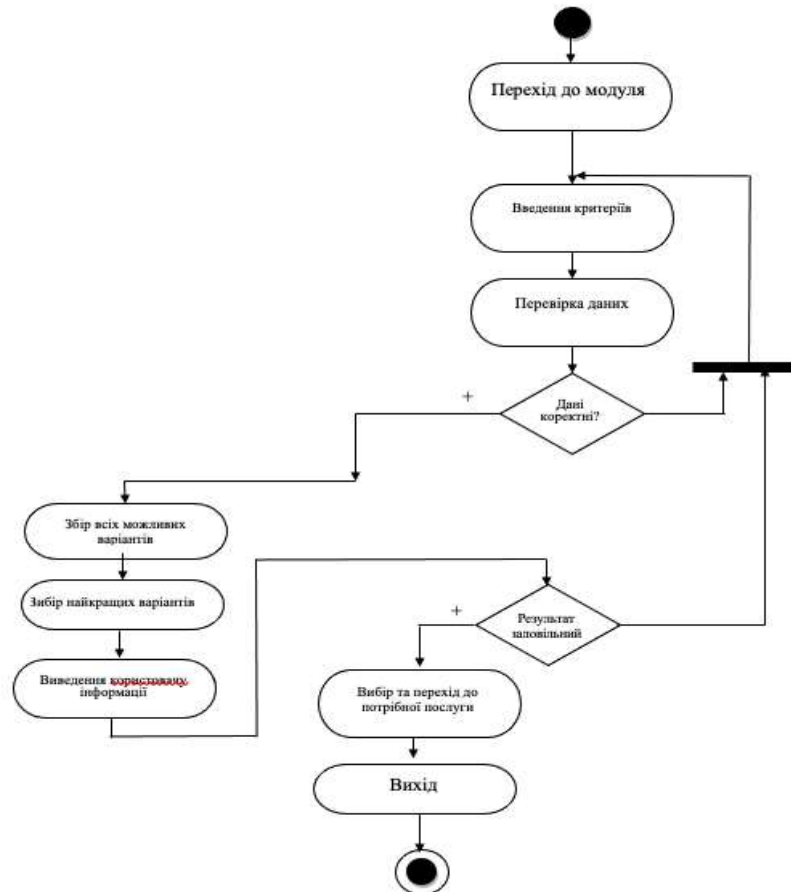


Рисунок 2.4 – Діаграма діяльності аналізу парсингованих даних для вибору вигіднішого варіанту

2.4 Побудова IDEF0 діаграм

IDEF0 — методологія функціонального моделювання (англ. Function modeling) і графічна нотація, призначена для формалізації і опису бізнес-процесів. Відмінною особливістю IDEF0 є її акцент на підпорядкованість об'єктів. В IDEF0 розглядаються не їх тимчасова послідовність, а логічні відносини між роботами.

Стандарт IDEF0 працює за правилом — найбільш важлива функція знаходиться у верхньому лівому кутку, крім того кожна з чотирьох сторін функціонального блоку має своє певне значення (роль):

- Верхня сторона має значення "Управління" (Control);
- Ліва сторона має значення "Вхід" (Input);
- Права сторона має значення "Вихід" (Output);
- Нижня сторона має значення "Механізм" (Mechanism).

Опис виглядає як «чорний ящик» з входами, виходами, управлінням і механізмом, який поступово деталізується до необхідного рівня. Також існують словники опису робіт і стрілок, які створені для того, щоб вірно їх інтерпретувати. У цих словниках можна дати опис змісту, який вкладений в певну роботу або стрілку.

Методологію IDEF0 можна вважати наступним етапом розвитку мови описання функціональних систем SADT (Structured Analysis and Design Technique). IDEF0, як стандарт був розроблений в 1981 році в рамках широкої програми автоматизації промислових підприємств, яка носила позначення ICAM (Integrated Computer Aided Manufacturing) і була запропонована департаментом військово-повітряних сил США. Сімейство стандартів IDEF взяло своє позначення від назв цієї програми (IDEF=ICAM DEFinition). В процесі практичної реалізації, учасники програми ICAM зіткнулись з необхідністю розробки нових методів аналізу процесів взаємовідносин в промислових системах. При цьому крім покращеного набору функцій для опису бізнес-процесів, однією з вимог до нового стандарту була наявність ефективної методології взаємодій в рамках «аналітик-спеціаліст». Іншими словами, новий метод повинен був забезпечити групову роботу над створенням моделі, з безпосередньою участю всіх аналітиків і спеціалістів, які беруть участь в рамках проекту.

Контекстна діаграма, яка зображена на рисунку 2.5 показує призначення системи і її взаємодію з зовнішнім середовищем.

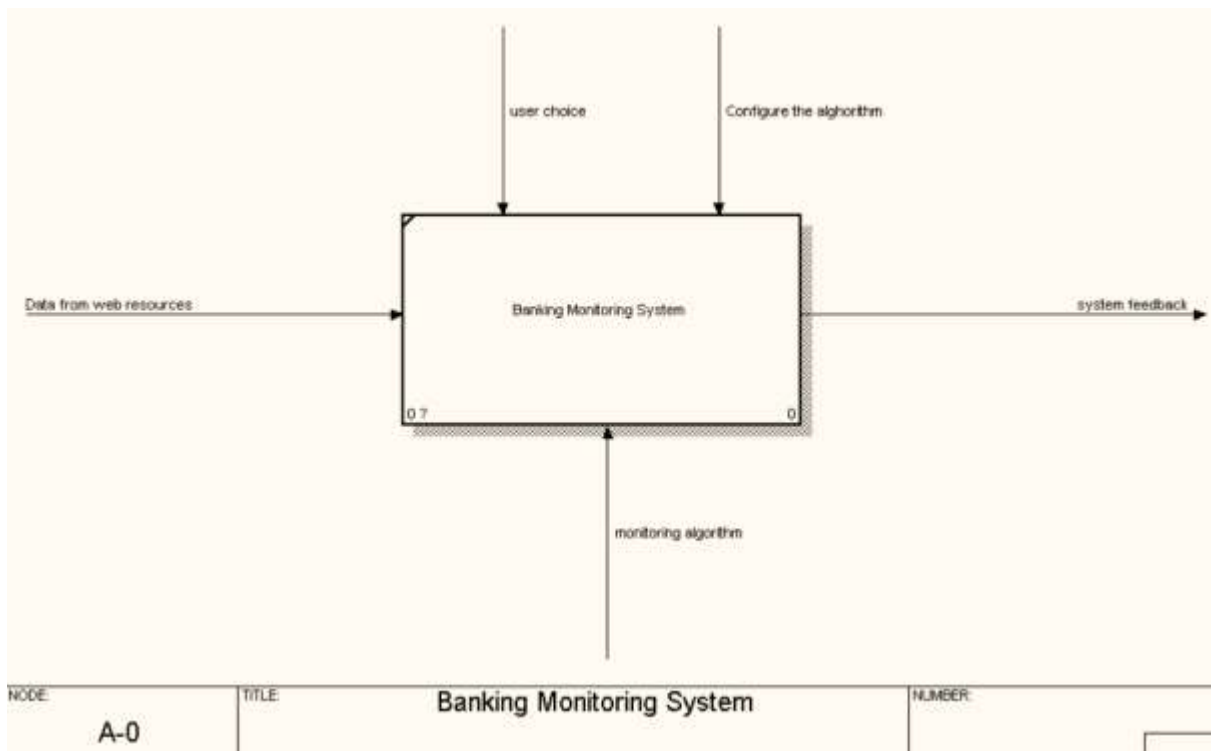


Рисунок 2.5 – Контекстна діаграма моніторингу банківських послуг

Діаграма декомпозиції дозволяє розбити складний процес на складові його функції, тобто представити модель системи поступово і структуровано для спрощення розуміння.

У процесі декомпозиції, функціональний блок, який в контекстній діаграмі відображає систему як єдине ціле, піддається деталізації. Отримана діаграма першого рівня декомпозиції містить функціональні блоки, що відображають головні підфункції функціонального блоку контекстної діаграми. Кожен функціональний блок в діаграмі першого рівня декомпозиції називається дочірнім блоком, а функціональний блок контекстної діаграми – батьківським. Також кожен функціональний блок в дочірній діаграмі може бути деталізований шляхом створення діаграми декомпозиції другого рівня для цього блоку.

На рисунку 2.6 зображена діаграма декомпозиції першого рівня моніторингу банківських послуг.

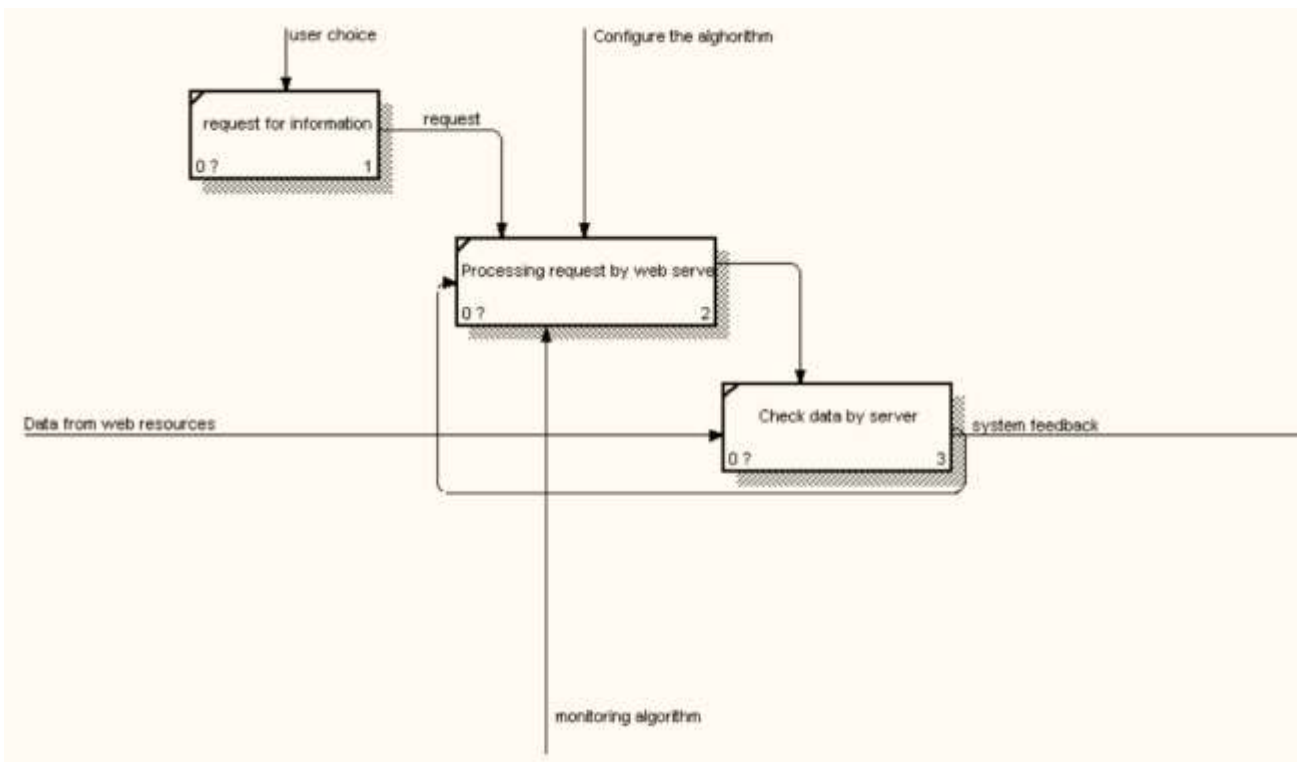


Рисунок 2.6 – Діаграма декомпозиції першого рівня моніторингу банківських послуг

На діаграмі декомпозиції першого рівня зображені функціональні блоки, які відображають під функції роботи системи контролю споживання електроенергії. Кожен блок виконує певну функцію.

2.5 Побудова діаграми потоку даних моніторингу банківських послуг

Діаграми потоків даних (англ. Data Flow Diagramming) є основним засобом моделювання функціональних вимог до проектованої системи. Вимоги представляються у вигляді ієрархії процесів, зв'язаних потоками даних. Діаграми потоків даних показують, як кожен процес перетворює свої вхідні дані у вихідні, і виявляють відношення між цими процесами. DFD-діаграми успішно використовуються як доповнення до моделі IDEF0 для опису документообігу й обробки інформації. Подібно IDEF0, DFD представляє систему, що моделюється, як мережу зв'язаних робіт. Основні компоненти DFD (як було сказано вище) – процеси або роботи, зовнішні сутності, потоки даних, сховища даних.

На рисунку 2.7 зображена діаграма потоку даних моніторингу банківських послуг.

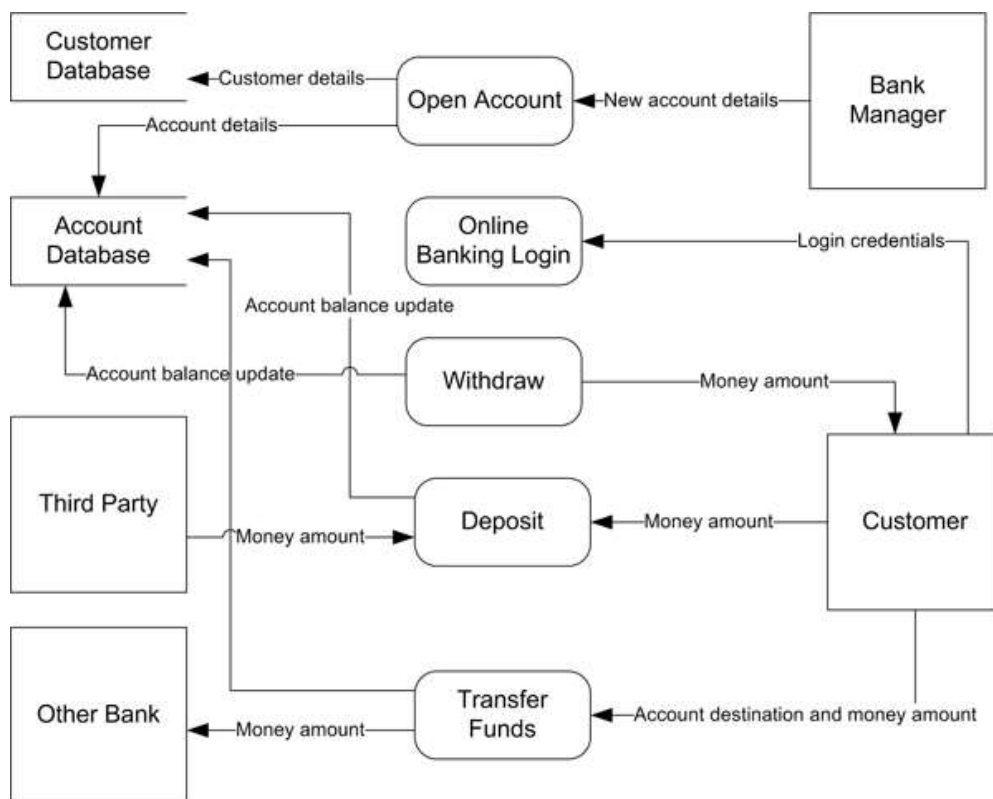


Рисунок 2.7 – Діаграма потоку даних моніторингу банківських послуг

На діаграмі потоку даних зображені зовнішні сутності такі як користувач та процеси обробки даних та сховища даних, в яких зберігається інформація про список та стан.

2.6 Побудова схеми бази даних

База даних системи складається з двох таблиць – списку банків та записів про стан послуг. Записи створюються по мірі отримання даних від сервера. На їх основі формується висновок про поточний стан кожного побутового приладу та усієї системи вцілому. Поняття схеми бази даних відіграє ту ж роль, що і поняття теорії у численні предикатів. Модель цієї «теорії» точно відповідає базі даних, яку можна побачити в будь-який момент часу як математичний об'єкт. Таким чином, схема може містити формули, що представляють обмеження цілісності спеціально для

застосунків і обмеження спеціально для типу бази даних, які виражені однією мовою баз даних. В реляційній базі даних, схема визначає таблиці, поля, відношення, індекси, пакети, процедури, функції, черги, тригери, типи даних, послідовності, матеріалізовані уявлення, синоніми, посилання баз даних, каталоги, Java, XML-схеми та інші елементи. Схема, як правило, зберігається в словнику даних.

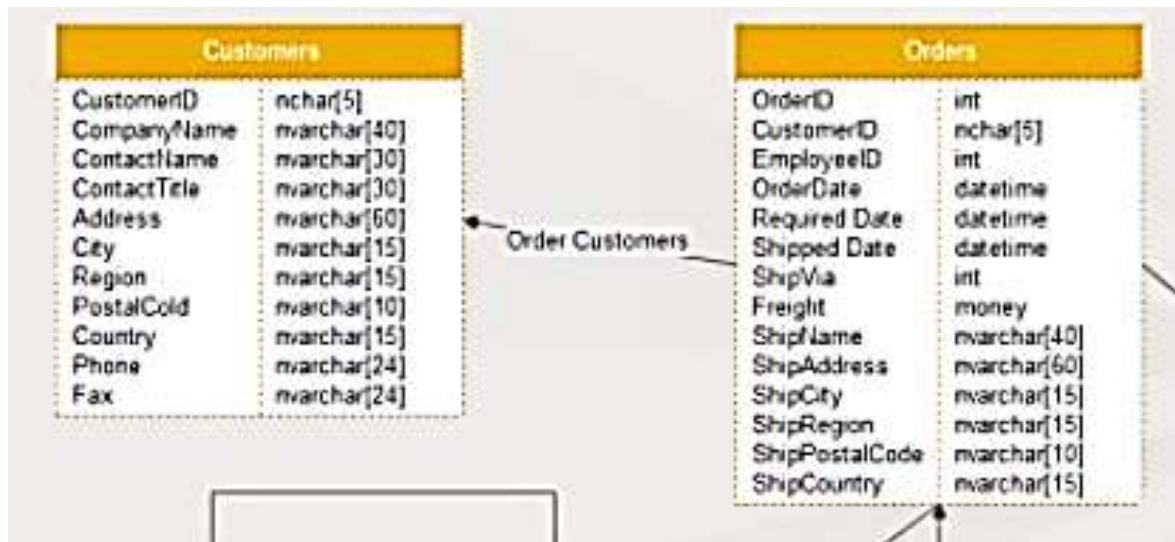


Рисунок 2.8 – ER-діаграма бази даних системи моніторингу банківських послуг

Хоча схема визначена в тексті мовою бази даних, цей термін часто використовується для графічного позначення структури бази даних. Іншими словами, схема — це структура бази даних яка визначає об'єкти в базі даних. Entity Relations діаграму бази даних наведено на рисунку 2.8.

2.7 Побудова діаграми класів

Діаграма класів — статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.[6] Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

На рисунку 2.9 наведена діаграма класів системи моніторингу банківських послуг для фізичних осіб.

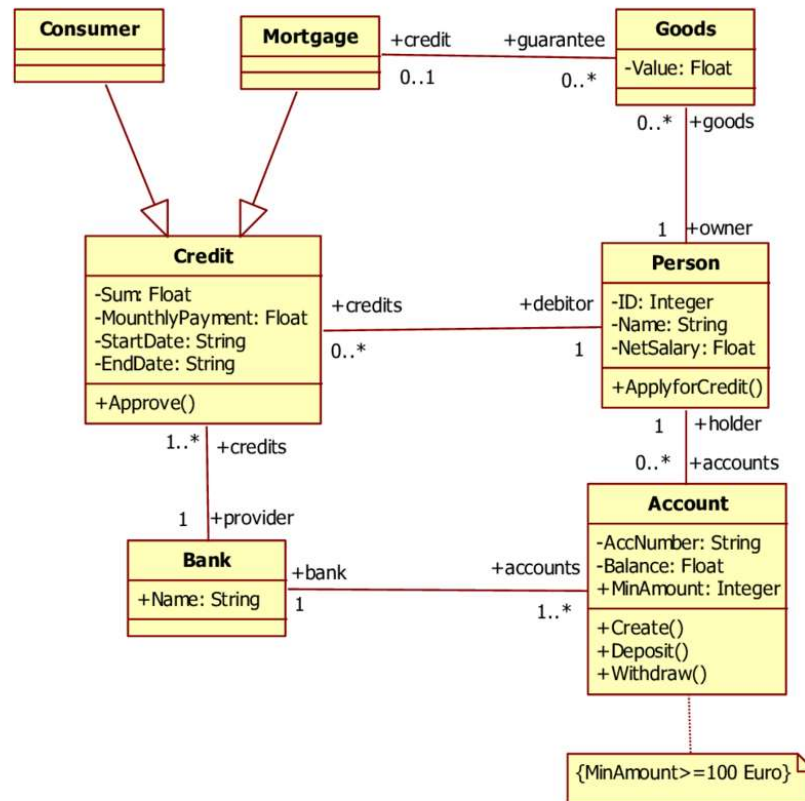


Рисунок 2.9 – Діаграма класів системи моніторингу банківських послуг для фізичних осіб

2.8 Розробка алгоритму функціонування інформаційної технології моніторингу банківських послуг

В функціонуванні інформаційної технології можна виокремити 3 частини.

В першу чергу, відбувається вихід в інтернет та з'єднання з кодом веб-ресурсу на якому буде відбуватися парсинг банківських послуг. Для цього потрібно, щоб наша база даних була з'єднана з відповідним сервером.

Наша система автоматично виконує моніторинг банківських послуг кожний день, оновлює дані в базі даних, видаляючи стару неактуальну інформацію використовуючи API. Після цього відбувається парсинг даних і занесення їх в базу даних (рисунок 2.10).

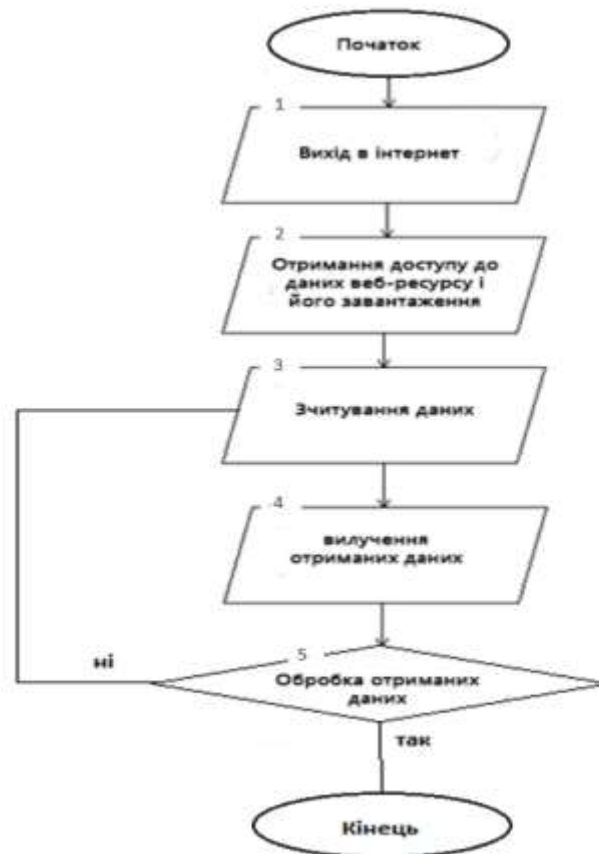


Рисунок 2.10 – Алгоритм функціонування технології моніторингу банківських послуг

Після введення вхідних даних в пошуковому фільтрі та їх підтвердження, розпочинається зчитування відфільтрованої інформації з бази даних за критеріями.

2.9 Висновок

У даному розділі було розроблено структуру інформаційної технології моніторингу банківських послуг для фізичних осіб. Побудовано діаграми прецедентів, діаграми діяльності інформаційної технології моніторингу банківських послуг для фізичних осіб. Побудовано діаграми IDEF0, IDEF1 та DFD, схема бази даних, діаграми класів. Розроблено алгоритм функціонування інформаційної технології моніторингу банківських послуг для фізичних осіб.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ МОНІТОРИНГУ БАНКІВСЬКИХ ПОСЛУГ ДЛЯ ФІЗИЧНИХ ОСІБ

3.1 Вибір мови та середовища програмування

PHP (Hypertext Preprocessor) – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP підтримується переважною більшістю хостинг-провайдерів.

Головним чинником мови PHP є практичність. PHP повинен надати програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційність;
- простота;
- ефективність;
- безпека;
- гнучкість.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. PHP – мова, що поєднує достоїнства Perl, C і спеціально націлена на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. Вона може бути вбудована безпосередньо в html-код сторінок, які, у свою чергу будуть коректно оброблятися PHP-інтерпретатором. Велика різноманітність функцій PHP позбавляє від написання багаторядкових, призначених для користувача функцій на C або Pascal.

Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслюючим інтерпретатором. Такий «движок» PHP дозволяє обробляти сценарії з достатньо високою швидкістю.

PHP надає в розпорядження розробників та адміністраторів гнучкі та ефективні засоби безпеки, які умовно поділяються на дві категорії: засоби системного рівня та засоби рівня програми.

У PHP реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні PHP це забезпечує максимальну

свободу дій і безпеку. PHP може працювати в так званому безпечному режимі (safe mode), який обмежує можливості застосування PHP користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання та використання пам'яті (неконтрольована витрата пам'яті негативно впливає на швидкодію сервера).

У стандартний набір функцій PHP входить ряд надійних механізмів шифрування. PHP також сумісний з багатьма додатками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції (e-commerce). Інша перевага полягає в тому, що вихідний текст сценаріїв PHP не можна переглянути у браузері, оскільки сценарій компілюється до його відправлення за запитом користувача.

Оскільки PHP є вбудовуванню (embedded) мовою, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати у поєднанні з HTML, він з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані програми PHP легко розширюються в міру необхідності (втім, це відноситься до всіх основних мов програмування). Поріг входження цієї мови є досить низьким, що дозволяє великій кількості програмістів використовувати її, затративши дуже мало часу на процес навчання.

Отже, проаналізувавши web-технології програмування було вирішено використовувати мову програмування PHP для написання сценаріїв.

Хоча PHP є досить молодою мовою, вона знайшла таку популярність серед web-програмістів, і на даний момент є мало чи не найпопулярнішою мовою для створення web-додатків (скриптів).

Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal, що помітно знижує початкові зусилля при вивченні PHP.

Головні особливості PHP – практичність, легкість у використанні, ефективність, продуктивність та гнучкість.

Прийняття стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціненну послугу користувачам. До того, співтовариство

користувачів РНР є свого роду «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді на найскладніші питання.

Node.js — програмна платформа з відкритим кодом, яка використовує рушій V8, для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript. Node.js дає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API (написаний на C ++), підключати інші зовнішні бібліотеки, написані на різних мовах програмування, забезпечуючи виклики до них з JavaScript-коду. Якщо раніше Javascript застосовувався для обробки даних в браузері на стороні користувача, то node.js надав можливість виконувати JS-скрипти на сервері та відправляти користувачеві результат, виконуючи роль веб-сервера. Також на Node.js є можливість розробляти десктопні віконні додатки (за допомогою NW.js, AppJS або Electron для Linux, Windows і macOS) і навіть програмувати мікроконтролери. В основі Node.js лежить подійно-орієнтоване і асинхронне програмування. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Node.js має наступні властивості:

- асинхронна однопотокова модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;
- рушій JavaScript Google V8;

Git – це розподілена система для контролю версій. Це і є головною різницею між svn та git. Іншими словами можна сказати, що кожен користувач може створити на своєму домашньому комп'ютері повноцінний, окремий репозиторій.

Використовуючи цей репозиторій можна роботи все теж, що і у svn – створювати гілки, переглядати зміни, виконувати комміти. Для того, щоб можна було працювати з віддаленими репозиторіями і обмінюватися змінами з іншими користувачами, в git є дві команди, які не мають аналогів – git push і git pull.

Git push – влиття локальних змін у віддалений репозиторій. Git pull – влиття змін з віддаленого репозиторія в локальний. Обмін даними переважно відбувається з використанням протоколу SSH.\

Git підтримується декількома великими репозиторіями – GitHub, SourceForge, BitBucket та Google Code. Зручно використовувати один з вище описаних в якості основного сховища для корпоративних проектів.

У даному розділі було розглянуто проблеми, що можуть виникнути на шляху розробки чат-боту. Побудовано дерево цілей, розглянуто різні засоби розробки. Також побудовано і описано концептуальну модель.

3.2 Вибір технологій для роботи з базою даних системи моніторингу банківських послуг для фізичних осіб

База даних MySQL - реляційна база даних. У реляційній базі даних дані зберігаються не всі скопом, а в окремих таблицях, завдяки чому досягається вигреш у швидкості та гнучкості. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частина системи MySQL можна охарактеризувати як мова структурованих запитів плюс найбільш поширений стандартний мову, що використовується для доступу до баз даних .

СУБД MySQL є системою клієнт-сервер, яка містить багато-поточний SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також кілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API).

Сервер MySQL постійно працює на комп'ютері. Клієнтські програми (наприклад, скрипти PHP) посилають серверу MySQL SQL-запити через механізм сокетів (тобто за допомогою мережевих засобів), сервер їх обробляє і запам'ятовує результат. Тобто скрипт (клієнт) вказує, яку інформацію він хоче отримати від сервера баз даних. Потім сервер баз даних посилає відповідь (результат) клієнтові (скрипту).

Структура MySQL трирівнева:

- а) бази даних;
- б) таблиці;

в) записи.

Логічно - таблиця являє собою сукупність записів. А записи - це сукупність полів різного типу. Ім'я бази даних MySQL унікально в межах системи, а таблиці - в межах бази даних, поля – в межах таблиці. Один сервер MySQL може підтримувати відразу декілька баз даних, доступ до яких може розмежовуватися логіном і паролем. Знаючи ці логін і пароль, можна працювати з конкретною базою даних. Наприклад, можна створити або видалити в ній таблицю, додати записи і т. д. Зазвичай ім'я-ідентифікатор та пароль призначаються хостинг провайдерами, які і забезпечують підтримку MySQL для своїх користувачів. Програмне забезпечення MySQL є додатком з відкритим кодом. Кожен користувач може вивчити вихідний код і змінити його у відповідності зі своїми потребами.

Основні переваги пакету MySQL:

- багатопоточність. Підтримка декількох одночасних запитів;
- оптимізація зв'язків з приєднанням багатьох даних за один прохід;
- записи фіксованої і змінної довжини;
- ODBC драйвер в комплекті з вихідним текстом;
- гнучка система привілеїв і паролів;
- до 16 ключів в таблиці. Кожен ключ може мати до 15 полів;
- заснована на потоках, швидка система пам'яті;
- всі операції роботи з рядками не звертають уваги на регістр символів в оброблюваних рядках.

3.3 Побудова діаграми компонентів

Діаграма компонентів – стратегічна структурна діаграма, яка ілюструє розбиття системи на структурні компоненти і зв'язки між ними. Структурними компонентами можна вважати: файли, бібліотеки, модулі, виконувані файли (exe-файли) та пакети.

На рисунку 3.1 зображена діаграма компонентів системи моніторингу банківських послуг для фізичних осіб.

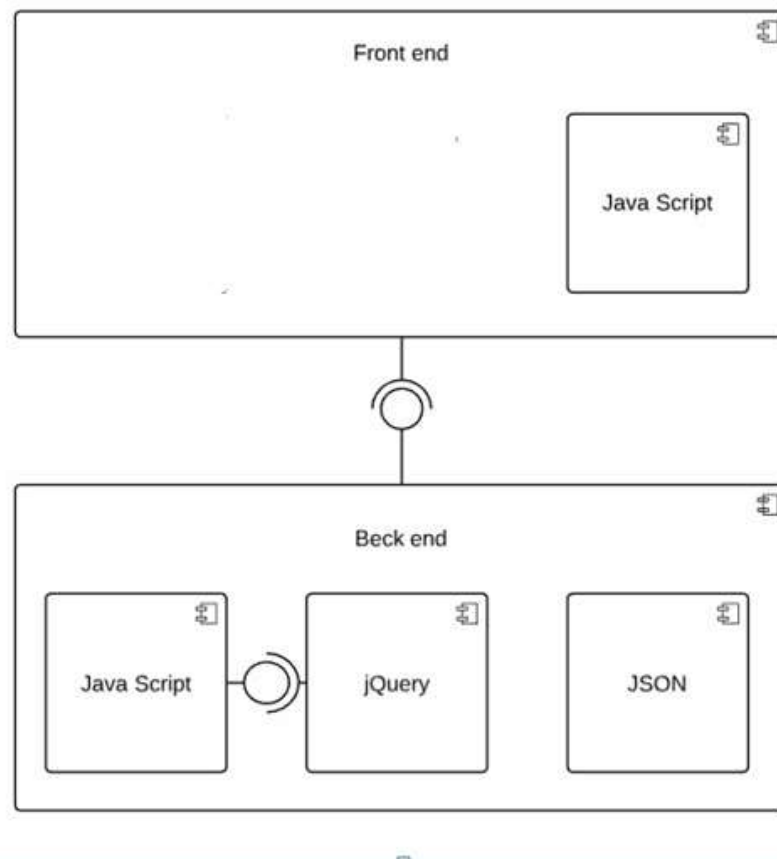


Рисунок 3.1 – Діаграма компонентів системи моніторингу банківських послуг для фізичних осіб

Front end це інтерфейс для взаємодії користувача та програмно-апаратної частини системи. При front end розробці були використані три компоненти для створення зручного користувацького інтерфейсу: PHP та Java Script. JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування, яка найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

Back end це частина системи, яка є невидимою для користувача, але відповідає за роботу всієї системи. При back end розробці були використані три компоненти: JavaScript, jQuery та JSON. jQuery – це бібліотека JavaScript, яка сфокусована на взаємодії JavaScript та HTML. Синтаксис jQuery був розроблений, щоб збільшити зручність орієнтування у навігації, за допомогою легкого доступу до елементів DOM, створенню анімації, обробки подій і розробки AJAX-застосунків. jQuery також надає

можливості розробникам, для створення плагінів у верхній частині бібліотеки JavaScript. Використання цих об'єктів дозволяє розробникам створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. JSON (англ. JavaScript Object Notation) — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, який може бути прочитаним людиною. Такий формат дозволяє описувати об'єкти та інші структури даних, тому він головним чином використовується для передачі структурованої інформації через мережу. JSON, який знайшов своє головне призначення у написанні веб-програм, а саме при використанні технології AJAX, виступає як заміна XML під час асинхронної передачі структурованої інформації між клієнтом та сервером.

3.4 Опис використаних бібліотек та модулів системи моніторингу банківських послуг для фізичних осіб

JavaScript – назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови – написання сценаріїв для веб-сторінок, але, також, використовується для впровадження сценаріїв керування об'єктами вбудованими в інші програми.

Незважаючи на схожість назв, JavaScript та мова програмування Java мають дуже мало спільного. Права на назву JavaScript належать компанії Sun Microsystems .

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів, підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має ряд властивостей, притаманних функціональним мовам – функції як об'єкти першого рівня, об'єкти як списки, каррінг (currying), анонімні функції, замикання (closures), що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів;

- функції як об'єкти першого класу;
- обробка винятків;
- автоматичне приведення типів;
- автоматичне прибирання «сміття»;
- анонімні функції.

JavaScript містить декілька вбудованих об'єктів: Global, Object, Error, Function, Array, String, Boolean, Number, Math, Date, RegExp. Крім того, JavaScript містить набір вбудованих операцій, які, строго кажучи, не обов'язково є 36 функціями або методами, а також набір вбудованих операторів, що управляють логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто в кінцевому рахунку успадкований від C), але спрощений порівняно з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, приміром, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може стояти в тексті програми після неї

3.5 Розробка та опис програмних модулів системи моніторингу банківських послуг для фізичних осіб

Для реалізації системи, що об'єднує описані раніше пристрої в єдину мережу, було спроектовано серверну архітектуру. Вона містить наступні процеси:

Розробка сервера

Цей модуль призначений для отримання списку банківських послуг відповідно до заданих параметрів. Для вибору обробника запиту в Go використовується HTTP-мультиплексор. У деяких бібліотеках він називається «muxer» або «router», але суть та ж. Мультиплексор вибирає обробник на основі аналізу шляху запиту. Якщо вам потрібна просунута підтримка маршрутизації, слід звернутися до сторонніх бібліотекам. Гідні альтернативи - gorilla / mux і go-chi / chi - дозволяють простим чином реалізувати проміжну обробку, налаштувати wildcard-маршрутизацію і вирішити ряд інших завдань. Однак важливіше те, що вони сумісні зі стандартними HTTP-обработчиками. Таким чином, збережеться простота коду і можливість його

безболісного зміни в майбутньому. Зазвичай вони використовують нестандартні рішення, що ускладнює інтеграцію зі стандартними обробчиками. Як правило, комбінації стандартної бібліотеки і легкого маршрутизатора буде досить для розробки більшості додатків. Для реалізації даного модулю було використано API (рисунок 3.2).

Удалить 3427	1	2016-05-19 16:36:30	2016-05-19 13:36:30	<blockquote>О том, что автомобиль уже давно не явл...	Как лучше всего оформить автокредит		publish	open
Удалить 3428	1	2016-05-19 16:36:30	2016-05-19 13:36:30	<blockquote>О том, что автомобиль уже давно не явл...	Где лучше всего оформить автокредит		inherit	closed
Удалить 3430	1	2016-05-19 16:48:25	2016-05-19 13:48:25	<blockquote>О том, что автомобиль уже давно не явл...	Как лучше всего оформить автокредит		inherit	closed
Удалить 3433	1	2016-06-07 19:01:20	2016-06-07 16:01:20	<blockquote>О том, что автомобиль уже давно не явл...	Как лучше всего оформить автокредит		inherit	closed
Удалить 3435	1	2016-06-14 14:52:03	2016-06-14 11:52:03	Современный рынок кредитования весьма развит...	Что такое потребительский кредит?		publish	open
Удалить 3436	1	2016-06-14 14:52:03	2016-06-14 11:52:03	<h2>Что такое потребительский кредит</stro...	Что такое потребительский кредит?		inherit	closed
Удалить 3437	1	2016-06-14 15:54:00	2016-06-14 12:54:00	Современный мир не стоит на месте. Во всех сфе...	Какую банковскую карту можно назвать кредитной		publish	open
Удалить 3438	1	2016-06-14 15:54:00	2016-06-14 12:54:00	Современный мир не стоит на месте. Во всех сфе...	Какую банковскую карту можно назвать кредитной		inherit	closed
Удалить 3439	1	2016-06-14 15:55:15	2016-06-14 12:55:15	<h2>Что	Что такое потребительский кредит?		inherit	closed

Рисунок 3.2 – Результат роботи модулю парсингу банківських послуг за критеріями

Серед параметрів якими можна фільтрувати дані з допомогою програмного модулю парсингу банківських послуг за критеріями, можна визначити такі:

- Категорія
- Регіон
- Валюта
- Опис банку який надає послугу
- Посилання на веб-сайт банку
- Емблема банку
- Назва банку
- Ідентифікаційний номер

Результатами роботи даного модулю є список id банків записаних в базу даних.

Реалізований модуль включає себе клас Parsing, та містить наступні методи:

- downloadCompanies – заносить всі банківські послуги в БД

- getCompanies – отримуємо всі банківські послуги
- getImages – завантажує всі картинки

Одним із аспектів роботи програмного модулю є використання бази даних MySQL. Для взаємодії з базою даних розроблено програмний модуль, який включає в себе клас Database.

Приклад використання бази даних MySQL в системі моніторингу банківських послуг для фізичних осіб зображено на рисунку 3.3.

```
//insert companies into the database
$mysqli->query("INSERT INTO `wp_posts` (`post_author`, `post_title`, `post_content`, `po

//select Id of last record
$last_id = $mysqli->query("SELECT `ID` FROM `wp_posts` WHERE `post_type` LIKE 'post' ORDER B
$last = $last_id->fetch_assoc();
echo $last["ID"];

//insert images of companies into the database
$mysqli->query("INSERT INTO `wp_posts` (`post_author`, `post_date`, `post_date_gmt`, `p
VALUES ('1', '".date('Y-m-d H-i-s')."', '".date('Y-m-d H-i-s')."', '".substr($name_image,

//insert countries into the database

if(count($country[$i])>0){
    $countries = "";
    foreach($country[$i] as $value){
        $countries .= $value.", ";
    }
    $countries = trim($countries, ", ");
}
else{
    $countries = "";
}
}
```

Рисунок 3.3 – Приклад використання бази даних MySQL в системі моніторингу банківських послуг для фізичних осіб

Даний клас містить наступні методи:

- __construct – конструктор класу у якому встановлюються з'єднання з базою даних та відбувається налаштування бази даних
- query_all – даний метод дозволяє отримати всі записи з таблиці з використанням SQL
- query_one – метод дозволяє отримати один запис з використанням SQL
- get_all – отримання всіх записів з конкретної таблиці
- get_one – метод дозволяє отримати один запис з таблиці по id

- get_all_by_param – отримуємо всі записи по заданному параметру
- get_one_by_params – отримання одного запису по заданих параметрах
- get_all_by_params – отримання всіх записів по декількох параметрах
- insert – вставлення запису в таблицю бази даних
- del – видалення запису з таблиці бази даних
- update – оновлення запису в таблиці бази даних
- last – повертає значення id останнього доданого запису

3.6 Розробка інтерфейсу користувача системи моніторингу банківських послуг для фізичних осіб

Перш ніж переходити безпосередньо до створення інтерфейсу системи, необхідно розробити структуру основних модулів, щоб користувачі мали змогу легко переміщатися між ними, а також, щоб вони у будь-який момент могли повернутися на домашнє вікно.

Перше вікно, в якому опиняється користувач, це домашнє вікно. Воно, без сумніву, повинно містити заголовок, фільтр для пошуку банківських послуг, посилання на сайти банків.

Важливим поняттям в розробці системи є схема навігації, яка визначає, як користувач буде переміщатися між вікнами сторінок. За допомогою схеми навігації користувач отримує уявлення про структуру системи, а також уявлення про те, в якому місці вузла він знаходиться в даний момент. Правильно розроблена схема дозволить користувачеві легко переміщатися між сторінками системи в пошуках потрібної йому інформації.

Після того як розроблена схема навігації, необхідно продумати назви навігаційних посилань, а також заголовки сторінок, на які з їх допомогою здійснюється перехід. Назви посилань повинні відповідати назвам представлених ними матеріалів. У цьому випадку відвідувач системи має можливість визначити, яке посилання треба вибрати для перегляду тієї чи іншої інформації. Навігаційні посилання об'єднують в панелі посилань, звані також панелями навігації. Панелі

посилань – це кілька розташованих поруч посилань, призначених для переміщення між сторінками сайту. Розміщуються панелі посилань, як правило, на сторінках у спеціально відведених для них місцях, званих загальними областями, у верхній частині або з лівого боку сторінки. Панелі посилань можуть бути відображені на веб сторінці у вигляді кнопок або тексту. Панелі посилань можна розташовувати і з правого боку сторінки. Але цього небажано робити, так як при перегляді веб сторінки в оглядачі з неповним відкриттям вікна ця область вікна може виявитися не видна користувачеві.

Структурна схема системи моніторингу банківських послуг для фізичних осіб представлена на рисунку 3.4.

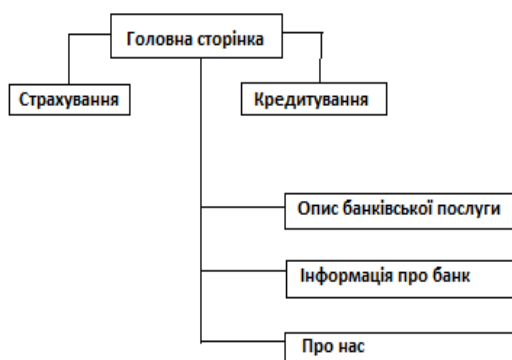


Рисунок 3.4 – Структурна схема системи моніторингу банківських послуг для фізичних осіб

В системі буде розроблена панель навігації, що повинна розміщуватись у правій частині сторінки та бути видною з будь-якого місця сторінки. Також панель, як і вся сторінка повинні, зважаючи на потреби сучасного споживача, легко адаптуватись під екрани різних розмірів (в тому числі екрани мобільних пристроїв).

Панель навігації грає важливу роль на сайті, так як забезпечує переходи між сторінками сайту або на зовнішні ресурси. Розглянемо, як створити простеньку панель навігації.

Фактично панель навігації - це набір посилань, часто у вигляді нумерованого списку. Панелі навігації бувають найрізноманітнішими: вертикальними і горизонтальними, однорівневими і багаторівневими, але в будь-якому випадку в

центрі кожної навігації знаходиться елемент <a>. Тому при створенні панелі навігації ми можемо зіткнутися з низкою труднощів, які випливають з обмежень елемента посилання. А саме, елемент <a> є рядковим, а це значить, що ми не можемо вказати для нього ширину, висоту, відступи. По ширині посилання автоматично займає те місце, яке їй необхідно.

Панель навігації повинно містити наступні розділи:

- Кредитування
- Страхування

Приклад сторінки банківських послуг зображена на рисунку 3.5.

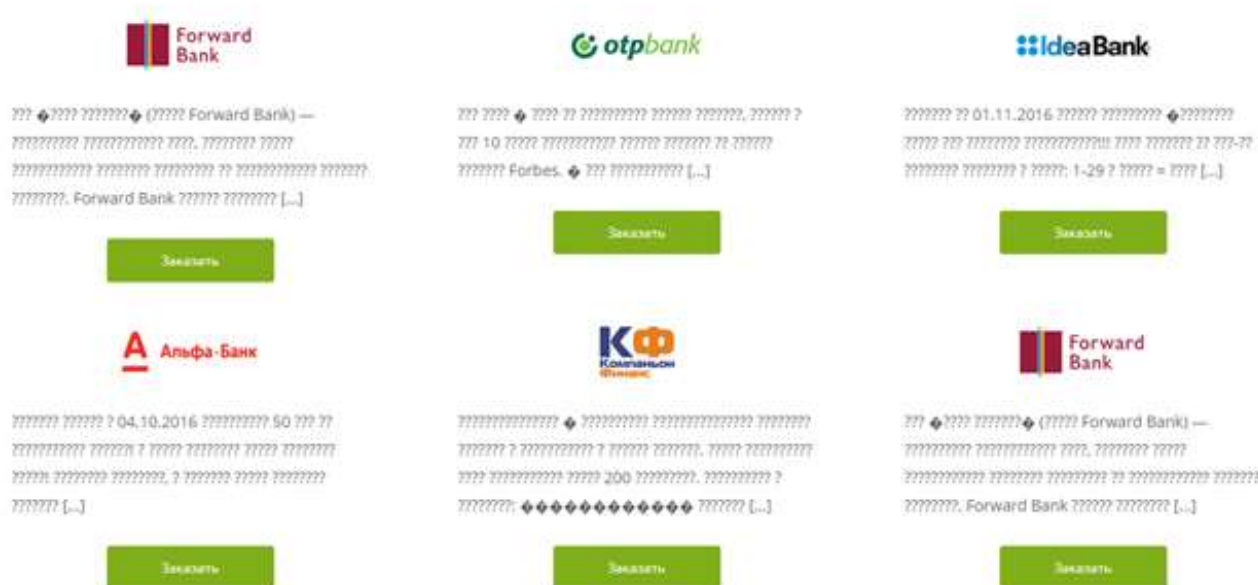


Рисунок 3.5 – Сторінка банківських послуг

При реалізації інтерфейсної частини було використано технологію AJAX. AJAX це підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звітти довантажує потрібні користувачу дані. AJAX — один з компонентів концепції DHTML (рисунок 3.6).

```

$("#searchgroups").click(function() { //оформити зупинку групи

    var btn = $(this)
    btn.button("loading");

    var data={
        event:      "searchgroups",
        query:      $("#query").val()
    }

    $.ajax({
        type: "POST",
        url: "/ajax/_parse.php?callback=?",
        data: data,
        dataType: "jsonp",
        cache: false,
        timeout: 30000000000,
        beforeSend: function() { $("#loading").fadeIn("fast"); },
        success: function(data) {
            console.log();
            if (data.type=="alert") {
                $("#place_to_message").html("<div class='alert alert-warning fade in'><button type='button' class='close'>
            }

            if (data.type=="info") {
                $("#result").html(data.result);
            }

            btn.button("reset");
        }
    });
}

```

Рисунок 3.6 – Приклад використання технології ажах

Використання цих підходів дозволяє створювати набагато зручніші веб-інтерфейси користувача на тих сторінках сайтів, де необхідна активна взаємодія з користувачем. AJAX — асинхронний, тому користувач може переглядати далі контент сайту, поки сервер все ще обробляє запит. Браузер не перезавантажує веб-сторінку і дані посилаються на сервер без візуального підтвердження (крім випадків, коли ми самі захочемо показати процес з'єднання з сервером). Використання AJAX стало популярним після того, як компанія Google почала активно використовувати його при створенні своїх сайтів, таких як Gmail, Google Maps і Google Suggest. Створення цих сайтів підтвердило ефективність використання даного підходу. З допомогою даної технології на сайті було реалізовано наступні розділи:

- Пошук банківських послуг за допомогою фільтра;
- Парсинг банківських послуг за критеріями.

3.8 Опис альтернативних підходів, використаних для розробки системи моніторингу банківських послуг для фізичних осіб

Під час розробки системи було розглянуто два альтернативні підходи реалізації функцій системи моніторингу банківських послуг для фізичних осіб.

Перший був здійснений при розробці веб-серверу. Вибір постав перед двома мовами програмування: JS та PHP. Розглядаючи різні аспекти використання PHP, можна виділити такі основні її переваги: традиційність, простота, ефективність, безпека, гнучкість. PHP може бути вбудована безпосередньо в html-код сторінок, які коректно обробляються PHP-інтерпретатором. PHP містить величезну кількість різних функцій, що позбавляє нас необхідності писати багаторядкові скрипти для виконання простого завдання. Головне для розробника – правильно вибрати функцію відповідно до конкретного завдання. Крім того, не потрібно завантажувати бібліотеки, вказувати спеціальні параметри компіляції.

Другий був здійснений при розробці бази даних. Вибір постав перед двома системами керування базами даних: PostgreSQL та MySQL. Серед основних переваг MySQL відзначають наступні:

Масштабованість. MySQL може підтримувати роботу БД значних розмірів, що підтверджують її реалізації в Yahoo!, Google, HP, Associated Press. Згідно документації, що додається до MySQL, деякі БД, що використовуються компанією MySQL AB (розробником MySQL), зберігають до 50 млн. записів.

Переносність. MySQL працює на різних платформах, серед яких Unix, Linux, Windows, OS/2, Solaris, Mac OS. Окрім того, MySQL працює на різних платформах.

Зв'язаність. MySQL має мережеву структуру. До MySQL можна одержувати доступ із будь-якої точки Internet кільком користувачам одночасно. MySQL має цілий ряд програмних інтерфейсів додатків (Application Programming Interface – API), які дозволяють встановлювати з'єднання з MySQL із додатків, написаних на таких мовах як C, C++, Perl, PHP, Java, Python. **Безпека.** MySQL має систему контролю доступу до даних, забезпечує шифрування даних при передаванні.

Швидкість функціонування. Зручність експлуатації. MySQL досить зручно встановлюється та реалізується, легко адмініструється. Відкритий код.

3.9 Тестування системи моніторингу банківських послуг для фізичних осіб та аналіз результатів її роботи

Тестування програмного забезпечення (англ. Software Testing) — це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки. Тестування програмного забезпечення - процес перевірки відповідності заявлених до продукту вимог і реально реалізованої функціональності, здійснюваний шляхом спостереження за його роботою в штучно створених ситуаціях і на обмеженому наборі тестів, обраних певним чином. Може оцінюватись:

- відповідність вимогам, якими керувалися проєктувальники та розробники
- правильна відповідь для усіх можливих вхідних даних
- виконання функцій за прийнятний час
- практичність
- сумісність з програмним забезпеченням та операційними системами
- відповідність задачам замовника.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, тому стратегія тестування полягає в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат програмне забезпечення (ПЗ) тестується стандартним виконанням програми з метою виявлення багів (помилки або інших дефектів). Тестування ПЗ може надавати об'єктивну, незалежну інформацію про якість ПЗ, ризики відмови, як для користувачів, так і для замовників. Тестування може проводитись, як тільки створено виконуваний код (навіть частково завершений). Процес розробки зазвичай передбачає, коли та як буде відбуватися тестування. Наприклад, при поетапному процесі, більшість тестів відбувається після визначення системних вимог і тоді вони реалізуються в тестових програмах. На противагу цьому, відповідно до вимог гнучкої розробки ПЗ, програмування і тестування часто відбувається одночасно.

Тестувати ми будемо через різні браузеры для коректної роботи. Запускаємо браузер .На початку користування сайтом виберемо мову локалізації розробленої системи та будемо тестувати наш сайт.

На даний момент сайт працює на трьох мовах (англійська, українська, російська). Після вибору мови, сторінка перезапущається і пристає перед нами на потрібній мові. На головній сторінці ми можемо задати параметри пошуку потрібних послуг.

Таким чином користувач отримує всю інформацію про послуги, які потрапили під його запит. Також користувачу надається інформація про контакти такі як номер телефону, адреса, емейл та веб-сайт. Це дає можливість потенційному клієнту зв'язатися з банком тим методом, який йому найбільш зручний. Так як програмний модуль працює у режимі веб-серверу, то інтерфейс представлений у вигляді html-сторінок.

Сторінка із заданням параметрів для пошуку банківських послуг, зображена на рисунку 3.7.



Рисунок 3.7 – Сторінка із заданням параметрів

Да даній сторінці ми зазначаємо тип послуги як «кредит». Після чого, зазначаємо, що нас цікавлять кредитні карти. Обравши потрібні критерії користувачу залишається натиснути пошук та ПЗ підбере всі варіанти, які попадають під запит користувача з БД та виведе інформацію про них на сторінці.

Результат роботи програмного забезпечення зображено на рисунку 3.8.



Рисунок 3.8 – Результат роботи модулю парсингу банківських послуг за критеріями

Програмне забезпечення орієнтоване перш за все на швидкість результатів. Відповідною задачею, поставленою для виконання в даному курсовому проєкті, є досягнення максимальної швидкості роботи програмного забезпечення.

Для перевірки працездатності програмного забезпечення було вибрано «інвестиція», для перевірки швидкодії. Потім ми вибрали «форекс», після чого з'явився список банків.

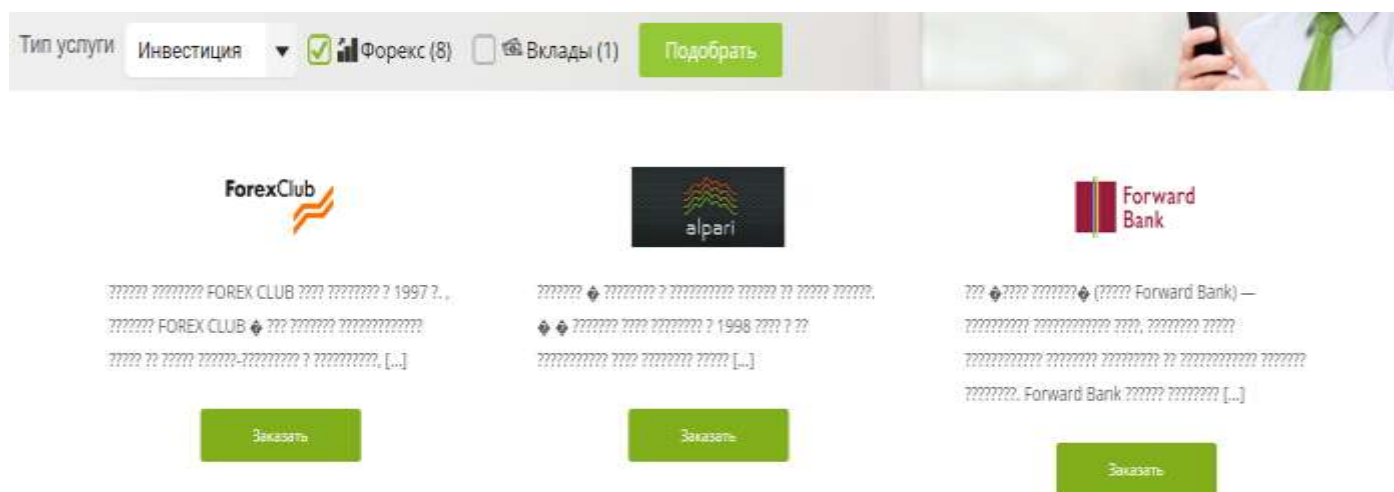


Рисунок 3.9 – Тестування системи моніторингу банківських послуг для фізичних осіб

Під час проведення експериментів було проаналізовано більше 30 різних варіантів вибірки.

3.10 Висновок

У даному розділі обґрунтовано вибір мови програмування PHP та бази даних. Обґрунтовано необхідність використання технології аґах для побудови користувацького інтерфейсу системи моніторингу банківських послуг для фізичних осіб, за яких веб-сторінка, не перезавантажуючись надсилає запит на сервер.

Розроблено алгоритми системи моніторингу банківських послуг для фізичних осіб.

Описана програмна реалізація системи моніторингу банківських послуг для фізичних осіб. Побудована та описана діаграма компонентів, описані використані бібліотеки та модулі а також описані альтернативні підходи реалізації програмного рішення.

Дане програмне забезпечення дозволяє виконувати пошук серед всіх банківських послуг за критеріями, які задає користувач найбільш вигідні пропозиції. Все це дозволило додати до розробленого програмного забезпечення 2 додаткові функції, яких немає у програм аналогів.

Також було проведено тестування розробленого програмного забезпечення. Під час проведення експериментів було проаналізовано більше 30 різних варіантів вибірки. Доведено, що програма працює коректно при різному заваданні початкових критеріїв відбору банківських даних.

4 ЕКОНОМІЧНА ЧАСТИНА

Темою магістерської кваліфікаційної роботи є «Інформаційна технологія моніторингу банківських послуг для фізичних осіб». За цією темою в економічній частині проводяться розрахунки економічних показників на розробку програмного продукту та впровадження його на ринок аналогічних товарів.

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів. Такими експертами будуть Озеранський В.С. та Яровий А.А.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними у таблиці 4.1, за 5-ти бальною шкалою.

Таблиця 4.1 – Критерії оцінювання комерційного потенціалу розробки та їх бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри- те- Рій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта	
	1. Озеранський В.С.	2. Яровий А.А.
	Бали, виставлені експертами:	
1	4	4
2	0	1
3	4	4
4	2	3
5	4	3
6	2	3
7	2	3
8	3	3
9	4	4
10	4	4
11	4	3
12	4	4
Сума балів	СБ ₁ = 37	СБ ₂ = 39
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 38$	

Отже, з отриманих даних таблиці 4.1 видно, що рівень потенціалу нової розробки – вище середнього. Дана розробка є конкурентоспроможною з аналогами, так як для її розробки було проаналізовано недоліки та переваги аналогових продуктів, та на основі цього впроваджену у розробку. Вона має соціологічний вплив, так як покращує ефективність контролю бюджету користувачів, що є важливим у житті людей з обмеженими фінансами.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою (4.1):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M- місячний посадовий оклад конкретного розробника;

T_p - кількість робочих днів у місяці, $T_p = 20$ днів;

t - число днів роботи розробника, t = 37 днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.3.

Таблиця 4.3 – Розрахунки основної заробітної плати

Працівник	Оклад M, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	7500	375	4	1500
Інженер-програміст	9600	480	33	15840
Всього:				17340

Розрахуємо додаткову заробітну плату, вона розраховується як 10-12% від суми основної заробітної плати всіх розробників:

$$Z_{\text{дод}} = 0,1 \cdot 17340 = 1734 \text{ (грн.)} \quad (4.2)$$

Нарахування на заробітну плату $H_{\text{зп}}$ для працівників бюджетної сфери становить 22% від суми основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (4.3)$$

$$H_{\text{зп}} = (17340 + 1734) \cdot \frac{22}{100} = 4196,28 \text{ (грн.)}$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{C \cdot N_a}{100} \cdot \frac{T}{12}, \quad (4.4)$$

де C – балансова вартість обладнання, грн;

N_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання ($T=2$ міс.).

Таблиця 4.4 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	12700	25	2	529,1
Мобільний пристрій	4300	15	2	107,5
Всього:				636.6

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n N_i \cdot C_i \cdot K_i, \quad (4.5)$$

де n – кількість комплектуючих;

N_i - кількість комплектуючих i -го виду;

C_i – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (приймемо $K_i = 1,1$).

Таблиця 4.5 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
USB – microUSB кабель	шт.	75	1	75
Пачка паперу	уп.	100	1	100
Ручка	шт.	4	1	4
Всього з урахуванням транспортних витрат				196,9

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi} ; \quad (4.6)$$

де V – вартість 1кВт-години електроенергії ($V=2.35$ грн/кВт);

Π – установлена потужність комп'ютера ($\Pi=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=296$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,7$).

$$V_e = 2,35 \cdot 0,6 \cdot 296 \cdot 0,7 = 292,15 \text{ (грн.)}$$

Розрахуємо інші витрати $V_{ін}$. Інші витрати I_v можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p). \quad (4.7)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 * (17340 + 1734) = 19074 \text{ (грн.)}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = Z_o + Z_d + H_{зп} + A + K + B_e + I_B$$

(4.8)

$$B = 17340 + 1734 + 4196,28 + 636,6 + 196,9 + 292,15 + 19074 = 43\,469,83 \text{ (грн.)}$$

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha} \quad (4.9)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{43\,469,83}{1} = 43\,469,83 \text{ грн.}$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \quad (4.10)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{43\,469,83}{0,9} = 48\,299,8 \text{ грн.}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо кількісно, яку вигоду можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Основні позитивні результати від впровадження розробки очікуються протягом 3-х років після її впровадження. Одним із основних позитивних результатів є зростання величини прибутку.

При реалізації результатів наукової розробки покращується якість програмного продукту, що дозволяє підвищити ціну його реалізації на 150 грн. Кількість одиниць реалізації програмного засобу також збільшиться: протягом першого року — на 700 шт., протягом другого року — ще на 450 шт., протягом третього року — ще на 300 шт.

Реалізація продукції до впровадження результатів наукової розробки складала 50 шт., а ціна — 250 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового за такою формулою 4.11:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \Delta N)_i \quad (4.11)$$

де $\Delta\Pi_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт 0,8333.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $= 0,2...0,3$;

v – ставка податку на прибуток. $v = 18\%$.

Тоді, збільшення чистого прибутку підприємства протягом першого року складе:

$$\Delta\Pi_1 = [150 \cdot 50 + (250 + 150) \cdot 700] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 39290,09 \text{ грн}$$

Протягом другого року:

$$\Delta\Pi_2 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 61\,710 \text{ грн}$$

Протягом третього року:

$$\Delta\Pi_3 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450 + 300)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 79900 \text{ грн}$$

Отже, протягом трьох років підприємство може розраховувати на збільшення чистого прибутку від реалізації наукової розробки.

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{\text{абс}}$ вкладених інвестицій розраховується за формулою:

$$E_{\text{абс}} = (\text{ПП} - \text{PV}), \quad (4.12)$$

де ПП – приведена вартість всіх чистих прибутків, які отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $\text{PV} = \text{ЗВ} = 48\,299,8$ грн.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, рисунок 4.1.



Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів наукової роботи

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки “0”.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{48\,063,6}{(1+0,1)^0} + \frac{39\,290,09}{(1+0,1)^2} + \frac{61\,710}{(1+0,1)^3} + \frac{79\,900}{(1+0,1)^4} = 181\,474,87 \text{ грн.}$$

Тоді розрахуємо $E_{\text{абс}}$:

$$E_{абс} = 181\,474,87 - 48\,299,8 = 133\,175,07 \text{ грн.}$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_B за формулою:

$$E_B = \sqrt[T]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.14)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{133\,175,07}{48\,063,6}} - 1 = 0,56 \text{ або } 56\%$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3$$

Оскільки $E_B = 56\% > \tau_{\text{мін}} = 0,3 = 30\%$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ розраховується за формулою:

$$T_{ок} = \frac{1}{E_B}, \quad (4.16)$$

Якщо $T_{ок} < 3 \dots 5$ -ти років, то фінансування наукової розробки є доцільним.

$$T_{ок} = \frac{1}{0,56} = 1,7 \text{ років}$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним, оскільки $T_{ок} < 3$ років.

4.5 Висновок

Результати проведених розрахунків дають можливість зробити висновок про доцільність розробки та впровадження нашої наукової роботи. Це підтверджують такі показники як:

- абсолютна ефективність вкладених інвестицій, яка дорівнює 133 175,07 грн., що є більшим 0 і вказує на те, що інвестор може бути зацікавленим у нашій розробці;
- відносна ефективність наукової розробки становить 56%, що є вищим за мінімальну ставку дисконтування (30%), тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит;
- термін окупності вкладених у реалізацію наукового проекту інвестицій складе 1,7 років, що є менше 3-ох і вказує швидко окупність вкладених інвестицій.

Крім того, розраховано, що наукова розробка принесе підприємству додатковий прибуток протягом 3-х років за рахунок покращення її якості порівняно з існуючими аналогами.

Усе це, узятє разом, забезпечує прийняття рішення про доцільність виготовлення нового продукту.

ВИСНОВКИ

В магістерській дипломній роботі було проведено удосконалено інформаційну технологію моніторингу банківських послуг для фізичних осіб, яку можна застосувати в сфері інтернет-банкінгу, наприклад, компаніям які займаються банківськими справами.

В першому розділі було проведено опис предметної області та сфери застосування програмних рішень моніторингу банківських послуг. Виконано порівняльний аналіз програм аналогів, вивчено їх переваги та недоліки. В роботі було удосконалено інформаційну технологію моніторингу банківських послуг для фізичних осіб, яка відрізняється від існуючих аналізом парсингованих даних про банківські послуги, що дозволяє розширити функціональні можливості системи моніторингу банківських послуг для фізичних осіб, зокрема, вибирати банківські послуги, які найбільш вигідні для фізичної особи.

Також було розроблено алгоритм моніторингу банківських послуг для фізичних осіб з вибором найбільш вигідних пропозицій. На основі даного алгоритму було розроблено програмне забезпечення що дозволяє здійснювати моніторинг банківських послуги для фізичних осіб. Дане програмне забезпечення дозволяє виконувати пошук серед всіх банківських послуг за критеріями, які задає користувач найбільш вигідні пропозиції. Все це дозволило додати до розробленого програмного забезпечення 2 додаткові функції, яких немає у програм аналогів.

Результати проведених економічних розрахунків дають можливість зробити висновок про доцільність розробки та впровадження нашої наукової роботи. Абсолютна ефективність вкладених інвестицій складає 133 175,07 грн., що вказує на те, що інвестор може бути зацікавленим у нашій розробці. Відносна ефективність наукової розробки становить 56%, тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит. Термін окупності вкладених у реалізацію наукового проекту інвестицій складе 1,7 років.

Отже, всі задачі, які ставились в магістерській кваліфікаційній роботі виконані, мету роботи досягнуто.

ЛІТЕРАТУРА

1. Іржавський О.Ю., Озеранський В.С. «Розробка веб-сервісу моніторингу банківських послуг» Матеріали 12-ї міжнародної науково-практичної конференції «Інтернет-Освіта-Наука-2020», Вінниця, 2020. [Електронний ресурс]. Режим доступу: <http://ies.vntu.edu.ua/reports/program/WORK-IES-2020.pdf> Дата звернення: листопад 2020
2. Діброва Т. Реалії ринку Інтернет-реклами в Україні: [Про активне використання Інтернету як засобу просування товарів та послуг в Україні] / Т. Діброва, М. Лебеденко. // Маркетинг в Україні. – 2007. – № 2. – с.21 – 24.
3. Лаура Томсон, Люк Веллинг Разработка веб-приложения на PHP и MySQL. - ДиаСофтЮП, 2003. – 872 с.
4. Велихов С. HTML для тех, кто в танке. <http://mediaray-blog.ru>
5. Мержевич В. Справочник по CSS. <http://mediaray-blog.ru>
6. Мазуркевич.А. PHP настольная книга программиста <http://mediaray-blog.ru>
7. Храмов П. Введение в программирование на JavaScript.
8. Матросов А.В., Чаунин М.П. Самоучитель Perl. <http://mediaray-blog.ru>
9. Попов Е.Г. Теоретический курс по PHP и MySQL – Москва, Триумф, 2008. – 513с.
10. Шмидт Г.А. Практическое введение в программирование на JavaScript – Оренбург, 2011. 458с.
11. Шапошников И.Е. PHP 5.3 - Москва, Питер, 2011 г. - 192 с.
12. Пауэрс Д. Adobe Dreamweaver, CSS, Ajax и PHP - Москва, БХВ-Петербург, 2009 г. - 1058 с.
13. Колисниченко Д.Н. PHP 5 и MySQL 5. Разработка Web-приложений - Москва, БХВ-Петербург, 2010 г. - 546 с.
14. Чаффер Д., Шведберг К. Изучаем jQuery 1.3. Эффективная веб-разработка на JavaScript - Москва, Символ-Плюс, 2010 г. - 448 с.
15. Гизберт Д. PHP и MySQL - Санкт-Петербург, ИТ Пресс, 2008 г. - 320 с.

16. Бугаєнко А.В. Обліково-аналітичне забезпечення фінансового моніторингу
17. Сімпсон К. You don't know JS 2019 р – 302 с.
18. Michael McMillan Data Structures and Algorithms with JavaScript – 200 с.
19. Херман Дэвид Сила JavaScript. 68 способов эффективного использования JS; Питер - М., 2015. - 952 с.
20. Чаффер Д. Изучаем jQuery 1.3. Эффективная веб-разработка на JavaScript; Символ-плюс - М., 2015. - 391 с.
21. Шпильман Сью JSTL. Практическое руководство для JSP-программистов; КУДИЦ-Образ - М., 2016. - 272 с.
22. Бибо Бер , Кац Иегуда jQuery. Подробное руководство по продвинутому JavaScript; Символ-плюс - М., 2017. - 624 с.