КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: « Розробка інформаційного забезпечення системи страхування банківських операцій»

Микитенко Сергій Володимирович

(прізвище, ім’я та по батькові студента повністю)

Київ, 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
к.т.н., доцент Гончаренко Т.А.

„___” _____2024 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Розробка інформаційного забезпечення системи страхування банківських операцій»

Виконав: студент 4-го курсу, групи КН-20-1

Спеціальності: 122 «Комп`ютерні науки

Спеціалізація: «Інформаційні управляючі системи та технології»

(шифр і назва напряму підготовки, спеціальності)

Микитенко С. В.

(прізвище та ініціали)

Керівники к.т.н., доц., Гончаренко Т.А., к.т.н., доц., Поплавський О.А.

(прізвище та ініціали)

Рецензент к.т.н., доц. Шабала Є.Є.

(прізвище та ініціали)

Київ, 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Факультет: автоматизації і інформаційних технологій
Кафедра: інформаційних технологій
Освітній рівень: «бакалавр» за ОП
Спеціальність: 122 «Комп`ютерні науки»
Спеціалізація: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ
Завідувач кафедри ІТ
к.т.н., доцент Гончаренко Т.А.

_____
„___” _____2024 року

З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

Микитенко Сергій Володимирович
Тема роботи: Розробка інформаційного забезпечення системи страхування банківських операцій
затверджена наказом ректора КНУБА № 2650/2 від 18.11.2023.
2. Керівники роботи: Гончаренко Тетяна Андріївна, к.т.н., доц., Поплавський Олександр Анатолійович к.т.н., доц..
3. Строк подання студентом роботи до захисту:_____
4. Зміст пояснювальної записки за розділами:
Р.1. Аналіз сучасного стану систем страхування в банківських операціях
Р.2. Проектування інформаційного забезпечення для систем страхування в банківських операціях
Р.3. Реалізація інформаційного забезпечення для систем страхування в банківських операціях
Р.4. Ергономіка інтерфейсу користувача в системах страхування для банківських операцій
5. Інформаційні слайди:
С.1._____
С.2._____
С.3._____

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз сучасного стану систем страхування в банківських операціях	Січень 2024 р.
Р. 2. Проектування інформаційного забезпечення для систем страхування в банківських операціях	Лютий 2024 р.
Р. 3. Реалізація інформаційного забезпечення для систем страхування в банківських операціях	Березень 2024 р.
Р. 4. Ергономіка інтерфейсу користувача в системах страхування для банківських операцій	Травень 2024 р.
Остаточне оформлення роботи	Травень 2024 р.
Направлення роботи на рецензування	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	доц. Ачкасов І.А.		
Прийом програмного продукту	доц. Рябчун Ю.В.		

8. Дата видачі завдання: 18.11.2023

Завідувач _____ Гончаренко Т.А.
(підпис) (прізвище та ініціали)

Керівник _____ Гончаренко Т.А.
(підпис) (прізвище та ініціали)

Керівник _____ Поплавський О.А.
(підпис) (прізвище та ініціали)

Студент _____ Гаранський К.О.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП

Актуальність теми. У мірах стрімкої еволюції банківських операцій та постійної необхідності забезпечення безпеки й ефективності, інформаційні системи страхування набувають ключового значення. Забезпечення інформаційної безпеки та надійності у контексті фінансових трансакцій є викликом для сучасних банківських установ. Розробка інформаційного забезпечення систем страхування у банківських операціях стає важливим етапом у підтримці стійкої та безпечної фінансової інфраструктури.

Ключовою проблемою в цьому контексті є забезпечення конфіденційності, цілісності та доступності фінансової інформації. Зловмисні атаки, експлуатація вразливостей систем, а також несприятливі фактори зовнішнього середовища потенційно можуть порушити навіть найбільш добре структуровані системи. У цьому контексті важливо розглядати розробку інформаційного забезпечення систем страхування не лише як технічну задачу, але і як стратегічну ініціативу, спрямовану на захист інтересів клієнтів та забезпечення стабільності фінансових потоків.

Підходи до розробки інформаційного забезпечення систем страхування включають у себе використання передових методів криптографії, механізмів аутентифікації та авторизації, а також вдосконалення алгоритмів моніторингу та виявлення вразливостей. Інтеграція цих підходів дозволяє створити систему, яка забезпечить високий рівень захисту інформації та дозволить оперативно реагувати на загрози безпеки.

У цьому проекті мета полягає в розробці комплексної системи інформаційного забезпечення, яка враховує специфіку банківських операцій та відповідає вимогам сучасних стандартів безпеки. Це передбачає не лише створення технічно ефективних рішень, але й впровадження проактивних стратегій

моніторингу та аналізу, спрямованих на виявлення потенційних загроз та запобігання їх наслідкам.

Тема дипломного проекту "Розробка інформаційного забезпечення систем страхування у банківських операціях" є досить актуальною, оскільки фінансові установи постійно стикаються з ризиками та необхідністю захисту фінансових активів. З розвитком технологій та зростанням обсягів фінансових транзакцій стає важливим розробка спеціалізованих систем страхування, які забезпечать надійний захист та оптимізацію фінансових операцій у банківській сфері.

Метою даного дослідження є розробка та впровадження інформаційного забезпечення системи страхування, спрямованого на зменшення фінансових ризиків та підвищення ефективності банківських операцій. Конкретні завдання включають аналіз поточного стану систем страхування, проектування необхідних функцій і модулів, розробку та тестування інформаційної системи.

Об’єктом дослідження є системи страхування у банківських операціях. **Предметом дослідження** є розробка та впровадження інформаційного забезпечення цих систем.

Завдання дослідження:

1. Провести аналіз сучасного стану систем страхування в банківських операціях;
2. Розробити проектування інформаційного забезпечення для систем страхування в банківських операціях;
3. Провести реалізацію інформаційного забезпечення для систем страхування в банківських операціях.

Для досягнення поставлених цілей будуть використані наступні **методи:** аналіз фінансових даних, проектування баз даних та інформаційних систем, програмування, тестування та експериментальні методи для оцінки ефективності системи.

Практичне та наукове значення результатів дослідження. Даний проект має наукову цінність через розробку нових методів управління ризиками та оптимізації банківських операцій. Практична цінність полягає у можливості підвищення ефективності та безпеки фінансових транзакцій за допомогою впровадження нової системи страхування.

Розробка інформаційного забезпечення систем страхування у банківських операціях є складною і многогранною задачею, що вимагає інтеграції різноманітних технологій, методів та підходів. Вона також вимагає глибокого розуміння як технічних, так і стратегічних аспектів інформаційної безпеки в контексті банківської діяльності. Цей проект спрямований на створення інформаційного середовища, що дозволить забезпечити надійну захищеність даних та оптимізувати банківські операції з урахуванням сучасних викликів і загроз.

У цьому розділі буде проведений детальний аналіз поточного стану систем страхування у банківських операціях. Він включатиме в себе огляд існуючих методів та технологій, використовуваних у банківській сфері для захисту фінансових активів, а також виявлення недоліків та потенційних напрямків удосконалення.

Цей розділ описуватиме процес проектування інформаційної системи страхування. Він включатиме у себе розробку вимог до системи, проектування архітектури та бази даних, визначення функціональності та інтерфейсів користувача.

У цьому розділі буде описано процес реалізації розробленої інформаційної системи. Це включатиме програмування, тестування та внедрення системи в робоче середовище банківських установ.

Розділ про тестування відобразить методи, процеси та результати тестування інформаційної системи страхування. Буде проведений аналіз якості та надійності системи шляхом різноманітних тестів.

В останньому розділі будуть представлені висновки з роботи, підсумки дослідження, виявлені проблеми та досягнуті результати. Також будуть надані рекомендації щодо подальших кроків у розвитку та вдосконаленні системи страхування у банківських операціях.

1. АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ СТРАХУВАННЯ В БАНКІВСЬКИХ ОПЕРАЦІЯХ

1.1 Огляд існуючих підходів до страхування в банківських установах

У світі банківської сфери існує різноманітність типів страхування, які використовуються для забезпечення безпеки та захисту фінансових інтересів клієнтів. Одним з основних типів є страхування депозитів, яке спрямоване на забезпечення відшкодування втрат, що можуть виникнути внаслідок банкрутства банківської установи або неплатоспроможності. Цей тип страхування дозволяє клієнтам бути впевненими у безпеці своїх фінансових коштів та збереженні їх в цілковитості навіть у випадку фінансових труднощів банку.

Ще одним важливим типом страхування є страхування кредитів, яке надає захист клієнтам від непередбачених обставин, таких як втрата роботи або тимчасові труднощі в погашенні кредитних зобов'язань [14]. Цей тип страхування дозволяє зменшити фінансові ризики для як боржника, так і кредитора, тим самим забезпечуючи стабільність банківського сектора та сприяючи розвитку кредитного ринку.

Крім того, важливим аспектом є страхування ризикованих операцій, яке забезпечує захист від негативних наслідків фінансових транзакцій, таких як валютні ризики, інтересні ризики та інші. Цей тип страхування дозволяє банкам знизити експозицію до ризиків та забезпечити стабільність свого фінансового стану в умовах непередбачуваних ринкових умов.

Варто зазначити, що кожен з цих типів страхування має свої особливості та вимоги, які визначаються конкретними умовами контрактів та регулятивним середовищем. Інноваційні технології та нові підходи до страхування постійно розвиваються, щоб забезпечити ефективний захист фінансових інтересів учасників банківського ринку.

У сучасному банківському середовищі страхування викликає значні особливості та виклики, які потребують уваги та комплексного аналізу. Однією з основних особливостей є постійна зміна умов та умов контрактів страхування, що вимагає від банківських установ не лише гнучкості, а й здатності оперативно реагувати на нові виклики та ризики.

Ще однією важливою особливістю є висока ступінь регулювання та нагляду з боку фінансових установ та державних органів. Банки повинні дотримуватися ряду вимог, що стосуються капіталізації, ліквідності та управління ризиками, що вносять додаткові обмеження та виклики для реалізації програм страхування.

Паралельно з особливостями існує ряд викликів, з якими стикаються банківські установи при впровадженні програм страхування. Один з таких викликів полягає у забезпеченні адекватної оцінки та управління ризиками, пов'язаними зі страхуванням, особливо в умовах нестабільності ринку та економічних турбулентностей.

Інший важливий виклик полягає у виявленні та врахуванні нових та емерджентних ризиків, таких як кіберзлочинність, технологічні збої та кліматичні зміни. Банки повинні постійно оновлювати свої стратегії та процедури страхування, щоб ефективно захищати свої та клієнтські активи в умовах швидко змінюючогося середовища.

У сучасній банківській сфері використання технологій та інструментів страхування є невід'ємною складовою для забезпечення безпеки та надійності фінансових операцій [25]. Одним з ключових технологічних інструментів є системи аналізу даних, які дозволяють банкам ефективно оцінювати ризики та прогнозувати події, що можуть вплинути на їхню діяльність. Використання алгоритмів машинного навчання та штучного інтелекту дозволяє автоматизувати процеси прийняття рішень та реагування на зміни в ринкових умовах.

Для забезпечення безпеки та захисту фінансових активів від кіберзагроз та інших технологічних ризиків банки використовують сучасні кібербезпекові

рішення, включаючи файрволи, антивірусне програмне забезпечення, системи виявлення вторгнень та інші. Ці заходи дозволяють забезпечити надійність та цілісність фінансових даних та операцій в електронному середовищі.

Також важливим інструментом є блокчейн технологія, яка забезпечує безпеку та недоторканність даних шляхом розподіленого зберігання та криптографічного захисту. Банки використовують блокчейн для створення захищених та недійсних доказів, використання яких дозволяє зменшити шахрайство та зловживання у фінансових операціях.

Крім того, розумні контракти на основі блокчейн технології використовуються для автоматизації та стандартизації умов страхових полісів, що дозволяє покращити ефективність та надійність страхових програм. Такі контракти автоматично виконуються при виконанні певних умов, що знижує ризик помилок та зловживань у страхувальній сфері.

У таблиці 1.1 наведено короткий огляд сучасних технологій та інновацій, що використовуються у галузі страхування в банківських установах.

Таблиця 1.1 – Технології та інновації в галузі страхування в банківських установах.

Технологія / Інновація	**Опис**	**Переваги**
Аналітичні системи та алгоритми машинного навчання	Використовуються для оцінки ризиків та прийняття рішень у реальному часі.	- Підвищення ефективності та точності визначення ризиків - Автоматизація процесів прийняття рішень
Блокчейн	Забезпечує недоступність даних і безпеку транзакцій, що важливо для зменшення ризиків кіберзлочинності.	- Надійність та не відмінність даних - Відсутність посередників та ризиків маніпуляції
Цифрові платформи та мобільні додатки	Надають клієнтам зручний доступ до страхових послуг та можливість управління полісами на власних пристроях.	- Зручний та швидкий доступ до інформації - Підвищення клієнтської задоволеності
Інтерфейси користувача	Забезпечують персоналізований та інтуїтивно зрозумілий досвід для клієнтів банківських установ.	- Зменшення часу на навчання користувачів - Підвищення лояльності та відчуття комфорту користувачів

Загалом, використання сучасних технологій та інструментів страхування дозволяє банкам ефективно захищати фінансові активи та забезпечувати безпеку та надійність фінансових операцій в умовах швидкозмінюючогося ринкового середовища.

У сучасному банківському секторі спостерігається постійний наплив інновацій та технологічних змін, що значно впливає на галузь страхування. Однією з головних тенденцій є широке використання аналітичних систем та алгоритмів машинного навчання для оцінки ризиків і прийняття рішень у реальному часі. Технології штучного інтелекту дозволяють банкам автоматизувати процеси оцінки ризиків та реагування на них, що сприяє підвищенню ефективності та точності страхових програм [33].

Ще однією важливою інновацією є використання блокчейн технологій для підвищення безпеки та надійності страхових транзакцій. Блокчейн дозволяє забезпечити недоступність даних, що робить їх невідмінними від впливу зовнішніх факторів. Це особливо важливо в умовах зростаючої кількості кіберзлочинності та кібератак на банківську інфраструктуру.

Також слід відзначити розвиток цифрових платформ та мобільних додатків, які надають клієнтам зручний доступ до страхових послуг та можливість управління своїми полісами. Інноваційні рішення в галузі інтерфейсів користувача дозволяють забезпечити персоналізований та інтуїтивно зрозумілий досвід для клієнтів банківських установ.

Напрямок розвитку інновацій в галузі страхування в банківських установах визначається потребами ринку та стратегіями конкурентів. Застосування передових технологій та інноваційних підходів дозволяє банкам зберігати конкурентну перевагу та забезпечувати високий рівень обслуговування для своїх клієнтів.

1.2 Технологічні аспекти страхування в банківських операціях

Використання передових технологій у банківській сфері страхування вимагає постійного оновлення технічної інфраструктури для забезпечення ефективного функціонування систем. Аналіз сучасного стану технологічної інфраструктури в цьому контексті стає критичним завданням, оскільки від нього залежить успішність впровадження та функціонування страхових продуктів та послуг.

На сьогоднішній день, банківська сфера страхування активно використовує інформаційні технології для автоматизації процесів управління ризиками та створення інформаційних систем, які забезпечують надійність і безпеку операцій. Основними складовими сучасної технологічної інфраструктури є великі обсяги даних, хмарні технології, штучний інтелект та блокчейн.

Зростаюча кількість та різноманітність даних, що генеруються в банківській сфері страхування, потребує використання потужних технологій обробки та аналізу даних. Великі обсяги даних вимагають використання спеціалізованих систем зберігання та обробки даних, таких як системи великих даних (Big Data), щоб забезпечити швидку та ефективну обробку інформації.

Хмарні технології відіграють ключову роль у покращенні доступності та масштабованості інформаційних систем. Вони дозволяють зберігати та обробляти дані на віддалених серверах, що забезпечує гнучкість та можливість швидкої реакції на зміни в обсязі та потребах в обробці даних.

Блокчейн-технології стають все більш популярними у банківській сфері страхування, оскільки вони забезпечують безпеку та недоторканність даних. Використання блокчейну для збереження та обміну інформацією дозволяє знизити ризики шахрайства та підвищити довіру між сторонами в операціях страхування.

В цілому, аналіз сучасного стану технологічної інфраструктури в банківській сфері страхування свідчить про постійний розвиток і вдосконалення систем,

спрямованих на забезпечення надійності, ефективності та безпеки страхових операцій [14].

Інтеграція передових технологій у процеси страхування в банківських установах є складним, але необхідним завданням, спрямованим на підвищення ефективності та надійності операцій. Для досягнення цієї мети потрібно здійснити інтеграцію різних технологічних рішень та систем в єдину інформаційну архітектуру банку.

Першим кроком у процесі інтеграції є аналіз потреб банківської установи та вибір оптимальних технологій, що відповідають її специфіці та стратегії розвитку. Це можуть бути системи аналізу даних, моделі штучного інтелекту для прогнозування ризиків, чи блокчейн-платформи для забезпечення безпеки та відстеження операцій.

Після вибору технологій необхідно здійснити їх інтеграцію з існуючими системами банку, такими як CRM (Customer Relationship Management) та ERP (Enterprise Resource Planning). Це може вимагати розробки спеціалізованих API (Application Programming Interface) для забезпечення взаємодії між системами, а також налаштування інтеграційних модулів.

Один з ключових аспектів інтеграції технологій в процеси страхування - це забезпечення взаємодії між різними внутрішніми та зовнішніми системами, такими як платіжні шлюзи, партнерські платформи та сторонні сервіси. Це дозволяє оптимізувати процеси обробки страхових виплат, підвищує точність та швидкість обробки даних.

Окрім того, інтеграція технологій також передбачає впровадження механізмів моніторингу та аналізу результатів використання нових рішень. Це дозволяє вчасно виявляти проблеми та вдосконалювати процеси з використанням зібраної інформації.

Забезпечення кібербезпеки у банківських операціях страхування є надзвичайно важливим аспектом в контексті використання інформаційних

технологій. Постійна загроза кібератак та несанкціонованого доступу до конфіденційної інформації вимагає впровадження комплексних заходів з захисту даних та мережевих ресурсів.

Для ефективного забезпечення кібербезпеки в банківській сфері страхування необхідно використовувати передові технології шифрування даних та захисту від шкідливих програм. Використання сучасних шифрувальних алгоритмів та протоколів забезпечує захист інформації під час її передачі та зберігання в базах даних.

Окрім цього, важливо регулярно проводити аудит безпеки систем та мереж, щоб виявляти та усувати можливі вразливості та порушення безпеки. Використання сучасних методів аналізу та виявлення вторгнень дозволяє оперативно реагувати на потенційні загрози та запобігати їх наслідкам.

Паралельно з цим, необхідно надавати співробітникам банку відповідну підготовку з питань кібербезпеки та створювати відповідні політики та процедури для захисту інформації. Освіченість персоналу та встановлення правил внутрішньої безпеки є ключовими елементами у забезпеченні безпеки банківських операцій.

Перспективи розвитку технологічних рішень у сфері страхування банківських операцій відкривають широкі можливості для вдосконалення процесів управління ризиками та надання клієнтам кращих страхових продуктів [24]. Одним із ключових напрямків розвитку є подальше використання штучного інтелекту та машинного навчання для автоматизації процесів прийняття рішень та аналізу даних.

Штучний інтелект в сфері страхування може бути використаний для прогнозування ризиків, розробки персоналізованих страхових продуктів, а також для виявлення та запобігання шахрайству. Застосування алгоритмів машинного навчання дозволяє аналізувати великі обсяги даних та виділяти патерни, що допомагають у прийнятті ефективних рішень.

Далі, розвиток блокчейн-технологій може змінити підхід до зберігання та обміну даними в страховій сфері. Використання блокчейну для створення децентралізованих та недоторканих систем забезпечення даних може підвищити рівень безпеки та довіри між сторонами.

Крім того, розвиток інтернету речей (IoT) може вплинути на страховий ринок, дозволяючи страховим компаніям отримувати доступ до різних даних з об'єктів страхування та в реальному часі моніторити їх стан. Це відкриває можливості для введення нових страхових продуктів, які базуються на конкретних ризиках та потребах клієнтів.

У підсумку, перспективи розвитку технологічних рішень у сфері страхування банківських операцій обіцяють значні переваги для страхових компаній та їх клієнтів, сприяючи підвищенню ефективності, безпеки та зручності у веденні страхового бізнесу.

1.3 Потреби та очікування користувачів від систем страхування в банківських операціях

В реалізації банківських операцій без страхового покриття виявлено низку системних проблем, які безпосередньо впливають на користувачів. Першою серйозною проблемою є відсутність вбудованих механізмів захисту, що призводить до великої вразливості фінансових операцій перед різноманітними загрозами, такими як кібератаки, шахрайство та технічні помилки. Порушення цілісності та конфіденційності даних стають особливо нагальними проблемами відсутності адекватного страхового покриття [18]. У відсутності цього, користувачі ризикують втратити доступ до своїх фінансових активів через несанкціонований доступ або зловмисні дії.

Додатково, недостатня увага до захисту від технічних помилок та інфраструктурних проблем може призвести до втрат фінансових активів або навіть

до тимчасової недоступності банківських сервісів для користувачів. Ці ризики стають особливо актуальними в умовах постійного розвитку кіберзлочинності та технічних збоїв. Без страхового покриття, користувачі не мають надійного механізму компенсації збитків у разі таких подій.

Крім того, відсутність адекватного страхового покриття обмежує можливості користувачів у плануванні та управлінні ризиками. Недостатній вибір страхових продуктів та відсутність індивідуального підходу до кожного клієнта може призвести до недооцінки потенційних загроз і втрат для їхніх фінансових активів. Таким чином, відсутність системи страхового покриття у банківських операціях створює значні ризики для користувачів і перешкоджає їхній фінансовій безпеці та стабільності.

Під час аналізу очікувань користувачів від систем страхування в банківських операціях виявлено високу вимогливість до рівня захисту та стабільності. Користувачі очікують, що страхові продукти будуть забезпечувати надійний захист їхніх фінансових активів від широкого спектру загроз, включаючи кіберзлочинність, крадіжки, технічні помилки та інші ризики. Вони сподіваються, що система страхування буде оперативно реагувати на можливі загрози та надавати вчасну компенсацію у разі виникнення непередбачених ситуацій.

Крім того, користувачі очікують від систем страхування гнучкості та індивідуалізації. Вони бажають мати можливість обирати страхові продукти, які відповідають їхнім конкретним потребам та ризикам. Це може включати можливість налаштування обсягу страхового покриття, вибір оптимальних тарифних планів та доступ до додаткових послуг і сервісів.

Зокрема, користувачі очікують від систем страхування високого рівня прозорості та доступності інформації. Вони бажають мати чітку уяву про умови та обсяг страхового покриття, а також про процедури виплати компенсацій у разі потреби. Доступ до зрозумілої та достовірної інформації є ключовим фактором для підтримки довіри користувачів до системи страхування.

Нарешті, користувачі очікують від систем страхування високого рівня сервісу та підтримки. Вони сподіваються на швидке та професійне реагування на їхні запити та вирішення будь-яких проблем чи питань, що виникають у процесі використання страхових послуг. Забезпечення зручного та ефективного взаємодії з системою страхування є важливим аспектом задоволення потреб користувачів у цій сфері.

Відповідно до потреб та очікувань користувачів, пропонуються декілька стратегій для вдосконалення систем страхування в банківських операціях. Перш за все, необхідно розробити та впровадити нові страхові продукти, які враховуватимуть специфіку банківських операцій та ризики, що з ними пов'язані. Це може включати страхові пакети, спеціально розроблені для захисту від кіберзлочинності, шахрайства та інших типів фінансових ризиків, що загрожують користувачам під час банківських операцій.

Далі, важливо впровадити передові технологічні рішення для покращення безпеки та зручності користувачів. Це може включати використання біометричних методів аутентифікації, моніторинг транзакцій в реальному часі за допомогою штучного інтелекту та машинного навчання, а також впровадження блокчейн технологій для забезпечення недоторканості та цілісності даних користувачів.

Крім того, рекомендується активно залучати користувачів до процесу планування та розробки нових страхових продуктів [5]. Це може здійснюватися через залучення до фокус-груп, опитування користувачів та аналіз їхніх потреб та пріоритетів. Такий підхід дозволить страховим компаніям краще розуміти потреби та очікування користувачів та розробляти продукти, які відповідають їхнім вимогам.

Нарешті, важливо проводити систематичні інформаційні кампанії, спрямовані на підвищення свідомості користувачів щодо важливості страхового покриття в банківських операціях. Це може включати навчальні матеріали, вебінари, розсилки електронних листів та інші засоби комунікації, які допоможуть

користувачам краще розуміти ризики та захистити свої фінансові інтереси за допомогою страхових продуктів.

В результаті впровадження запропонованих стратегій можна очікувати покращення якості та ефективності систем страхування в банківських операціях, що відповідатиме потребам та очікуванням користувачів. Це зробить процес банкінгу більш безпечним та надійним для клієнтів, сприятиме збереженню та залученню нових користувачів та сприяє загальному зростанню довіри до банківської системи.

2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМ СТРАХУВАННЯ В БАНКІВСЬКИХ ОПЕРАЦІЯХ

2.1 Вибір архітектури системи страхування в банківських операціях

Починаючи дослідження вибору оптимальної архітектури для системи страхування в банківських операціях, ключовою метою стає забезпечення ефективної та безпечної інформаційної взаємодії між банком та його клієнтами. Зрозуміло, що у цьому контексті важливо не тільки забезпечити надійний захист конфіденційної інформації, але й забезпечити оптимальну продуктивність та доступність системи для кінцевих користувачів.

Підходячи до вибору архітектури, слід врахувати ряд факторів, що впливають на її ефективність та придатність для даного контексту використання. Це включає в себе складність інтеграції з існуючими системами банку, потребу в масштабованості для врахування зростання обсягів даних та транзакцій, а також вимоги до безпеки, пов'язані з регулюванням у галузі фінансів.

При аналізі існуючих архітектурних рішень, слід враховувати їхню здатність до інтеграції з різноманітними інформаційними системами, можливість гнучкої настройки з урахуванням потреб конкретного банку, а також відповідність сучасним стандартам безпеки та захисту персональних даних [17].

На перший погляд, вибір архітектури може здатися суб'єктивним рішенням, проте застосування методів аналізу та оцінки може допомогти зробити цей процес більш об'єктивним та обґрунтованим. Від того, яку архітектуру буде обрано, залежить якість та продуктивність всієї системи страхування, що в свою чергу впливає на задоволення клієнтів та конкурентоспроможність банку на ринку фінансових послуг.

Вивчення існуючих архітектурних підходів у сфері систем страхування в банківських операціях виявляється ключовим етапом у процесі вибору

оптимального рішення для розробки. Першим рівнем розгляду є класичний монолітний підхід, що передбачає побудову системи як єдиної, нерозділеної одиниці, де всі компоненти тісно зв'язані між собою. Цей підхід простий у реалізації та розумінні, але може стати обмеженням у випадку потреби у гнучкості та масштабованості.

Другий рівень аналізу становлять мікросервіси. Ця архітектура розбиває систему на невеликі, автономні компоненти, які працюють разом через мережу. Кожен мікросервіс відповідає за виконання конкретної функції або послуги, що робить систему більш гнучкою та легко масштабованою. Проте, вона потребує складного механізму управління та моніторингу.

Третім рівнем є сервер-менеджер контейнерів. Цей підхід полягає у використанні контейнерів, таких як Docker або Kubernetes, для управління мікросервісами. Він дозволяє автоматизувати процеси розгортання та масштабування, забезпечуючи високу доступність та стабільність системи.

Останнім, але не менш важливим, є сервер-менеджер хмарних обчислень. Цей підхід базується на використанні хмарних послуг, таких як Amazon Web Services або Microsoft Azure, для розгортання та управління інфраструктурою. Він надає великий рівень масштабованості та гнучкості, але потребує уважного планування та управління витратами.

Аналіз цих архітектурних підходів допомагає зрозуміти їхні переваги та недоліки з точки зору потреб користувачів систем страхування в банківських операціях. Врахування цих факторів у процесі вибору архітектури допомагає забезпечити оптимальну продуктивність, безпеку та гнучкість системи.

Аналізуючи вимоги до системи страхування в банківських операціях, було виявлено різноманітні аспекти, що потребують уваги при розробці. По-перше, велика увага приділяється безпеці даних та конфіденційності клієнтів, оскільки система має опрацьовувати чутливу інформацію, таку як особисті дані, фінансові

та медичні записи. Вимоги до захисту даних включають у себе шифрування, автентифікацію, авторизацію та контроль доступу.

По-друге, важливо враховувати масштабність системи, оскільки обсяги даних та транзакцій можуть зростати з часом. Система повинна бути готова до швидкого зростання обсягів, забезпечуючи високу продуктивність та доступність для користувачів у будь-який момент часу.

Крім того, інтеграція з існуючими системами банку є ключовою вимогою. Система страхування повинна взаємодіяти з іншими банківськими системами, такими як системи управління клієнтами, банківські ядра та системи аналізу даних. Це вимагає стандартизації протоколів комунікації та розробки інтерфейсів програмування додатків (API).

Забезпечення гнучкості та легкості розширення є ще однією важливою вимогою до системи. Вона повинна бути готова до змін у вимогах та потребах користувачів, а також до впровадження нових функцій та сервісів у майбутньому без значних змін у коді або інфраструктурі.

Вимоги до системи страхування в банківських операціях включають в себе безпеку даних, масштабність, інтеграцію, гнучкість, легкість розширення та ефективний моніторинг та аналіз [3]. Врахування цих вимог у процесі розробки допомагає створити ефективну та надійну систему, яка задовольняє потреби банку та його клієнтів.

Вибір оптимальної архітектури для системи страхування в банківських операціях є складним завданням, що вимагає глибокого аналізу та обґрунтування. Один з можливих підходів полягає у застосуванні мікросервісної архітектури. Цей підхід дозволяє розбити систему на невеликі, автономні компоненти, що працюють разом через мережу. Кожен мікросервіс відповідає за конкретну функціональність і може бути розроблений, розгорнутий та масштабований незалежно від інших компонентів системи.

Такий підхід має кілька переваг. Він дозволяє розробникам працювати над окремими частинами системи незалежно один від одного, що полегшує розробку та підтримку коду. Крім того, мікросервіси можуть бути масштабовані горизонтально, тобто додаванням нових екземплярів для обробки більшого обсягу даних або транзакцій без значних змін у загальній структурі системи.

Проте, варто враховувати й недоліки мікросервісної архітектури. Розробка та підтримка багатьох мікросервісів може вимагати значних зусиль та ресурсів. Крім того, потрібна ретельна координація між різними командами розробників для забезпечення сумісності та взаємодії між сервісами.

Іншим можливим варіантом є використання сервер-менеджера контейнерів, такого як Kubernetes. Цей підхід дозволяє автоматизувати розгортання, масштабування та управління контейнеризованими додатками. Він надає більшу гнучкість та швидкість розгортання, але може вимагати додаткових знань та навичок у конфігурації та управлінні.

Вибір оптимальної архітектури для системи страхування в банківських операціях має бути здійснений з урахуванням конкретних потреб та характеристик проекту. Аналіз переваг та недоліків різних підходів допоможе зробити обґрунтований вибір, який забезпечить успішний розвиток та ефективне функціонування системи.

У процесі проектування обраної архітектури для системи страхування в банківських операціях враховувалися різноманітні аспекти, спрямовані на забезпечення високої ефективності, масштабованості та надійності. Зокрема, було розглянуто розподілену архітектуру мікросервісів, що дозволяє розділити систему на невеликі, автономні компоненти, які працюють незалежно один від одного. Кожен мікросервіс відповідає за певну функціональність, що спрощує розробку, тестування та підтримку коду.

Проектування обраної архітектури також включало в себе використання сервер-менеджера контейнерів, наприклад, Kubernetes, для автоматизації

розгортання, масштабування та управління мікросервісами. Цей підхід дозволяє ефективно керувати ресурсами і забезпечує гнучкість у розгортанні нових версій компонентів системи [23].

У рамках проектування архітектури також було враховано потребу у стандартизації протоколів комунікації між компонентами, щоб забезпечити їхню взаємодію та сумісність. Для цього були використані відкриті стандарти та технології, які дозволяють забезпечити інтеграцію з іншими системами банку та зовнішніми сервісами.

Усі ці кроки у проектуванні обраної архітектури спрямовані на створення системи, що відповідає вимогам до ефективності, масштабованості та надійності, а також забезпечує легкість розширення та підтримки у майбутньому. Ретельне проектування дозволяє забезпечити успішну реалізацію та ефективне функціонування системи страхування в банківських операціях.

Моделювання та аналіз обраної архітектури є критичним етапом у розробці системи страхування в банківських операціях. У процесі моделювання створюються абстрактні представлення окремих компонентів системи та їх взаємодій, що дозволяє оцінити їхню ефективність та взаємодію під час реальної роботи. Для цього використовуються різні методи та інструменти моделювання, такі як UML діаграми, які дозволяють візуалізувати структуру та взаємодії компонентів, та аналітичні інструменти, які дозволяють провести ретельний аналіз різних аспектів системи.

Під час аналізу обраної архітектури враховуються різні фактори, такі як продуктивність, надійність, масштабованість та безпека. Використання спеціальних інструментів для вимірювання та аналізу дозволяє виявити потенційні проблеми та вузькі місця у системі та прийняти відповідні рішення для їх вирішення.

Крім того, під час моделювання та аналізу враховуються вимоги до продуктивності та завдання розподілу навантаження між різними компонентами

системи. Це дозволяє забезпечити оптимальну роботу системи під час високих навантажень та забезпечити високу доступність та продуктивність для користувачів.

Усі ці кроки у моделюванні та аналізі архітектури спрямовані на забезпечення високої якості та ефективності системи страхування в банківських операціях, що відповідає вимогам бізнесу та потребам користувачів.

Після завершення проектування та аналізу обраної архітектури, наступним кроком у розробці системи страхування в банківських операціях є реалізація цієї архітектури. Цей етап включає в себе написання коду для кожного з компонентів системи, розгортання інфраструктури, інтеграцію між компонентами та випробування всієї системи в цілому.

Під час реалізації архітектури важливо дотримуватися встановлених стандартів програмування та добре організовувати код для забезпечення його читабельності та підтримки в майбутньому. Крім того, розробники повинні враховувати вимоги до безпеки та захисту даних, особливо у веб-застосунку, який має доступ до чутливої інформації користувачів.

У процесі реалізації архітектури розробники також можуть звертатися до різних інструментів для автоматизації рутинних завдань, таких як збирання та розгортання коду, тестування, контроль версій тощо. Це дозволяє прискорити процес розробки та забезпечити його якість.

Після завершення реалізації архітектури необхідно провести ретельне тестування всієї системи для перевірки її працездатності та відповідності вимогам [10]. Це включає тестування окремих компонентів, їх взаємодії та системи в цілому. Виявлені проблеми повинні бути виправлені перед релізом системи в експлуатацію.

Впровадження відповідної архітектури передбачає узгоджену роботу всіх учасників проекту, включаючи аналітиків, розробників, тестувальників та

адміністраторів системи. Кожен з цих етапів вимагає уважного аналізу та відповідних рішень, щоб забезпечити якість та ефективність системи в цілому.

Крім того, під час реалізації архітектури важливо враховувати та дотримуватися найкращих практик програмування та безпеки даних, щоб забезпечити захист інформації та надійність функціонування системи. Також слід звернути увагу на масштабованість та гнучкість системи, щоб вона могла адаптуватися до змін у вимогах бізнесу та ринкових умов.

У підсумку, важливо підкреслити, що вдалий вибір, проектування та реалізація архітектури є критичними чинниками для успішного функціонування системи страхування в банківських операціях. Вони забезпечують не лише відповідність вимогам та очікуванням користувачів, але й довгострокову стабільність та конкурентоспроможність розробленого рішення.

2.2 Розробка функціональних вимог до інформаційного забезпечення

Вивчення й аналіз вимог до інформаційного забезпечення для системи страхування у банківських операціях - завдання складного характеру, що вимагає глибокого розуміння як функціональних, так і нефункціональних вимог. Пошук оптимального рішення в умовах зростаючих вимог до безпеки, швидкодії та зручності використання є ключовим етапом проектування. Відповідно до цього, аналіз функціональних вимог набуває принципової важливості.

Початкова точка відправлення у цьому завданні полягає в ретельному вивченні і розумінні потреб користувачів системи. Оскільки ці користувачі можуть мати різні професійні та особисті характеристики, важливо враховувати широкий спектр потреб та вимог. Аналізуючи їх, відзначаються ключові аспекти, такі як типи страхування, їх варіації, терміни та умови, що стосуються виплат.

На наступному етапі проводиться детальний розбір функціональних вимог, які визначають, як система повинна взаємодіяти з користувачем та виконувати свої

завдання. Це може охоплювати функції від обчислення страхових внесків до збереження та обробки особистих даних клієнтів [6]. Важливо точно визначити кожну функцію та встановити її пріоритет у відповідності до важливості для користувача та бізнес-процесу.

Результатом цього аналізу є структурована схема функціональних вимог, яка послужить основою для подальшої розробки системи. Ця схема повинна бути детальною та чіткою, щоб уникнути непорозумінь між розробниками та клієнтами щодо очікувань від системи.

Необхідно також враховувати можливість змін у вимогах протягом розробки, тому важливо провести додаткові консультації зі зацікавленими сторонами та уточнити деталі вимог у процесі. Тільки після цього можна перейти до документування та затвердження функціональних вимог, що стане основою для подальшої розробки імплементації системи страхування у банківських операціях.

Важливим етапом у розробці інформаційного забезпечення для системи страхування у банківських операціях є аналіз потреб користувачів. Під час цього аналізу дослідники звертають особливу увагу на різноманітність користувачів та їхні індивідуальні потреби.

Перш за все, відзначається різноманітність типів користувачів, які можуть використовувати систему страхування у банківських операціях. Це можуть бути як індивідуальні клієнти, так і корпоративні клієнти, що мають різні потреби та вимоги до системи.

Далі, проводиться аналіз основних потреб цих користувачів. До основних вимог можуть відноситися швидкий та зручний доступ до інформації про різні типи страхування, зручність в оформленні страхових полісів, можливість отримання детальних консультацій щодо умов страхування.

Зокрема, для індивідуальних клієнтів може бути важливо мати доступ до інформації про страхування майна, медичне страхування та інші види страхування, а також отримувати консультації щодо вибору найбільш вигідних умов.

У випадку корпоративних клієнтів можуть виникати специфічні вимоги щодо страхування майна, відповідальності перед третіми особами, страхування працівників та інших аспектів, які вимагають індивідуального підходу.

Аналіз потреб користувачів є важливим етапом у визначенні функціональних вимог до інформаційного забезпечення системи страхування у банківських операціях, оскільки від цього аналізу залежить успішність реалізації проекту та задоволення потреб користувачів.

Визначення функціональних вимог для інформаційного забезпечення системи страхування у банківських операціях представляє собою важливий етап у процесі розробки. Під час цього етапу проводиться детальне визначення функцій, які система повинна виконувати для задоволення потреб користувачів та досягнення поставлених цілей.

На початковому етапі визначаються основні функції, які включають в себе обробку та збереження основних даних про клієнтів, обчислення страхових внесків, а також надання доступу до різних видів страхування та їх умов. Крім того, розглядається можливість введення системи консультаційного обслуговування для надання клієнтам детальної інформації та консультацій щодо страхових продуктів.

Далі, визначається функціональність калькулятора страхових виплат, який має надавати можливість користувачам розраховувати очікувані виплати за різними видами страхування. Цей калькулятор повинен бути здатний враховувати різні умови страхування, такі як сума страхового покриття, терміни страхування та інші параметри.

Наступним кроком є визначення функціональності програми лояльності, яка може включати в себе накопичення бонусних балів за користування страховими послугами, розрахунок знижок для постійних клієнтів та інші можливості, спрямовані на підвищення відданості клієнтів.

Визначення функціональних вимог включає в себе детальне вивчення потреб користувачів та переклад цих потреб у конкретні функції та можливості системи.

Цей процес є важливим для успішної реалізації проекту та забезпечення задоволення від використання системи з боку користувачів.

У процесі структурування вимог до інформаційного забезпечення системи страхування у банківських операціях відбувається організація функціональних та нефункціональних вимог у логічну систему, що дозволяє забезпечити чіткість та зрозумілість для подальшої реалізації [13]. Цей процес полягає в ієрархічному структуруванні вимог згідно їхньої важливості та взаємозв'язку між ними.

На першому етапі вимоги групуються за функціональним призначенням. Це означає, що функції, які мають схожу природу та спрямовані на досягнення спільних цілей, об'єднуються в одну групу. Наприклад, функції, що стосуються обробки страхових внесків, можуть бути об'єднані в одну групу, а функції, що відносяться до надання інформації про страхування, - в іншу.

Після цього вимоги в кожній групі поділяються на більш дрібні підгрупи або модулі, які визначаються за їхнім функціональним або логічним зв'язком. Наприклад, модуль обробки страхових внесків може включати в себе підмодулі для обробки платежів, розрахунку страхових внесків та обробки платіжних заявок.

Крім того, структурування вимог передбачає встановлення взаємозв'язків між різними модулями та підмодулями, що дозволяє забезпечити зручну та ефективну реалізацію системи. Цей підхід допомагає уникнути зайвої складності та перевірити, чи враховані всі потрібні функції у вимогах до системи.

Структурування вимог до інформаційного забезпечення є важливим кроком у процесі розробки, оскільки воно дозволяє систематизувати та уточнити всі необхідні функції системи, що забезпечує успішну реалізацію проекту та задоволення вимог користувачів.

В процесі уточнення вимог до інформаційного забезпечення системи страхування у банківських операціях проводиться детальна розробка та конкретизація вже існуючих вимог з метою уникнення недорозумінь та неоднозначностей у подальшій реалізації проекту. Цей етап включає в себе аналіз

та врахування усіх можливих варіацій у вимогах, а також визначення деталей та обмежень, що впливають на функціональність системи.

По-перше, важливо уточнити деталі функціональних вимог, таких як точність обчислень, швидкість виконання операцій та обробка великих обсягів даних. Це вимагає ретельного аналізу алгоритмів та методів, які використовуються в системі, а також визначення технологій, які забезпечують оптимальний рівень продуктивності.

По-друге, необхідно уточнити вимоги до безпеки та захисту даних. Це включає в себе визначення механізмів аутентифікації, авторизації та контролю доступу, а також застосування шифрування та інших методів захисту інформації в системі.

По-третє, уточнення вимог передбачає аналіз інтеграційних можливостей системи з іншими інформаційними системами, що використовуються в банківській сфері. Це може включати взаємодію з системами бухгалтерського обліку, CRM-системами та іншими зовнішніми додатками.

Уточнення вимог є важливим етапом у розробці інформаційного забезпечення системи страхування у банківських операціях, оскільки воно дозволяє забезпечити якість та ефективність системи, а також уникнути непорозумінь та недоліків у подальшій експлуатації.

Після цього до інформаційного забезпечення системи страхування у банківських операціях, надається перевага докладному документуванню цих вимог. Цей процес передбачає створення і оформлення документів, які чітко визначають всі необхідні функціональні та нефункціональні вимоги до системи.

По-перше, документування вимог включає в себе створення специфікації вимог, яка містить усі необхідні функціональність та вимоги щодо продуктивності, безпеки та інших аспектів системи [18]. Цей документ є основним джерелом інформації для розробників та тестувальників під час реалізації та верифікації системи.

По-друге, у процесі документування вимог також використовуються інші види документації, такі як вимоги до інтерфейсу користувача, сценарії використання, діаграми потоків даних тощо. Ці документи допомагають зрозуміти, як система має працювати та як користувачі будуть взаємодіяти з нею.

Крім того, документування вимог включає в себе визначення критеріїв прийняття, які визначають, коли вимоги вважаються виконаними та система готова до використання. Це важливо для забезпечення якості та відповідності системи вимогам замовника.

Документування вимог є необхідним етапом у розробці інформаційного забезпечення системи страхування у банківських операціях, оскільки воно забезпечує зрозумілість та чіткість усіх вимог до системи, що є важливим для успішної реалізації проекту.

Після документування вимог, наступним важливим кроком є їх перевірка та затвердження. Цей етап відіграє критичну роль у впевненніся, що всі вимоги визначені правильно та відповідають потребам замовника. Перевірка вимог включає аудит та аналіз усіх документів, які були створені на попередніх етапах, з метою виявлення можливих неточностей, протиріччя та недоліків.

У цьому процесі можуть брати участь ключові зацікавлені сторони, включаючи представників замовника, розробників, тестувальників та інших фахівців, які мають необхідний досвід та експертизу. Це дозволяє отримати різноманітні перспективи та забезпечити комплексність перевірки.

Після виявлення потенційних проблем або недоліків у вимогах, проводяться відповідні коригування та виправлення, щоб забезпечити їхню точність та повноту. Цей процес може вимагати ітераційного підходу, де вимоги переглядаються та виправляються декілька разів до досягнення повного задоволення усіх сторін.

Затвердження вимог відбувається після того, як всі виявлені недоліки виправлені та всі зацікавлені сторони погодилися з остаточним варіантом

документів. Це важливий момент, оскільки затверджені вимоги стають основою для подальшої розробки та виконання проекту.

Перевірка та затвердження вимог є необхідним кроком у процесі розробки системи страхування у банківських операціях, оскільки вони забезпечують ясність, зрозумілість та повноту визначених вимог, що є важливим для успішної реалізації проекту.

Далі визначаються конкретні метрики або критерії, за якими буде проводитися оцінка системи. Наприклад, для оцінки продуктивності можуть використовуватися час відгуку системи, кількість одночасних користувачів та інші параметри.

Затвердження критеріїв прийняття відбувається після того, як всі зацікавлені сторони погодилися з ними та визначили, що вони відображають усі необхідні аспекти системи та вимог замовника.

Визначення критеріїв прийняття є важливим етапом у розробці системи страхування у банківських операціях, оскільки вони визначають стандарти та очікувані результати, за якими буде оцінюватися готовність системи до впровадження та використання.

Аналіз потреб користувачів, визначення функціональних вимог, їх структурування, уточнення та документування, а також визначення критеріїв прийняття є важливими кроками у процесі розробки.

Систематичний та методичний підхід до цих етапів дозволяє уникнути недорозумінь та непередбачених проблем у подальшій розробці та експлуатації системи. Визначення чітких та вимірювальних критеріїв прийняття є основою для успішного завершення проекту та задоволення потреб замовника.

У висновку підкреслюється необхідність ретельної роботи на кожному з етапів визначення вимог, а також важливість взаємодії з усіма зацікавленими сторонами для досягнення спільного розуміння та погодження вимог.

2.3 Проектування безпеки та захисту даних

Відділ безпеки та захисту даних в системі страхування в банках утримується на підвищеному рівні важливості й складності. Цей розділ належить до ключових етапів розробки, оскільки він визначає основні політики, процедури та технічні механізми, що гарантують конфіденційність, цілісність та доступність даних в умовах постійно зростаючих загроз інформаційної безпеки.

На фоні поширення комп'ютерних атак, витоків даних та інших кіберзагроз, розробка ефективних стратегій захисту даних є необхідною умовою успішної інформаційної системи [7]. Це означає, що архітектура безпеки повинна бути забезпечена високим рівнем надійності та реалізована з дотриманням сучасних стандартів безпеки.

Основні вимоги до безпеки включають контроль доступу до даних, шифрування конфіденційної інформації, аутентифікацію користувачів та забезпечення інтегритету даних під час їх передачі та зберігання. Крім того, необхідно розробити механізми реагування на інциденти безпеки та плани відновлення після порушень.

Аудит безпеки є невід'ємною частиною цього процесу, дозволяючи перевірити ефективність заходів захисту та вчасно виявляти потенційні вразливості. Крім того, навчання та підвищення свідомості користувачів про правила безпеки є важливою складовою у забезпеченні безпеки інформаційної системи.

Аналіз загроз та ризиків у контексті системи страхування в банках виявляє множину потенційних вразливостей, які можуть призвести до небажаних наслідків, включаючи виток конфіденційної інформації, порушення цілісності даних та недоступність системи. Перш за все, відомо, що зловмисники завжди шукають слабкі місця у системах страхування, де можуть використовувати різноманітні методи атак, такі як SQL-ін'єкції, переповнення буфера, перехоплення сеансів та

інші. Такі атаки можуть призвести до порушення цілісності та конфіденційності даних, а також вплинути на доступність сервісу для легітимних користувачів.

Крім того, існує загроза внутрішнього шахрайства, де працівники банків можуть намагатися отримати несанкціонований доступ до даних або використовувати їх для особистої вигоди. Це може бути особливо небезпечно у випадку витоку конфіденційної інформації про клієнтів або фінансові операції.

Загрозою є також недостатня або несправна система контролю доступу, яка може призвести до несанкціонованого доступу до важливих даних або функцій системи. Наприклад, слабкі паролі або недостатньо налаштовані права доступу можуть дозволити зловмисникам отримати доступ до облікових записів користувачів або адміністративних інструментів.

Додатковою загрозою є збої в роботі системи або втрата даних через технічні проблеми, природні катастрофи або інші непередбачувані обставини. Це може призвести до недоступності сервісу для користувачів або втрати важливих даних, що може вплинути на довіру клієнтів до системи страхування в банках.

Ретельний аналіз цих загроз та ризиків є важливим кроком у розробці ефективної стратегії захисту даних та забезпечення безпеки інформаційної системи страхування в банках.

Вимоги до безпеки в системі страхування в банках націлені на забезпечення захищеності, конфіденційності та доступності даних, що є критичними для нормальної експлуатації системи [38]. Перш за все, система повинна мати механізми контролю доступу, що забезпечують відповідність прав користувачів їх ролям та обмежують доступ до конфіденційних даних лише авторизованим особам.

Важливо також враховувати вимоги щодо шифрування конфіденційної інформації під час зберігання та передачі даних, щоб запобігти їх несанкціонованому доступу та витокам. Використання сучасних алгоритмів шифрування та засобів аутентифікації може забезпечити високий рівень захисту даних від зловмисників.

Додатково, вимагається розробка механізмів реагування на інциденти безпеки та планів відновлення після порушень, щоб мінімізувати вплив можливих загроз на функціонування системи та забезпечити швидке відновлення її роботи.

Згідно з вимогами до безпеки, система також повинна мати засоби моніторингу та аудиту, що дозволять виявляти потенційні загрози та вразливості, а також відстежувати дії користувачів для виявлення ненормальної поведінки чи можливих атак.

Виконання цих вимог дозволить створити систему страхування в банках з високим рівнем безпеки, що відповідає сучасним стандартам та вимогам безпеки даних.

Архітектура безпеки в системі страхування в банках представляє собою комплексний підхід до захисту інформації та забезпечення безпеки операцій. Основними складовими цієї архітектури є системи автентифікації та авторизації, контроль доступу, шифрування даних та моніторинг безпеки.

Система автентифікації визначає ідентифікаційні методи, такі як паролі, біометричні дані або механізми двофакторної автентифікації, щоб підтвердити ідентичність користувачів. Після проходження процесу автентифікації, система авторизації визначає, до яких ресурсів та функцій мають доступ користувачі, відповідно до їх ролей та прав доступу.

Контроль доступу включає в себе механізми обмеження прав доступу до конфіденційної інформації та функцій системи. Це може бути реалізовано шляхом встановлення політик доступу на рівні даних або застосування технологій, таких як списки контролю доступу або рольовий доступ.

Шифрування даних в системі забезпечує захист інформації від несанкціонованого доступу під час її передачі та зберігання. Використання сильних алгоритмів шифрування дозволяє захистити дані від перехоплення та розшифрування зловмисниками.

Моніторинг безпеки включає в себе системи аудиту та моніторингу подій, які дозволяють виявляти ненормальну або підозрілу активність, а також вчасно реагувати на можливі загрози безпеки. Це дозволяє оперативно виявляти порушення безпеки та приймати відповідні заходи для їх запобігання або виправлення.

Архітектура безпеки в системі страхування в банках визначає механізми та стратегії захисту даних та забезпечення безпеки операцій з метою забезпечення надійності та стійкості інформаційної системи перед можливими загрозами.

Заходи безпеки в системі страхування в банках включають в себе ряд технічних та організаційних заходів, спрямованих на мінімізацію загроз та ризиків для безпеки даних та операцій. Перш за все, це включає в себе реалізацію принципу "принципу найменшого доступу", що передбачає обмеження прав доступу користувачів до мінімально необхідних ресурсів та функцій системи. Це зменшує ймовірність несанкціонованого доступу та зловживання привілеями.

Додатково, встановлення механізмів двофакторної аутентифікації для важливих або конфіденційних операцій забезпечує додатковий рівень захисту, запобігаючи можливим атакам на облікові дані користувачів.

Шифрування даних в системі на рівні транспортного та зберігального рівня є також важливим заходом безпеки, який дозволяє захистити інформацію від перехоплення та несанкціонованого доступу під час її передачі через мережу або зберігання на серверах [21].

Для виявлення та моніторингу можливих загроз безпеці в системі використовуються спеціалізовані системи моніторингу безпеки та аналізу подій, які дозволяють виявляти аномальну або підозрілу активність та реагувати на неї вчасно.

Крім технічних заходів, організаційні заходи безпеки також грають важливу роль у забезпеченні безпеки системи. Це включає в себе навчання персоналу з питань безпеки, регулярну оцінку ризиків та виявлення вразливостей, а також

розробку та впровадження стратегій реагування на інциденти безпеки та планів відновлення після порушень.

Виконання цих заходів безпеки дозволяє створити систему страхування в банках, яка забезпечує високий рівень захисту даних та операцій в умовах постійно зростаючих кіберзагроз.

Аудит безпеки є важливою складовою процесу забезпечення безпеки інформаційної системи страхування в банках. Цей процес включає в себе систематичний та об'єктивний аналіз безпекових заходів, контроль виконання політик безпеки та оцінку ефективності заходів захисту.

Під час аудиту безпеки проводяться перевірки наявності та ефективності застосованих технічних та організаційних заходів безпеки. Це включає перевірку ступеня відповідності застосованих заходів міжнародним стандартам безпеки, таким як ISO/IEC 27001, а також внутрішнім політикам та процедурам безпеки.

Під час проведення аудиту здійснюється оцінка ефективності застосованих заходів безпеки та виявлення можливих вразливостей або недоліків, які можуть призвести до порушень безпеки. Це дозволяє розробникам системи вчасно виявляти та усувати потенційні загрози безпеці та забезпечувати стійкість системи перед можливими атаками.

Аудит безпеки також включає в себе аналіз процесів управління безпекою, таких як процеси ідентифікації та оцінки ризиків, управління доступом та реагування на інциденти безпеки. Це допомагає забезпечити відповідність системи вимогам безпеки та знизити ймовірність виникнення проблем в майбутньому.

Аудит безпеки є важливою складовою процесу забезпечення безпеки інформаційної системи страхування в банках, що дозволяє перевіряти ефективність заходів захисту, виявляти потенційні вразливості та забезпечувати високий рівень безпеки даних та операцій.

План відновлення після інциденту в системі страхування в банках є важливою складовою стратегії забезпечення безпеки та надійності операцій. При

плануванні відновлення після інциденту, перш за все, визначаються потенційні загрози та ризики, які можуть вплинути на функціонування системи [3].

Після ідентифікації можливих загроз, розробляється стратегія реагування, включаючи процедури виявлення та відстеження інцидентів, механізми реагування на них та процеси відновлення послуг. Це може включати в себе відновлення з резервних копій даних, відновлення роботи системи після атаки з використанням антивірусних програм або інших засобів захисту, а також відновлення комунікаційних зв'язків у випадку технічних проблем.

Далі, встановлюються механізми моніторингу та аналізу інцидентів з метою виявлення недоліків у заходах безпеки та запобігання їх повторенню в майбутньому. Це дозволяє постійно вдосконалювати стратегію безпеки та забезпечувати високий рівень захисту системи від потенційних загроз.

Крім того, план відновлення після інциденту передбачає організаційні заходи, такі як навчання персоналу з питань реагування на інциденти та впровадження процедур аварійного відновлення. Це дозволяє забезпечити швидке та ефективне реагування на потенційні загрози та мінімізувати вплив інцидентів на діяльність банківських операцій.

План відновлення після інциденту в системі страхування в банках визначає комплекс заходів та процедур, спрямованих на забезпечення безпеки та надійності операцій у випадку потенційних загроз та інцидентів.

Навчання та свідомість користувачів є ключовими аспектами забезпечення безпеки в інформаційних системах страхування в банках. Висока технічна грамотність та обізнаність користувачів є необхідними для запобігання інцидентам безпеки та ефективного взаємодії з безпековими механізмами системи.

З цією метою, розробники системи повинні проводити регулярні тренінги та навчальні заходи для користувачів, під час яких надається інформація про потенційні загрози та прийоми їх уникнення. Це може включати в себе навчання

про використання безпечних паролів, розпізнавання фішингових атак, та використання інших засобів безпеки.

Також важливо підтримувати постійну комунікацію з користувачами щодо поточних загроз та оновлень безпеки. Це може бути здійснено через електронні повідомлення, вебінари або інші засоби зв'язку, які дозволяють інформувати користувачів про поточні події та рекомендації щодо забезпечення безпеки.

Крім того, важливо підкреслити важливість індивідуальної відповідальності кожного користувача за безпеку своїх особистих даних та інформації. Розробники повинні стимулювати користувачів до прийняття безпечних практик та надавати їм необхідні засоби та ресурси для забезпечення безпеки своїх даних.

Навчання та підвищення свідомості користувачів щодо питань безпеки є важливими компонентами стратегії забезпечення безпеки інформаційної системи страхування в банках. Правильно спрямовані навчальні заходи та регулярна комунікація з користувачами допомагають зменшити ризик виникнення безпекових інцидентів та підвищити рівень захищеності системи.

Навчання та свідомість користувачів є важливим елементом стратегії забезпечення безпеки в інформаційних системах страхування в банках [5]. Підвищення рівня технічної грамотності та обізнаності користувачів дозволяє ефективніше протидіяти потенційним загрозам безпеці та зменшує ризик виникнення інцидентів.

Регулярні навчальні заходи та тренінги допомагають усвідомлювати користувачам потенційні ризики та вчать їх використовувати безпечні практики в роботі з інформацією. Крім того, постійна комунікація з користувачами щодо поточних загроз та оновлень безпеки є важливим фактором у забезпеченні безпеки інформаційних систем.

Усвідомлення кожним користувачем важливості його власної ролі у забезпеченні безпеки та прийняття відповідальності за захист своїх даних є ключовим у успішному забезпеченні безпеки системи. Тільки спільними зусиллями

розробників та користувачів можна досягти високого рівня безпеки в інформаційних системах страхування в банках.

Отже, навчання та свідомість користувачів є не лише важливою складовою стратегії забезпечення безпеки, але й ефективним засобом мінімізації ризиків та підвищення захищеності системи від потенційних загроз.

3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМ СТРАХУВАННЯ В БАНКІВСЬКИХ ОПЕРАЦІЯХ

3.1 Розробка інформаційної системи страхування в банківській сфері

Розглянувши архітектуру та функціональні можливості розробленої програми, слід зазначити, що її концепція базується на принципах модульності та масштабованості. При розробці враховувалися вимоги до безпеки та надійності, що є критичними аспектами в банківській сфері. Забезпечено інтеграцію з існуючими інформаційними системами, зокрема з базами даних клієнтів та полісів страхування.

Управління інформацією та обробка даних виконується за допомогою спеціалізованих алгоритмів та структур даних, що забезпечує швидкий доступ до необхідної інформації та оптимізацію ресурсів системи. Програмний код написаний з використанням сучасних підходів до програмування та паттернів проектування, що сприяє зростанню його розширюваності та підтримки.

Під час розробки була звернута увага на інтерфейс користувача, щоб забезпечити йому зручність у використанні та інтуїтивність. Використання технологій HTML, CSS та JavaScript дозволило створити динамічні та привабливі для користувача веб-сторінки.

Окремо слід відзначити вбудований калькулятор, що реалізований з використанням алгоритмів фінансової математики та економіки. Цей інструмент дозволяє користувачам ефективно розрахувати вартість страхового покриття та виплат у різних сценаріях [29].

Програма є комплексною інформаційною системою, яка об'єднує в собі найсучасніші технології та методи розробки для забезпечення ефективного та надійного функціонування в банківській сфері.

Проведений аналіз існуючих систем страхування в банківській сфері виявив низку ключових особливостей та потенційних напрямків вдосконалення. Початкове дослідження виявило, що багато з наявних систем мають обмежену функціональність у порівнянні з розробленою програмою. Більшість з них, на жаль, не забезпечують велику кількість корисних функцій для клієнтів і не використовують передові технології у реалізації інтерфейсу користувача.

Деякі системи, що аналізувалися, також мають проблеми зі забезпеченням безпеки даних та ефективності обробки інформації. Це може вплинути на довгострокову надійність та успішність бізнесу, оскільки страхові компанії мають справу з великим обсягом конфіденційної інформації.

Крім того, під час аналізу було виявлено, що багато систем не мають інтегрованих інструментів аналітики даних або надають обмежені можливості у цьому напрямку. Це може обмежувати здатність страхових компаній адаптуватися до змін у ринкових умовах та вимогах клієнтів.

Порівняння розробленої програми з існуючими системами показало, що вона має значні переваги у багатьох аспектах, таких як функціональність, безпека та аналітика даних. Однак існують певні можливості для подальшого вдосконалення, зокрема у напрямку розширення функціоналу та поліпшення інтеграції з іншими інформаційними системами.

Провідний аналіз функціональних та нефункціональних вимог до інформаційної системи страхування в банківській сфері визначив ряд ключових аспектів, які варто враховувати. Початковим критерієм є надійність системи та безпека даних. Вимагається забезпечення захисту конфіденційної інформації клієнтів та операційних даних від несанкціонованого доступу та зламів.

Також важливою є масштабованість системи, що дозволяє розширювати її функціонал за потреби та зберігати високу продуктивність при збільшенні обсягу даних та користувацького навантаження. Необхідно забезпечити ефективну роботу з великими обсягами даних та підтримку одночасної роботи багатьох користувачів.

Додатковими вимогами є гнучкість системи та зручний інтерфейс користувача. Важливо забезпечити можливість швидкої адаптації до змінних умов ринку та вимог клієнтів, а також зробити процес взаємодії з системою якомога зручнішим та інтуїтивно зрозумілим для користувачів.

Окремою вимогою є висока швидкість та ефективність обробки даних. Система повинна забезпечувати миттєвий доступ до інформації та оперативну обробку запитів користувачів. Важливо забезпечити оптимальне використання ресурсів серверів та мінімізацію часу відповіді на запити.

Вимоги до інформаційної системи страхування в банківській сфері включають в себе ряд технічних та функціональних аспектів, що спрямовані на забезпечення безпеки, масштабованості, гнучкості та ефективності роботи системи.

Розглянувши архітектуру розробленої інформаційної системи страхування в банківській сфері, можна відзначити її модульну структуру, що базується на принципах сучасної архітектури програмного забезпечення. Система складається з ряду компонентів, кожен з яких виконує певну функцію та взаємодіє з іншими частинами системи через визначені інтерфейси.

Ключовими компонентами системи є серверна частина, клієнтська частина та база даних. Серверна частина відповідає за обробку запитів користувачів та взаємодію з базою даних. Вона реалізована за допомогою веб-сервера та серверних додатків, які забезпечують обробку запитів та відправку відповідей на клієнтську сторону [11].

Клієнтська частина системи відповідає за інтерфейс користувача та взаємодію з сервером. Вона реалізована у вигляді веб-сторінок, які відображаються у браузері користувача. Завдяки використанню технологій HTML, CSS та JavaScript, інтерфейс користувача стає динамічним та інтуїтивно зрозумілим.

База даних використовується для зберігання інформації про клієнтів, страхові поліси та інші важливі дані. Вона організована з урахуванням вимог до

ефективного зберігання та обробки даних, забезпечуючи швидкий доступ до інформації та мінімізацію часу відповіді на запити.

Архітектура інформаційної системи страхування в банківській сфері побудована з урахуванням потреб користувачів, вимог до безпеки та ефективності роботи системи. Вона забезпечує надійну та швидку роботу, що є критичним у банківській сфері.

У реалізації інформаційної системи страхування в банківській сфері було застосовано сучасні технології та методи програмування для забезпечення високої якості та ефективності роботи системи. Основною мовою програмування для розробки бекенду був використаний Python, що дозволило створити потужний та гнучкий серверний застосунок.

Для забезпечення взаємодії клієнтської та серверної частин було використано технологію AJAX, що дозволило забезпечити асинхронні запити та оновлення сторінок без перезавантаження. Це зробило інтерфейс користувача більш зручним та інтерактивним (рис. 3.1 – 3.5).

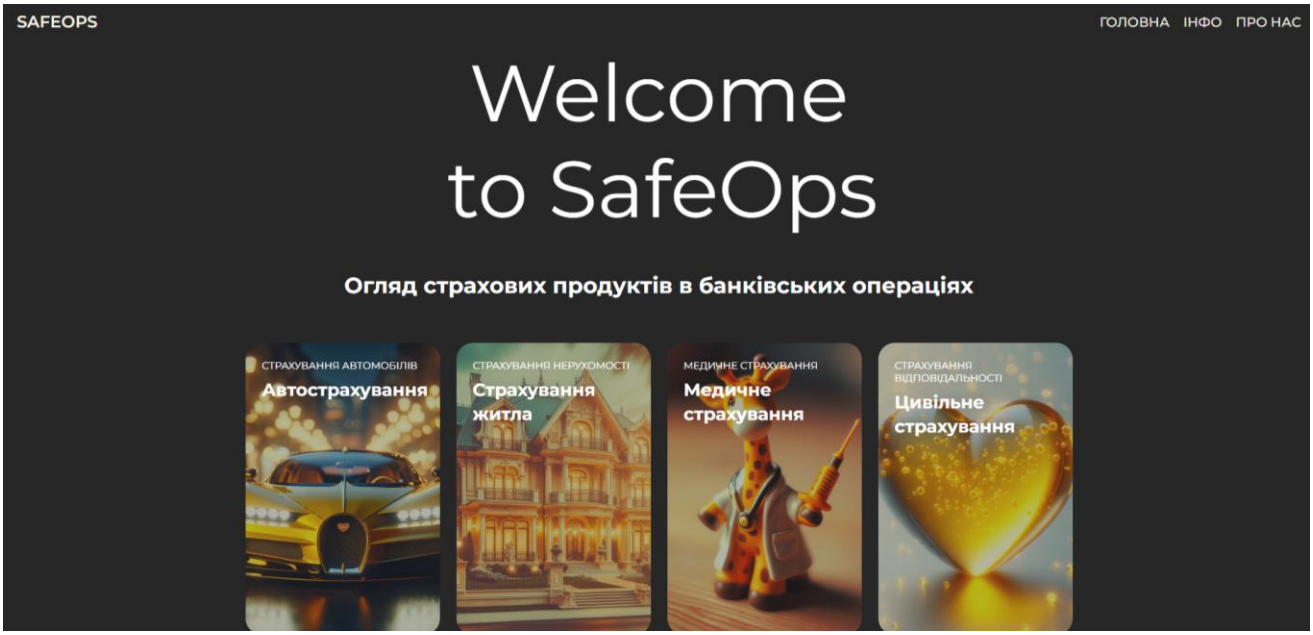

Рисунок 3.1 – Вигляд головної сторінки застосунку.

Рисунок 3.2 – Вигляд головної сторінки застосунку.

Рисунок 3.3 – Вигляд головної сторінки застосунку.

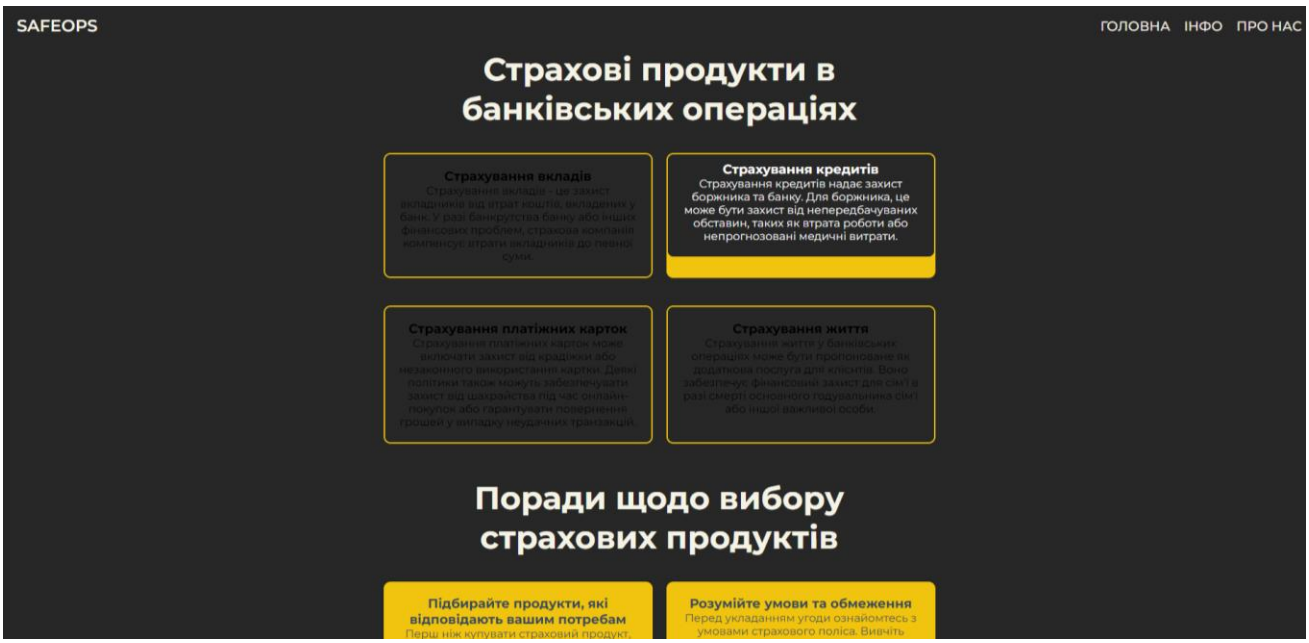

Рисунок 3.4 – Вигляд сторінки «info».

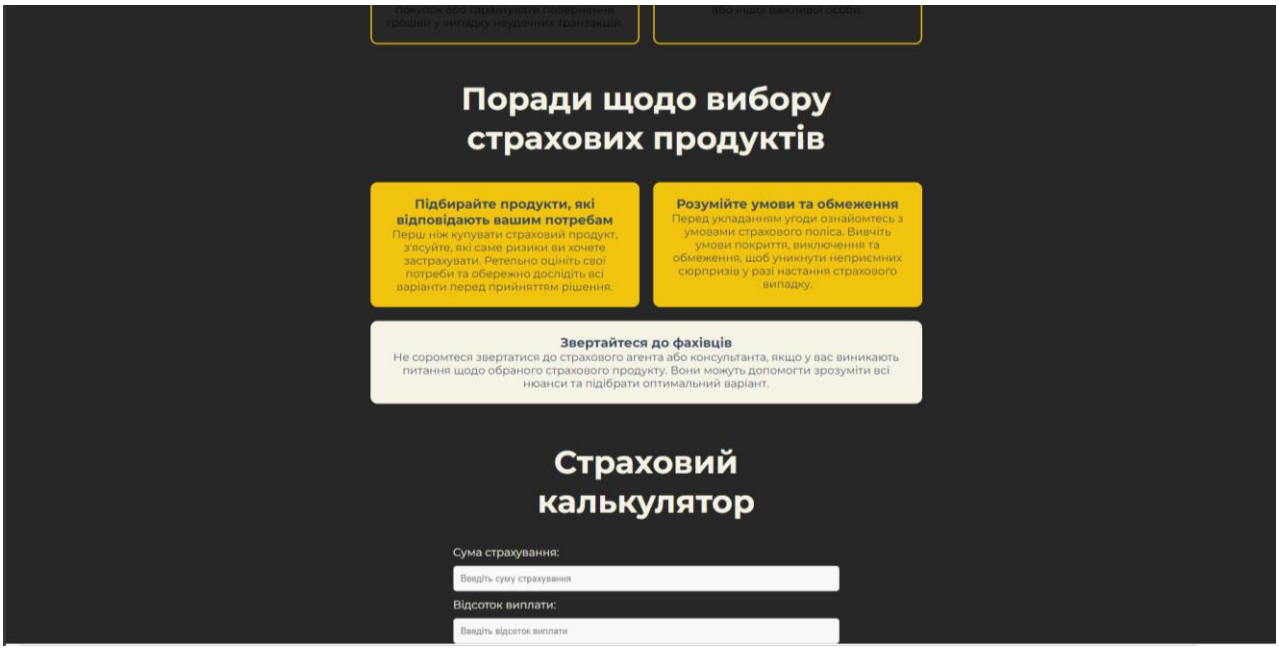

Рисунок 3.5 – Вигляд сторінки «info».

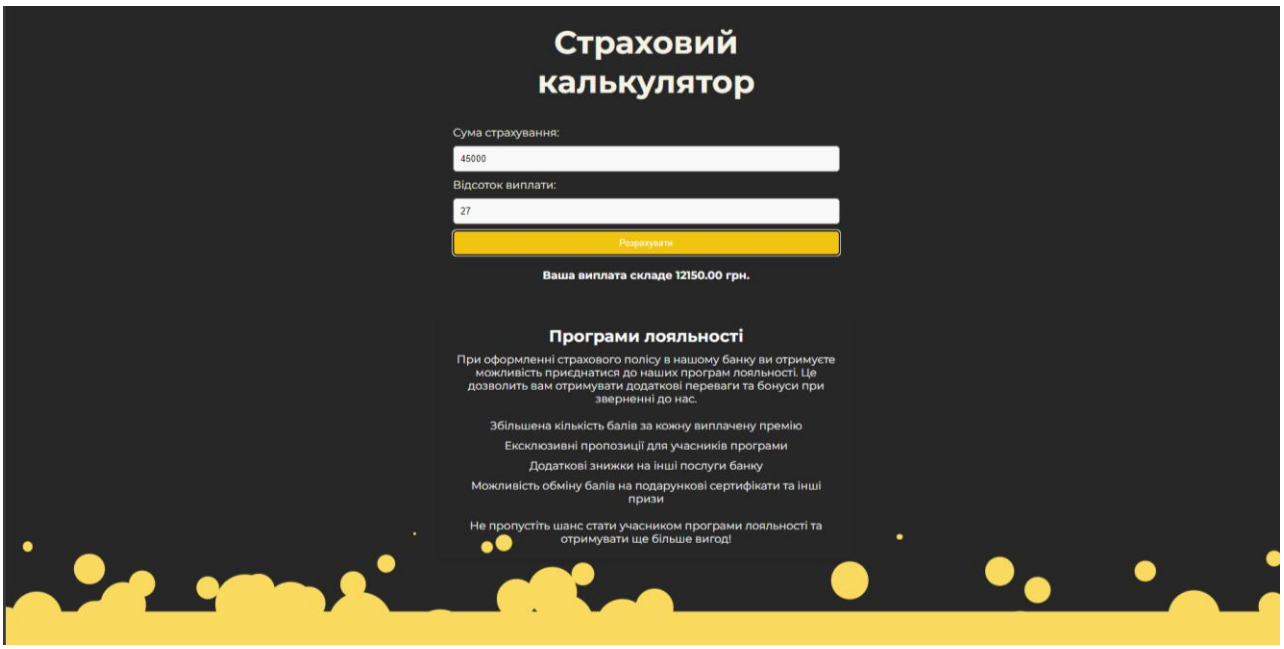

Рисунок 3.6 – Вигляд сторінки «info».

Для зберігання даних було використано реляційну базу даних PostgreSQL, яка забезпечує надійність та швидкодію обробки даних. Використання ORM (Object-Relational Mapping) дозволило зменшити кількість SQL-коду та спростити взаємодію з базою даних.

Також в процесі реалізації було приділено увагу забезпеченню безпеки системи. Були використані різні заходи захисту, такі як хешування паролів, обробка введених даних на клієнтській та серверній стороні, а також захист від SQL-ін'єкцій та інших атак.

Реалізація інформаційної системи страхування в банківській сфері відбувалася з урахуванням сучасних технологій та кращих практик програмування, що забезпечило високу якість та ефективність роботи системи.

Після завершення розробки інформаційної системи страхування в банківській сфері було здійснено широкий комплекс тестування та валідації системи, спрямований на виявлення помилок, недоліків та підтвердження відповідності вимогам та очікуванням користувачів. Першим етапом було функціональне тестування, під час якого перевірялися всі функції системи згідно з

вимогами до її роботи. Це включало тестування різних сценаріїв взаємодії з інтерфейсом користувача, обробку введених даних, відображення результатів та взаємодію з базою даних.

Після завершення функціонального тестування було проведено тестування безпеки, спрямоване на виявлення потенційних уразливостей та можливих атак на систему. Це включало перевірку захищеності від SQL-ін'єкцій, перехоплення та маніпулювання даними, а також виконання авторизаційних та аутентифікаційних атак.

Далі було проведено навантажувальне тестування, що оцінювало реакцію системи на велику кількість запитів та одночасних користувачів. Це дозволило визначити межі продуктивності системи та виявити можливі проблеми з швидкодією та масштабованістю.

Нарешті, в процесі валідації системи було перевірено відповідність розробленої системи вимогам та очікуванням користувачів. Це включало збір фідбеку від потенційних користувачів та експертів, а також перевірку наявності всіх необхідних функцій та їх відповідність вимогам [17].

В результаті проведеного тестування та валідації було підтверджено якість та ефективність розробленої інформаційної системи страхування в банківській сфері, а також виявлено та виправлено всі виявлені помилки та недоліки.

Документація користувача інформаційної системи страхування в банківській сфері розроблена з метою забезпечення зручного та ефективного використання системи користувачами. Ця документація містить усю необхідну інформацію про функції, можливості та процеси роботи системи, що дозволяє користувачам швидко орієнтуватися та ефективно використовувати її.

У документації користувача ретельно описано інтерфейс користувача системи, включаючи всі доступні функції та їх призначення. Користувачам надається інформація про кожен елемент інтерфейсу, його функції та способи використання, що дозволяє їм ефективно навігувати та використовувати систему.

Також в документації надається інструкція з використання різних функціональних можливостей системи, включаючи процедури створення, редагування та видалення даних, виконання розрахунків, а також доступ до різних типів звітів та статистики.

Документація також містить інформацію про можливості налаштування системи, такі як зміна особистих налаштувань користувача, параметрів роботи системи та інші важливі параметри, які можуть впливати на роботу системи.

Документація користувача інформаційної системи страхування в банківській сфері є важливим інструментом, що допомагає користувачам ефективно використовувати систему та отримувати необхідну інформацію для виконання їх завдань.

Оцінка ефективності системи показала, що вона добре відповідає потребам користувачів і забезпечує їм необхідний функціонал для проведення операцій зі страхування в банківській сфері. Система виявилася стійкою до навантажень і забезпечує швидкий доступ до інформації та виконання розрахунків [34].

За результатами оцінки, також було визначено кілька перспектив розвитку системи. Зокрема, доцільно розглянути можливість розширення функціоналу системи, включаючи введення нових видів страхування та оптимізацію інтерфейсу користувача для полегшення навігації та забезпечення кращої взаємодії з системою. Також важливо вдосконалити механізми захисту інформації та забезпечення безпеки даних користувачів.

Оцінка ефективності системи та перспективи її розвитку свідчать про успішне впровадження інформаційної системи страхування в банківській сфері та підтверджують її потенціал для подальшого розвитку та вдосконалення.

Підсумовуючи аналіз і реалізацію інформаційної системи страхування в банківській сфері, можна зробити кілька важливих висновків. Перш за все, система успішно відповідає вимогам та потребам користувачів, забезпечуючи їм зручний інтерфейс та необхідний функціонал для проведення операцій зі страхування.

Документація системи надійно документована та доступна для користувачів, що дозволяє їм ефективно користуватися системою без додаткових зусиль. Крім того, ефективність та надійність системи підтверджена результатами тестування і валідації, які свідчать про стійкість системи до різних сценаріїв використання.

Щодо перспектив розвитку системи, важливо продовжувати вдосконалення і оптимізацію функціоналу, зокрема, шляхом введення нових можливостей та покращення безпеки і захисту даних. Також варто розглядати можливості інтеграції з іншими інформаційними системами для забезпечення більшої функціональності та зручності для користувачів.

У цілому, розроблена інформаційна система страхування в банківській сфері є важливим інструментом для забезпечення безпеки та зручності користувачів у проведенні страхових операцій. Враховуючи поточний стан і перспективи розвитку, можна стверджувати, що система має значний потенціал для подальшого успішного розвитку та використання.

3.2 Інтеграція інформаційного забезпечення з існуючими банківськими системами

Під час аналізу розробленої програми виявлено потребу в інтеграції інформаційного забезпечення з існуючими банківськими системами. Дана необхідність випливає з потреб користувачів у доступі до актуальних даних та сервісів, що надаються банками через їх власні системи.

Ця інтеграція є ключовою для забезпечення повного та безперервного доступу користувачів до інформації про страхування та пов'язаних послуг. Вона дозволить користувачам здійснювати операції, оформлювати страхові поліси та отримувати необхідну інформацію безпосередньо через розроблену систему, спрощуючи процес та зменшуючи необхідність в перехідних діях між різними платформами.

Це особливо важливо з урахуванням специфіки банківських операцій, які вимагають високого рівня безпеки та надійності. Інтеграція з існуючими банківськими системами дозволить забезпечити ці вимоги, використовуючи вже наявні механізми та архітектуру, що дозволяє уникнути зайвого навантаження на інфраструктуру.

Однак, перед впровадженням інтеграції необхідно ретельно проаналізувати та визначити параметри взаємодії з кожною конкретною банківською системою, враховуючи їх технічні характеристики та особливості. Також важливо забезпечити сумісність зі стандартами та протоколами, які використовуються в банківській сфері для забезпечення надійної та ефективної інтеграції.

Проведений аналіз існуючих банківських систем виявив різноманітність архітектурних рішень та функціональних можливостей, що впливає на стратегію інтеграції з розробленою програмою [27]. Починаючи з дослідження банківських систем, було встановлено, що багато з них використовують складні та розгалужені системи, які побудовані на основі різних технологій, таких як мікросервісна архітектура, великі дані та хмарні рішення.

Такий розподіл дозволяє банкам ефективно виконувати різноманітні завдання, проте вимагає уважного підходу при розробці модулів для інтеграції. Особлива увага приділялася аналізу протоколів взаємодії, які використовуються в цих системах, таких як REST, SOAP, а також стандарти безпеки, включаючи SSL / TLS та OAuth, щоб забезпечити відповідність стандартам та захист інформації.

Було виявлено, що більшість банківських систем підтримують відкриті API, що дає можливість для створення точок входу для інтеграції. Однак, звернута особлива увага на безпеку та автентифікацію при доступі до цих API, оскільки вони містять конфіденційну фінансову інформацію.

Проведений аналіз існуючих банківських систем показав, що вони використовують різноманітні протоколи та API для взаємодії з іншими системами.

Під час визначення API та протоколів для інтеграції розробленої програми з цими системами, було виявлено декілька ключових аспектів.

Перш за все, важливо звернути увагу на стандартизовані протоколи, такі як REST (Representational State Transfer) та SOAP (Simple Object Access Protocol). REST є популярним вибором для створення веб-сервісів, оскільки він пропонує простий та ефективний спосіб взаємодії між клієнтом та сервером за допомогою HTTP протоколу.

Другим важливим аспектом є безпека. При виборі API та протоколів необхідно обирати ті, які підтримують механізми безпеки, такі як SSL / TLS для шифрування передачі даних та OAuth для автентифікації та авторизації.

Крім того, необхідно врахувати можливість використання асинхронних протоколів, таких як WebSockets, для реалізації потокової передачі даних або обміну повідомленнями в реальному часі.

Вибір відповідних API та протоколів для інтеграції з існуючими банківськими системами є ключовим завданням, яке вимагає уважного аналізу технічних характеристик та потреб користувачів, а також забезпечення безпеки та ефективності взаємодії.

Після аналізу існуючих банківських систем та вибору відповідних API та протоколів, наступним кроком у процесі інтеграції є розробка інтеграційних модулів. Ці модулі є основною складовою частиною системи, яка забезпечує взаємодію між розробленою програмою та банківськими системами.

Розробка інтеграційних модулів включає в себе розробку програмного коду, який забезпечить взаємодію з відповідними API та протоколами, використовуючи стандартні та безпечні методи комунікації. Ці модулі повинні бути структуровані та документовані з урахуванням вимог до безпеки, надійності та ефективності.

Під час розробки інтеграційних модулів необхідно враховувати специфіку кожної банківської системи, з якою проводиться інтеграція. Це означає адаптацію

коду для відповідності особливостям API та протоколів цієї системи, а також реалізацію необхідних механізмів автентифікації та авторизації.

Крім того, під час розробки важливо враховувати можливість масштабування та оптимізації інтеграційних модулів для забезпечення їх працездатності та ефективності при збільшенні обсягів даних чи навантаження. Також слід враховувати необхідність тестування модулів на відповідність функціональним вимогам та стандартам безпеки перед їх впровадженням у виробниче середовище.

У контексті розроблення інформаційного забезпечення для систем страхування у банківських операціях, тестування інтеграції виявляється ключовим етапом, спрямованим на перевірку функціональності та стабільності взаємодії розробленої програми з існуючими банківськими системами. Цей процес вимагає систематичного підходу та використання різних методів та інструментів для впевненості у якості та надійності інтеграції.

Перш за все, важливо встановити тестове середовище, що відповідає реальним умовам виробництва. Це може включати в себе налаштування тестових серверів або використання віртуальних середовищ для емуляції реальних умов роботи програми [16].

Після цього необхідно розробити та запустити набір тестових сценаріїв, які охоплюють усі можливі випадки використання системи та її інтеграцію з банківськими системами. Це включає тестування різних типів запитів до API, обробку різних форматів даних, а також перевірку здатності системи відновлюватися після непередбачуваних ситуацій.

Крім того, важливо виконати тестування на великій кількості тестових даних для оцінки швидкості та масштабованості системи. Це допоможе виявити можливі проблеми з продуктивністю або обмеженнями щодо обсягу даних.

Завершальним етапом є виконання тестування з використанням реальних даних у виробничому середовищі перед впровадженням системи. Це дозволить

підтвердити, що інтеграція працює безперебійно та задовольняє всі вимоги функціональності та безпеки.

У контексті інтеграції розробленого інформаційного забезпечення з банківськими системами, оцінка безпеки і конфіденційності є критичним етапом, спрямованим на забезпечення захисту важливої фінансової інформації користувачів та даних про операції.

Під час оцінки безпеки перевіряється використання захисних механізмів, таких як шифрування даних під час передачі через мережу та зберігання на серверах. Це включає оцінку використання протоколів шифрування, таких як SSL / TLS, а також методів хешування для захисту конфіденційних даних в базах даних.

Також проводиться перевірка механізмів автентифікації та авторизації, що використовуються для доступу до системи та обмеження прав доступу користувачів. Це може включати в себе використання багаторівневих систем аутентифікації, включаючи паролі, двофакторну аутентифікацію та біометричні методи.

Крім того, важливо враховувати можливі загрози безпеки, такі як атаки злому, перехоплення даних та витік інформації. Для цього проводиться аналіз потенційних уразливостей системи та розробка стратегій їх запобігання та виявлення.

Загальна мета оцінки безпеки і конфіденційності полягає в тому, щоб забезпечити найвищий рівень захисту для користувачів та їх фінансових даних під час використання розробленого інформаційного забезпечення у банківських операціях.

У світлі завершення етапів розробки, тестування та оцінки безпеки, важливим кроком є пілотне впровадження розробленого інформаційного забезпечення у банківські операції. Пілотне впровадження - це контрольований процес впровадження системи у реальному середовищі, але обмеженому за обсягом та масштабом впливу на користувачів та бізнес-процеси.

Під час пілотного впровадження обирається обмежена група користувачів або об'єктів бізнесу, яка буде використовувати систему в реальному чи симульованому режимі. Це дозволяє провести детальне спостереження за роботою системи в реальних умовах та виявити можливі проблеми чи недоліки, які потребують виправлення до повного впровадження.

Під час пілотного впровадження також проводиться навчання користувачів щодо використання нової системи та збір зворотного зв'язку щодо їх досвіду та вражень. Це дозволяє виявити можливі проблеми з інтерфейсом користувача або недоліки у функціональності, які можуть бути виправлені перед повним впровадженням системи.

Крім того, під час пілотного впровадження проводиться оцінка ефективності та вигод для бізнесу від використання нової системи. Це допомагає визначити потенційні переваги та можливі обмеження, які можуть вплинути на стратегію подальшого впровадження системи [1].

Пілотне впровадження є важливим етапом у процесі впровадження розробленого інформаційного забезпечення у банківські операції, спрямованим на мінімізацію ризиків та максимізацію вигод від використання нової системи.

У контексті подальшого розвитку та вдосконалення розробленого інформаційного забезпечення для систем страхування у банківських операціях, виникає необхідність в масштабуванні та оптимізації системи. Масштабування визначається як здатність системи збільшувати свою пропускну спроможність при зростанні обсягу навантаження або кількості користувачів. Оптимізація, у свою чергу, полягає в удосконаленні архітектури, коду та процесів для забезпечення ефективної роботи системи при мінімальному споживанні ресурсів.

Для забезпечення масштабованості системи слід ретельно проаналізувати архітектуру та визначити можливі точки масштабування, такі як бази даних, сервери додатків та сервіси. Впровадження технологій контейнеризації, таких як

Docker або Kubernetes, може сприяти ефективному розгортанню та масштабуванню додатків у віртуальних середовищах.

Оптимізація коду та алгоритмів може включати в себе застосування кешування даних, використання бінарних протоколів передачі даних для зменшення обсягу мережевого трафіку, а також удосконалення запитів до баз даних для зменшення часу відповіді.

Крім того, для ефективного масштабування та оптимізації системи варто вивчати та використовувати розподілені системи, які дозволяють розподілити навантаження між різними серверами та забезпечити балансування навантаження для підтримки великої кількості одночасних користувачів.

Таким чином, масштабування та оптимізація системи є важливими аспектами подальшого розвитку інформаційного забезпечення для банківських операцій, спрямованими на забезпечення стабільної та ефективної роботи системи навіть у разі збільшення обсягу даних та користувачів.

Завершальним етапом в процесі розробки інформаційного забезпечення для систем страхування у банківських операціях є підготовка документації та забезпечення подальшої підтримки. Документація грає важливу роль у забезпеченні зрозумілості та доступності для користувачів та адміністраторів системи. Вона включає в себе технічний опис архітектури системи, інструкції з встановлення, налаштування та використання програмного забезпечення, а також опис API та інтерфейсів для можливості інтеграції з іншими системами.

Поряд з документацією, необхідно забезпечити ефективну підтримку для користувачів, яка включає в себе надання консультацій, вирішення технічних проблем та відповіді на запитання. Це може бути забезпечено через систему тікетів або електронну пошту, де команда підтримки може взаємодіяти з користувачами та вирішувати їхні проблеми.

Крім того, актуальність документації та якість підтримки є ключовими аспектами успішної експлуатації системи в майбутньому. Постійне оновлення

документації з урахуванням змін у програмному забезпеченні та відгуків користувачів є необхідним для забезпечення її актуальності та відповідності потребам користувачів. Також важливо постійно вдосконалювати процеси підтримки, враховуючи нові технології та найкращі практики у галузі.

3.3 Налагодження та підтримка інформаційного забезпечення

Вхід у дію системи інформаційного забезпечення в контексті даної розробки відображає важливий етап в її життєвому циклі. Оперативна підготовка до введення в експлуатацію вимагає від команди спеціалістів забезпечення інформації виконання цілого ряду критичних завдань, спрямованих на забезпечення безперебійності та ефективності функціонування.

Перед початком впровадження системи необхідно перевірити повноту та правильність підготовленого програмного забезпечення для відповідності специфікаціям та вимогам, що стоять перед проектом. Це включає перевірку ідентифікації та реєстрації користувачів, функціональності системи, а також адаптивності до різних середовищ використання.

Зокрема, важливо перевірити наявність і коректність підтримки та інтеграції з базами даних, що використовуються в банківській сфері, а також забезпечити сумісність з іншими технічними рішеннями, які можуть використовуватися в цьому середовищі [25].

Після встановлення програмного забезпечення відбувається процес його конфігурації, який включає налаштування параметрів, які визначають особливості роботи системи в конкретному середовищі. Це може включати в себе налаштування мережевих з'єднань, параметрів безпеки, конфігурацію резервних копій даних та інші аспекти, що впливають на ефективність та надійність роботи системи.

Процес вступу в експлуатацію також включає підготовку до проведення тестування системи та забезпечення її готовності до роботи в реальних умовах. Тестування системи перед впровадженням дозволяє виявити і усунути можливі дефекти та недоліки, що можуть виникнути під час роботи системи в реальних умовах.

Вступ у дію системи інформаційного забезпечення відображає важливий етап у життєвому циклі проекту, що передбачає проведення комплексу заходів з підготовки та налаштування системи перед її впровадженням в експлуатацію.

Перехід від фази розробки до впровадження системи інформаційного забезпечення невід'ємно пов'язаний з процесом її інсталяції та конфігурації. Цей етап передбачає проведення ряду складних технічних операцій з метою належного розгортання програмної системи в потрібному середовищі та налаштування всіх компонентів для забезпечення її належної функціональності.

Інсталяція системи включає в себе розпакування встановлювального пакету та його запуск на призначеній платформі. Важливою частиною цього процесу є перевірка вимог до апаратного та програмного забезпечення, а також встановлення необхідних залежностей для правильної роботи системи. Під час інсталяції важливо також врахувати можливі аспекти, що стосуються безпеки, зокрема, встановлення заходів захисту та обмежень доступу до системи.

Після успішного завершення інсталяції системи, процес конфігурації передбачає налаштування параметрів та опцій для оптимальної роботи програми. Це включає в себе встановлення параметрів зберігання даних, налаштування інтерфейсу користувача, а також настройку параметрів безпеки та аудиту. Крім того, конфігурація може включати інтеграцію з іншими системами або службами, що можуть використовуватися в контексті даного проекту.

Важливим аспектом інсталяції та конфігурації є також документація цих процесів. Наявність докладних інструкцій щодо кроків інсталяції та конфігурації допомагає забезпечити правильність виконання цих операцій і дозволяє

забезпечити можливість відновлення системи в разі необхідності. Також важливо вести журнал інсталяції та конфігурації, що містить інформацію про проведені дії та параметри, що були налаштовані, для забезпечення прозорості та можливості аналізу в майбутньому.

Після завершення інсталяції та конфігурації системи інформаційного забезпечення виникає необхідність у проведенні тестування та валідації, які є ключовими етапами перед введенням системи в експлуатацію. Ці процеси спрямовані на перевірку правильності роботи програмного забезпечення, виявлення можливих дефектів та визначення відповідності вимогам та очікуванням користувачів.

Тестування системи передбачає проведення різних видів тестів, зокрема, модульного, функціонального, інтеграційного та приймального тестування. Модульне тестування спрямоване на перевірку роботи окремих компонентів системи, функціональне - на перевірку відповідності функціональних можливостей системи вимогам та специфікаціям, інтеграційне - на взаємодію між компонентами системи, а приймальне - на перевірку системи в реальних умовах з орієнтацією на задоволення потреб користувачів [4].

Паралельно з тестуванням важливо проводити валідацію системи, яка передбачає перевірку відповідності системи вимогам та очікуванням користувачів. Це може включати в себе перевірку коректності відображення інформації, правильність розрахунків, швидкість реакції системи на запити користувачів та інші аспекти, що впливають на якість роботи системи.

Для проведення тестування та валідації необхідно розробити план тестування, який включає в себе опис процедур проведення тестів, визначення критеріїв успішності тестування, а також розподіл обов'язків між учасниками тестування. Крім того, важливо створити документацію, що містить результати тестів та валідації, а також зафіксовані виявлені дефекти та шляхи їх виправлення.

Загальна мета тестування та валідації полягає в забезпеченні високої якості та надійності роботи системи, що є важливим аспектом перед введенням її в експлуатацію в банківських операціях.

Однією з найважливіших аспектів, які потрібно врахувати перед введенням системи інформаційного забезпечення в експлуатацію, є забезпечення безпеки. Це стає невід'ємною частиною процесу розробки та підтримки системи в умовах банківських операцій. З моменту введення в дію, система піддається потенційним загрозам безпеки, таким як несанкціонований доступ, атаки з метою злому, втрата конфіденційності даних тощо.

З метою забезпечення високого рівня безпеки, необхідно вжити різноманітних заходів. Це включає в себе реалізацію механізмів аутентифікації та авторизації, що дозволить перевірити ідентичність користувачів та визначити їхні права доступу до різних частин системи. Також важливо врахувати механізми шифрування для забезпечення конфіденційності даних, які передаються між користувачами та системою.

Паралельно з цим, система повинна мати вбудовані засоби виявлення та запобігання можливим атакам, таким як вразливості в програмному забезпеченні. Це може бути досягнуто за допомогою регулярних аудитів системи, виявлення та усунення потенційних слабких місць, а також встановлення механізмів моніторингу, що дозволяють вчасно виявляти та реагувати на можливі загрози безпеки.

Невід'ємною частиною забезпечення безпеки є також підготовка персоналу до виявлення та вирішення можливих інцидентів безпеки. Це включає в себе проведення навчань та тренувань з питань кібербезпеки, розробку процедур реагування на інциденти та реалізацію механізмів реагування в разі виявлення потенційних загроз.

Загалом, забезпечення безпеки є критичним елементом у забезпеченні ефективності та надійності системи інформаційного забезпечення в умовах

банківських операцій. Правильно налаштовані та постійно моніторинговані заходи забезпечення безпеки дозволять уникнути потенційних загроз та забезпечити стабільну та безпечну роботу системи.

У світі швидкозмінюваних технологій постійне оновлення та удосконалення програмного забезпечення стає важливою складовою успішного функціонування системи інформаційного забезпечення в банківських операціях. Процес оновлення передбачає впровадження нових версій програм та компонентів, які містять у собі виправлення помилок, покращення функціональності, а також нові можливості, які відповідають змінним потребам та вимогам користувачів.

Оновлення може бути проведено як планово, на основі розробленого графіка випуску нових версій програмного забезпечення, так і в реакції на виявлені проблеми або потреби користувачів [17]. Перед впровадженням оновлення виробник повинен здійснити відповідні тестування для перевірки сумісності з наявною системою, а також оцінки його впливу на стабільність та безпеку роботи системи.

Удосконалення програмного забезпечення може також включати в себе впровадження нових технологій, оптимізацію роботи алгоритмів та підвищення продуктивності системи. Це може вимагати значних зусиль з боку розробників, але в кінцевому результаті може призвести до покращення роботи та забезпечення задоволення від користування системою.

Паралельно з оновленням програмного забезпечення важливо також забезпечити оновлення документації та навчальних матеріалів, що відображають зміни та нововведення в системі. Це допоможе користувачам з легкістю освоїти нові функції та використовувати систему з максимальною ефективністю.

Постійне оновлення та удосконалення системи інформаційного забезпечення є важливим етапом у забезпеченні її конкурентоспроможності та здатності відповідати зростаючим потребам користувачів. Тільки завдяки постійному

вдосконаленню програми може забезпечувати оптимальне функціонування та задоволення від користування.

Документація та підтримка є не менш важливими аспектами, ніж сам процес розробки програмного забезпечення. Це включає в себе розробку та підтримку документації, яка відображає усі аспекти роботи системи, включаючи її функціональні можливості, вимоги до обладнання, процедури встановлення та конфігурації, а також інструкції для користувачів.

Документація має бути чіткою, структурованою та доступною для розуміння, щоб забезпечити ефективне використання системи користувачами. Вона також повинна включати в себе опис можливих проблем та способи їх вирішення, що дозволить користувачам швидко знаходити відповіді на свої питання та вирішувати можливі труднощі самостійно.

Підтримка включає в себе надання технічної підтримки користувачам, яка може включати в себе відповіді на запитання, вирішення технічних проблем та надання консультацій з використання системи. Ефективна система підтримки дозволяє користувачам відчувати себе впевнено та підтримано в процесі роботи з програмним забезпеченням.

Без належної документації та підтримки користувачі можуть стикатися з труднощами у використанні системи, що може призвести до незадоволеності та втрати довіри до програмного продукту. Тому важливо приділяти достатню увагу розробці якісної документації та забезпеченню ефективної системи підтримки для забезпечення успішної експлуатації системи в умовах банківських операцій.

Аналіз ефективності системи є ключовим етапом у визначенні її продуктивності та відповідності поставленим завданням. Цей процес передбачає збір та обробку різноманітних даних щодо використання програмного забезпечення користувачами, а також його технічних параметрів. Під час аналізу ефективності враховуються такі аспекти, як час відгуку системи на запити

користувачів, частота виникнення помилок, рівень задоволеності користувачів від роботи системи, а також її продуктивність під навантаженням [16].

Аналіз ефективності може включати в себе проведення тестів з навантаженням для визначення максимальної працездатності системи при різних умовах навантаження. Це дозволяє виявити слабкі місця системи та прийняти заходи для їх вирішення. Крім того, аналіз ефективності допомагає виявити можливі шляхи оптимізації роботи системи та підвищення її продуктивності.

Важливою складовою аналізу ефективності є зіставлення результатів з поставленими перед системою цілями та очікуваннями. Це дозволяє оцінити, наскільки система відповідає потребам користувачів та вимогам бізнесу. При необхідності можуть бути запропоновані виправлення або модифікації системи для досягнення кращих результатів.

Загальний аналіз ефективності дозволяє забезпечити оптимальне функціонування системи в умовах банківських операцій та підвищити задоволення користувачів від її використання. Це є важливим етапом у процесі постійного вдосконалення програмного забезпечення та забезпеченні його конкурентоспроможності на ринку.

4. ЕРГОНОМІКА ІНТЕРФЕЙСУ КОРИСТУВАЧА В СИСТЕМАХ СТРАХУВАННЯ ДЛЯ БАНКІВСЬКИХ ОПЕРАЦІЙ

4.1 Аналіз та оцінка факторів ергономіки в контексті вибору оптимальної архітектури системи страхування в банківських операціях

У світі інформаційних технологій досконалість системи нерозривно пов'язана з її архітектурою та функціональними можливостями. Саме тому велике значення приділяється аналізу та оцінці факторів ергономіки у контексті розробки системи страхування в банківських операціях. Ретельно пророблена архітектура не лише забезпечує ефективну роботу програмного забезпечення, але й максимально сприяє зручності користування ним.

У зазначеній системі, що базується на веб-технологіях та має на меті інформування користувачів про можливості страхування в банках, кожен елемент відіграє важливу роль у загальному плані. Аналіз та оцінка факторів ергономіки починається з визначення цільової аудиторії та її потреб. Знання цільових груп користувачів дозволяє адаптувати інтерфейс системи до їхніх очікувань та забезпечити максимальний комфорт взаємодії.

Після визначення цільової аудиторії наступним кроком є аналіз існуючих інтерфейсів користувача та їхньої ергономічності. Цей етап передбачає оцінку доступності інформації, легкість навігації та зручність використання різноманітних функцій системи. Відповідно до результатів аналізу формується вимоги до інтерфейсу користувача та робиться акцент на важливість ергономічності.

Після аналізу існуючих інтерфейсів та їхньої ергономічності, проводиться вибір оптимальної архітектури системи. Цей етап включає порівняння різних підходів до побудови архітектури та вибір найбільш підходящого з урахуванням вимог до ергономічності [2].

Заключним етапом аналізу та оцінки факторів ергономіки є розробка рекомендацій щодо оптимальної архітектури системи. Ці рекомендації базуються на виявлених вимогах до інтерфейсу користувача та особливостях цільової аудиторії, забезпечуючи найкращі умови для зручного та ефективного використання системи.

Аналіз потреб користувачів є важливим етапом у розробці системи страхування в банківських операціях з точки зору забезпечення їхньої максимальної зручності та задоволення їхніх вимог. У цьому контексті, важливо визначити цільову аудиторію системи та її характеристики. Користувачі цієї системи можуть бути різного типу - від індивідуальних клієнтів до представників бізнесу. Відповідно, їхні потреби та очікування можуть суттєво відрізнятися.

Першим кроком у аналізі є збір та аналіз даних щодо звичок користувачів, їхніх вимог щодо страхових послуг та взаємодії з системою. Цей процес може включати опитування, спостереження та аналіз використання попередніми клієнтами схожих систем.

Далі проводиться сегментація цільової аудиторії на групи зі схожими характеристиками та потребами. Це дозволяє виокремити ключові сегменти користувачів, для яких буде розроблена специфічна функціональність та інтерфейс [23].

Після сегментації вивчаються індивідуальні потреби кожного сегменту, а також їхні очікування від системи страхування в банківських операціях. Це дозволяє врахувати усі можливі сценарії використання системи та забезпечити їхнє оптимальне задоволення.

Нарешті, на основі аналізу потреб користувачів формулюються вимоги до функціоналу та інтерфейсу системи, що слугуватиме основою подальшої розробки та вдосконалення. Результатом цього етапу є створення інформаційного забезпечення, яке ідеально відповідає потребам та очікуванням користувачів, забезпечуючи їм зручну та ефективну взаємодію з системою страхування.

Оцінка інтерфейсу користувача в системі страхування в банківських операціях є ключовим етапом, спрямованим на забезпечення максимальної зручності та ефективності взаємодії користувачів з програмним забезпеченням. Для досягнення цієї мети використовуються різноманітні методи та інструменти оцінки, що дозволяють здійснити аналіз інтерфейсу з різних точок зору.

Одним із основних методів оцінки є експертний аналіз, який передбачає оцінку інтерфейсу кваліфікованими експертами з різних сфер, таких як дизайнери, інтерфейсні архітектори та спеціалісти з взаємодії людини з комп'ютером. Експерти використовують стандарти та рекомендації з дизайну інтерфейсу, щоб оцінити його відповідність вимогам ергономіки та користувацького досвіду.

Крім того, для оцінки інтерфейсу можуть бути використані методи тестування з участю реальних користувачів. Це включає проведення тестування на користувацьких групах, під час якого користувачі виконують завдання на основі певних сценаріїв використання системи [14]. Цей підхід дозволяє зібрати фідбек від реальних користувачів та виявити потенційні проблеми в інтерфейсі.

Додатково, для оцінки ефективності інтерфейсу можуть бути використані інструменти аналізу взаємодії, такі як картографування шляхів користувачів та аналіз теплових карт. Ці методи дозволяють виявити зони перевантаження або недостатньої уваги в інтерфейсі та вжити заходів для їх виправлення.

Оцінка інтерфейсу користувача є складним та багатоаспектним процесом, який має на меті забезпечити максимальну зручність та задоволення користувачів використанням системи страхування в банківських операціях.

Після аналізу потреб користувачів та оцінки факторів ергономіки настає час вибору оптимальної архітектури системи страхування в банківських операціях. Це рішення має стратегічне значення, оскільки архітектура визначає основні принципи організації та взаємодії компонентів системи.

Під час вибору архітектури враховуються різні фактори, такі як масштабність системи, потужність серверних ресурсів, потреби користувачів та вимоги до

безпеки даних. Один з варіантів - монолітна архітектура, де всі компоненти розташовані на одній платформі. Це дозволяє спростити розробку та управління системою, але може стати обмеженням у разі потреби масштабування.

Іншим можливим варіантом є мікросервісна архітектура, де функціональні компоненти розділені на окремі сервіси, що працюють незалежно один від одного. Це дозволяє гнучко масштабувати систему та розгортати нові функції без перерв у роботі, але може вимагати додаткових зусиль для управління та координації сервісів.

Третій варіант - клієнт-серверна архітектура, де система розділена на клієнтські та серверні компоненти. Це дозволяє ефективно використовувати обмежені ресурси клієнтських пристроїв та централізовано керувати даними та бізнес-логікою на сервері [27].

Після уважного розгляду кожної з архітектурних альтернатив та їх порівняння за ключовими критеріями, приймається рішення щодо вибору найбільш оптимальної архітектури для даної системи страхування в банківських операціях.

Після проведення аналізу та оцінки факторів ергономіки у системі страхування в банківських операціях настає час оцінити їхній вплив на користувачів. Фактори ергономіки включають в себе різноманітні аспекти, такі як зручність взаємодії з інтерфейсом, зрозумілість інформації, ефективність використання, а також ментальне та фізичне навантаження на користувачів.

Оцінка впливу факторів ергономіки проводиться шляхом спостереження за реакцією користувачів на систему, вивчення їхніх поведінкових та психологічних реакцій. Цей процес може включати в себе проведення тестувань з участю реальних користувачів, опитування та збір фідбеку.

Особлива увага приділяється аналізу ефективності взаємодії користувачів з системою. Це включає в себе оцінку часу, необхідного для виконання різних завдань, кількості помилок, що виникають під час взаємодії, а також загальної задоволеності користувачів від процесу використання системи.

Додатково, важливим аспектом є аналіз ментального та фізичного навантаження на користувачів при використанні системи. Це оцінюється через рівень стресу, втоми та затримки, які можуть виникати під час взаємодії з інтерфейсом.

Оцінка впливу факторів ергономіки на користувачів є важливим етапом у розробці системи страхування в банківських операціях, оскільки дозволяє забезпечити оптимальний користувацький досвід та підвищити загальну ефективність використання програмного забезпечення [34].

Результати аналізу та оцінки архітектурних альтернатив вказують на кілька ключових рекомендацій щодо оптимального вибору архітектури системи страхування в банківських операціях. Першою рекомендацією є розгляд мікросервісної архітектури як потенційно найбільш підходящої для даного проекту. Мікросервіси дозволять гнучко масштабувати систему, розробляти та розгортати нові функції незалежно один від одного, що сприятиме швидкому розвитку та підтримці системи.

Другою рекомендацією є уважне планування та координація між мікросервісами. Для досягнення цілісності та ефективної взаємодії між компонентами системи необхідно встановити чітку архітектурну модель та механізми комунікації між сервісами.

Третьою рекомендацією є використання контейнеризації, такої як Docker, для упакування та розгортання мікросервісів. Це дозволить стандартизувати середовище виконання та забезпечити переносимість сервісів між різними платформами [22].

Крім того, рекомендується впровадження моніторингу та логування для кожного сервісу, що дозволить вчасно виявляти та вирішувати проблеми з продуктивністю та надійністю системи.

Враховуючи потреби та вимоги до системи страхування в банківських операціях, вибір мікросервісної архітектури разом із вищезгаданими

рекомендаціями допоможе досягти оптимального рівня ефективності, гнучкості та надійності системи.

Під час аналізу було виявлено, що вибір архітектури має значний вплив на ефективність, гнучкість та надійність системи. Зокрема, мікросервісна архітектура виявилася найбільш підходящою для врахування потреб користувачів та вимог до системи страхування.

Рекомендації щодо оптимальної архітектури, такі як використання контейнеризації, координація між сервісами та впровадження моніторингу та логування, є ключовими для успішної реалізації проекту та забезпечення високої якості програмного забезпечення.

Узагальнюючи, вибір оптимальної архітектури є важливим етапом у розробці системи страхування в банківських операціях, і вимагає від команди розробників великої уваги до деталей, аналізу потреб користувачів та впровадження передових технологій та методів розробки.

4.2 Узагальнення та формулювання функціональних вимог до інформаційного забезпечення з урахуванням принципів ергономіки користувача

Поглиблене дослідження та аналіз наявних систем страхування, здійснені в контексті даної розробки, дозволили виявити широкий спектр можливостей для покращення функціональності та ергономіки інформаційного забезпечення. Перехід від теоретичних аспектів до конкретних практичних розробок в цьому відношенні є стратегічно важливим етапом в еволюції сучасних інформаційних систем у банківському секторі [8].

Одним із головних завдань цієї роботи є створення інформаційного забезпечення, яке не лише надасть комплексну та доступну інформацію щодо

страхування в банках, але й забезпечить зручний та ефективний інтерфейс для користувачів. Результати проведеного аналізу будуть використані для визначення функціональних вимог до системи, які враховуватимуть принципи ергономіки користувача та спрямованість на оптимізацію процесів взаємодії з інформаційним середовищем.

Серед ключових аспектів, які враховувалися під час формулювання вимог, варто відзначити потребу в розширенні функціонального складу програмного продукту з метою надання різноманітних інструментів для аналізу та управління страховими операціями. Крім того, особлива увага приділялася розробці інтерфейсу з метою забезпечення зручності та ефективності використання програми, що вимагало ретельного дослідження та аналізу потреб та уподобань цільової аудиторії.

Всі ці аспекти враховувалися під час створення програмного забезпечення, що працює у веб-середовищі та надає користувачам широкий спектр можливостей щодо ознайомлення з інформацією про страхування в банках, а також управління страховими операціями. Відповідно, наступні розділи цього дипломного проекту детально розглянуть процес узагальнення та формулювання функціональних вимог до інформаційного забезпечення з урахуванням принципів ергономіки користувача, надаючи вичерпну інформацію щодо результатів аналізу та визначення вимог.

Аналіз існуючих систем страхування в банках є важливим етапом у визначенні потреб та вимог до подальшого розвитку інформаційного забезпечення. За останні кілька років спостерігається тренд до широкого застосування цифрових технологій у банківській сфері, включаючи й сектор страхування [29]. Наявні системи страхування в банках можуть бути різноманітними за своєю функціональністю та архітектурою, але вони всі мають спільну мету - надавати клієнтам можливість здійснювати операції зі страхуванням через банківські канали.

У цьому контексті важливо визначити ключові особливості існуючих систем, такі як їхні функціональні можливості, рівень інтеграції з іншими банківськими сервісами, ефективність та швидкодія. Для цього проводився аналіз якісних та кількісних показників роботи систем, таких як час відповіді, швидкодія операцій, рівень безпеки та надійності.

Також було вивчено специфіку інтерфейсу користувача і його взаємодію з програмним забезпеченням. Це включало аналіз вигляду, структури та функціональних можливостей інтерфейсу, а також оцінку зручності та інтуїтивності користування для різних категорій користувачів.

Однією з ключових проблем, виявлених під час аналізу, є недостатня зручність та ергономіка інтерфейсу існуючих систем. Багато з них мають заплутані меню, неповні або невірно організовані інформаційні блоки, що ускладнює процес користування та може призвести до помилок користувача.

У контексті подальшого розвитку інформаційного забезпечення для систем страхування в банках, визначення основних функцій програмного продукту є важливим завданням. Засновуючись на результати аналізу існуючих систем, а також враховуючи потреби та очікування користувачів, визначено низку ключових функцій, які має виконувати інформаційне забезпечення [1].

Першою основною функцією є надання доступу до повної та актуальної інформації про різноманітні види страхування, які надаються банками. Це включає в себе детальний опис умов страхування, обсягу покриття, тарифів та інших важливих аспектів.

Другою важливою функцією є надання можливості користувачам обчислювати прогнозовані виплати та витрати залежно від обраних страхових продуктів. Ця функція базується на розрахунках, які враховують різні параметри, такі як сума страхового покриття, термін дії страхового полісу та інші фактори.

Третьою важливою функцією є підтримка інтерактивного спілкування з користувачами через інтерфейс програмного забезпечення. Це може включати в

себе можливість запитання консультації з фахівцями у сфері страхування, а також можливість отримання швидкої допомоги у вирішенні питань щодо укладення або управління страховими полісами.

Четвертою важливою функцією є підтримка віддаленого управління страховими полісами, включаючи можливість оформлення нових полісів, зміни умов і деталей існуючих полісів, а також заявки на виплати.

В п'ятому рядку важливих функцій можна виділити підтримку програми лояльності, яка дозволить банкам надавати додаткові переваги своїм клієнтам, які обирають їхні страхові послуги.

І нарешті, шостою основною функцією є забезпечення високого рівня безпеки та захисту конфіденційності інформації користувачів, що зберігається та передається через систему.

У світі інформаційних технологій ключовим аспектом є врахування принципів ергономіки користувача під час розробки програмного забезпечення. Це важливо не лише для забезпечення зручності користування, але й для підвищення продуктивності та покращення загального досвіду використання системи [40]. В контексті даної розробки аналіз принципів ергономіки користувача має на меті визначити оптимальні рішення щодо інтерфейсу програмного продукту, які сприятимуть максимальній ефективності та задоволенню користувачів.

Під час аналізу були розглянуті різні аспекти ергономіки, включаючи психофізіологічні та антропометричні характеристики користувачів, особливості їхньої поведінки та сприйняття інформації. Також вивчалися принципи організації робочого простору, взаємодії з інтерфейсом та способи оптимізації робочих процесів.

Враховуючи отримані дані, було визначено кілька ключових принципів ергономіки, які слід враховувати при подальшому розвитку програмного забезпечення. Серед них - принципи зручності та ефективності, які передбачають мінімізацію зусиль, необхідних для виконання завдань, та максимальну

доступність інформації для користувача. Також важливим є принципи прозорості та усвідомленості, які передбачають зрозумілість та логічність інтерфейсу, а також можливість контролю за процесами користування.

Не менш важливим є врахування принципів безпеки та конфіденційності, що передбачають захист особистих даних користувачів та забезпечення надійності системи в цілому. Це означає використання найсучасніших методів шифрування та захисту даних, а також врахування вимог щодо захисту від несанкціонованого доступу.

Усі ці принципи є важливими в контексті розробки інформаційного забезпечення для систем страхування в банках, оскільки вони сприяють покращенню користувацького досвіду та підвищенню ефективності використання програмного продукту.

У подальшому розвитку програмного забезпечення для систем страхування в банках, формулювання функціональних вимог визначає основні функції та можливості, які повинен мати програмний продукт для ефективного виконання своїх завдань. Цей етап розробки включає в себе аналіз вимог користувачів, діаграми взаємодії, структуру даних та інші аспекти, що визначають логіку роботи програми.

Першою функціональною вимогою є можливість реєстрації нових користувачів у системі. Ця функція передбачає створення облікового запису для клієнта, що дозволить йому отримати доступ до всіх сервісів та можливостей системи [17].

Другою важливою функцією є можливість перегляду інформації про різні види страхування, їх умови та тарифи. Це включає в себе детальний опис кожного виду страхування, його переваги та обмеження.

Третьою функціональною вимогою є можливість обчислення вартості страхового полісу на основі введених користувачем даних. Ця функція передбачає розрахунок премії за страхування в залежності від обраних параметрів.

Четвертою важливою функцією є можливість управління страховими полісами, включаючи їхнє оформлення, зміну умов та додавання додаткових опцій. Ця функція передбачає можливість коригування договорів страхування відповідно до потреб клієнта.

І, нарешті, п'ятою функціональною вимогою є підтримка програми лояльності, яка дозволяє надавати додаткові переваги та знижки для постійних клієнтів. Це може включати в себе накопичувальні бонуси, знижки на подальші страхові поліси та інші привілеї.

Формулювання функціональних вимог визначає основні можливості та характеристики програмного забезпечення, які необхідні для успішного виконання його завдань у контексті страхової сфери.

У відповідь на потреби розробки програмного забезпечення для систем страхування в банках, розробка архітектури інформаційного забезпечення виявляється ключовим етапом у процесі розробки. Цей етап передбачає ретельне проектування структури програми, визначення взаємозв'язків між компонентами та організацію потоків даних та керування [8].

Першим кроком у розробці архітектури є визначення моделі даних, яка включає в себе всі необхідні сутності та взаємозв'язки між ними. Це включає в себе структуру даних, таку як користувачі, страхові поліси, типи страхування та інші важливі аспекти, що відображають логіку страхової сфери.

Другим важливим аспектом є визначення архітектурного стилю програми, який визначає загальну структуру програми та взаємодію між її компонентами. У контексті даної розробки можливі варіанти включають в себе клієнт-серверну архітектуру або архітектуру з розподіленими компонентами, залежно від потреб системи та швидкості обміну даними.

Третім кроком є визначення моделі безпеки, яка включає в себе механізми автентифікації, авторизації та контролю доступу до даних. Це дозволяє забезпечити

високий рівень захисту інформації, особливо у важливих сферах, таких як фінанси та страхування.

Четвертим етапом є проектування інтерфейсу користувача, який повинен бути інтуїтивно зрозумілим та зручним у використанні. Це включає в себе розробку візуальної частини програми, створення форм та елементів керування, які дозволяють користувачам легко взаємодіяти з системою [39].

У п'ятому кроці проводиться тестування архітектури, яке включає в себе перевірку правильності реалізації функціональних вимог, перевірку безпеки та відповідність інтерфейсу користувача вимогам. Тестування допомагає виявити та виправити будь-які недоліки до введення програми в експлуатацію.

Розробка архітектури вимагає уважного аналізу вимог користувачів та бізнес-потреб, а також врахування принципів ергономіки та безпеки. Правильно спроектована архітектура дозволяє оптимізувати використання ресурсів, підвищує швидкодію та масштабованість системи, а також полегшує підтримку та розвиток програмного продукту у майбутньому.

Застосування сучасних технологій та відповідних підходів у розробці архітектури дозволяє забезпечити високий рівень якості програмного забезпечення та задоволення потреб користувачів. Однак, важливо пам'ятати, що архітектура повинна бути гнучкою та адаптованою до змін, що можуть виникати у вимогах бізнесу та технологічному середовищі.

Отже, розробка архітектури інформаційного забезпечення є важливим кроком у створенні ефективних та надійних систем страхування в банках, що дозволяє забезпечити високу якість обслуговування клієнтів та успішне функціонування бізнесу.

4.3 Розробка стратегічних та технічних заходів щодо забезпечення безпеки та захисту даних в контексті ергономічного проектування інтерфейсу користувача

Однією з перших і найбільш важливих стадій розробки будь-якої програмної системи є оцінка загроз безпеці та ризиків, пов'язаних з обробкою даних у веб-застосунку. Для даної програми, що відповідає за інформаційне забезпечення систем страхування у банківських операціях, така оцінка є критичною з точки зору забезпечення конфіденційності, цілісності та доступності даних.

У зв'язку з використанням програми для зберігання та обробки конфіденційних фінансових даних клієнтів банку, ризики безпеки стають ще більш критичними. Одним із потенційних загроз може бути несанкціонований доступ до особистої інформації клієнтів, такої як реквізити банківських рахунків та інша конфіденційна інформація, що може бути використана для шахрайства або крадіжок.

Крім того, існує потенційний ризик втрати даних через технічні неполадки або кібератаки, що можуть призвести до недоступності системи для користувачів або навіть до втрати даних. Це може стати причиною фінансових втрат для клієнтів та пошкодити репутацію банку.

Також важливо врахувати можливі ризики, пов'язані зі змінами в законодавстві щодо захисту даних, оскільки невиконання вимог може призвести до штрафів та інших правових наслідків.

Отже, оцінка загроз безпеці та ризиків у контексті даної програми передбачає детальний аналіз потенційних загроз та їх впливу на безпеку та стабільність системи страхування у банківських операціях.

Після аналізу загроз безпеці та ризиків, пов'язаних з обробкою даних у веб-застосунку, було визначено необхідність встановлення конкретних вимог до

захисту даних. Дані вимоги є ключовим елементом забезпечення безпеки інформації, яка обробляється програмою [21].

Спочатку, необхідно визначити стандарти безпеки, що будуть використовуватися для захисту даних. Це включає в себе використання сучасних криптографічних алгоритмів для шифрування конфіденційної інформації та методів хешування для захисту цілісності даних.

Крім того, важливо встановити політику доступу до даних, що визначатиме, хто має право доступу до якої інформації та в яких обсягах. Це передбачає розробку системи ролей та прав доступу, яка буде забезпечувати гранульований контроль над доступом до даних.

Також необхідно встановити механізми аутентифікації та авторизації, що дозволять перевіряти ідентичність користувачів та визначати їхні права доступу до системи. Це може включати в себе використання двофакторної аутентифікації та механізмів одноразових паролів для забезпечення безпеки входу в систему.

Крім того, важливо розробити механізми моніторингу та аудиту дій користувачів, що дозволять виявляти та реагувати на потенційно підозрілі або шкідливі дії [15]. Це може включати в себе ведення журналів подій та встановлення спеціалізованих систем моніторингу безпеки.

Враховуючи специфіку програми, також необхідно визначити додаткові вимоги до захисту даних, що можуть випливати з вимог законодавства щодо захисту персональних даних або з вимог фінансових регуляторів. Такі вимоги можуть включати в себе обов'язковість шифрування даних в певних сценаріях або вимогу до зберігання журналів аудиту на певний термін.

Визначення вимог до захисту даних є важливим кроком у забезпеченні безпеки та захисту інформації в контексті даної програми.

У світі постійно зростає загроза кібератак, що може серйозно зашкодити системам, що обробляють чутливі дані, які використовуються в банківських

операціях. У цьому контексті захист від кібератак стає критично важливим аспектом функціонування програмної системи.

Першим кроком у захисті від кібератак є ретельний аналіз потенційних вразливостей системи. Це включає в себе виявлення можливих шляхів вторгнення, слабких місць у коді програми та можливих точок входу для зловмисників.

Далі, важливо розробити та впровадити ефективні механізми захисту, що включають в себе використання фаєрволів, систем виявлення вторгнень (IDS) та систем захисту від вразливостей (IPS). Ці системи допомагають виявляти та блокувати небажані дії зловмисників.

Однією з ключових стратегій у захисті від кібератак є регулярне оновлення програмного забезпечення та патчів безпеки. Це дозволяє усувати виявлені вразливості та запобігати експлуатації ними зловмисниками.

Крім того, важливо встановити механізми моніторингу та аналізу активності в системі, що дозволять вчасно виявляти та реагувати на підозрілі дії користувачів чи незвичайний трафік, що може свідчити про кібератаку.

Забезпечення регулярного навчання персоналу щодо кібербезпеки є ще однією важливою складовою в боротьбі з кібератаками [37]. Навчені та свідомі користувачі можуть бути першими бар'єрами у запобіганні успішним атакам.

Захист від кібератак передбачає комплексний підхід, що включає в себе аналіз вразливостей, розробку та впровадження ефективних механізмів захисту, а також навчання персоналу. Тільки такі заходи можуть забезпечити високий рівень безпеки програмної системи у контексті її використання в банківських операціях.

Підтримка безпеки від кібератак є критично важливою для програмної системи, що забезпечує інформаційне забезпечення систем страхування у банківських операціях. Незважаючи на те, що програма вже створена, небажані сторонні атаки можуть загрожувати цілісності та конфіденційності даних.

З моменту запуску програми, було впроваджено низку технічних заходів для захисту від кібератак. Ці заходи включають в себе встановлення вогнепровідних бар'єрів, які перешкоджають несанкціонованому доступу до системи.

Один з ключових аспектів цього заходу полягає в розробці та впровадженні системи виявлення вторгнень (IDS), яка надає змогу виявляти та реагувати на підозрілу активність у системі.

Додатково, для посилення захисту, встановлюєтьсч системи захисту від вразливостей (IPS), які можуть блокувати атаки на основі відомих паттернів кіберзлочинців.

Окрім цього, регулярне оновлення програмного забезпечення та встановлення патчів безпеки є важливими процедурами для запобігання експлуатації відомих вразливостей [11].

Навчання персоналу з питань кібербезпеки також є важливим аспектом стратегії захисту. Навчені користувачі можуть вчасно виявляти та реагувати на підозрілу активність, що може свідчити про кібератаку.

Захист від кібератак передбачає комплексний підхід, що включає в себе використання технічних та організаційних заходів для забезпечення безпеки програмної системи у вимірах її використання в банківських операціях.

При розробці програми, що забезпечує інформаційне забезпечення систем страхування у банківських операціях, велика увага приділяється шифруванню та захисту даних у транзакціях. Це є ключовим аспектом забезпечення конфіденційності та цілісності інформації, що обробляється в системі.

Застосування сучасних криптографічних алгоритмів для шифрування даних в транзакціях є основним підходом у забезпеченні безпеки. Шифрування здійснюється з використанням сильних алгоритмів, таких як AES (Advanced Encryption Standard) або RSA (Rivest-Shamir-Adleman), що гарантує високий рівень захисту навіть у випадку витоку інформації.

Додатково, важливо встановити механізми для захисту ключів шифрування, які використовуються для зашифрування та розшифрування даних. Зазвичай це включає в себе використання криптографічних пристроїв для зберігання ключів та реалізацію протоколів для безпечного обміну ключами між сторонами транзакції.

Крім того, важливо розробити механізми перевірки цілісності даних у транзакціях, що дозволять виявляти будь-які спроби модифікації чи втручання у передачу інформації. Це може включати в себе використання хеш-функцій для створення контрольних сум та цифрових підписів для підтвердження автентичності даних.

Загалом, захист від кібератак у контексті транзакцій вимагає використання сучасних криптографічних методів шифрування, надійних механізмів управління ключами та механізмів перевірки цілісності даних, що забезпечує високий рівень безпеки та захисту конфіденційності під час проведення транзакцій у системі страхування.

При розробці програми, яка забезпечує інформаційне забезпечення систем страхування у банківських операціях, ергономічне проектування інтерфейсу користувача враховується з великою увагою, особливо з огляду на безпеку. Забезпечення ефективного та зручного взаємодії користувача з системою є критично важливим для підтримки безпеки та захисту даних.

При проектуванні інтерфейсу враховуються принципи зручності та доступності, що дозволяє користувачам легко орієнтуватися в системі та виконувати необхідні дії без зайвих зусиль. Запобігання помилкам та введення неправильних даних також є важливим аспектом ергономічного проектування інтерфейсу [27].

У той же час, важливо враховувати вимоги до безпеки під час взаємодії користувача з програмою. Це включає в себе використання механізмів автентифікації, таких як паролі, біометричні дані або двофакторна аутентифікація,

для перевірки ідентичності користувача перед доступом до конфіденційної інформації.

Крім того, важливо забезпечити конфіденційність та цілісність даних, що вводяться користувачем через інтерфейс. Це може включати в себе використання шифрування для захисту передачі даних по мережі, а також механізми перевірки цілісності даних для виявлення будь-яких спроб модифікації даних під час їх передачі.

Ергономічне проектування інтерфейсу користувача з огляду на безпеку передбачає забезпечення зручності та доступності для користувачів, одночасно з використанням механізмів захисту даних та перевірки ідентичності для забезпечення безпеки та конфіденційності усіх транзакцій та взаємодій з програмою.

У контексті забезпечення безпеки програми, необхідним етапом є тестування та аудит безпеки. Це процес, який спрямований на ідентифікацію потенційних уразливостей та проблем безпеки програмного забезпечення з метою їх виправлення перед експлуатацією програми.

Тестування безпеки може включати в себе різноманітні методики, включаючи сканування вразливостей, тестування на проникнення, аналіз коду, тестування на проникнення та інші техніки [6]. Ці методи допомагають виявити можливі проблеми безпеки, такі як вразливості в програмному коді, неправильні налаштування безпеки, а також потенційні шляхи атаки, які можуть бути використані зловмисниками.

Після проведення тестування безпеки зазвичай проводиться аудит безпеки, який включає в себе глибокий аналіз безпекових вимог, архітектури системи, методів аутентифікації та авторизації, шифрування даних та інших аспектів безпеки. Аудит допомагає виявити слабкі місця та рекомендувати стратегії для підвищення рівня безпеки програми.

Загалом, тестування та аудит безпеки є важливими етапами в розробці програмного забезпечення, оскільки вони дозволяють виявити та виправити потенційні проблеми безпеки перед їх використанням користувачами. Ці процеси допомагають забезпечити високий рівень безпеки програми та захистити користувачів від можливих загроз.

ВИСНОВКИ

У світлі зазначених технічних та програмних характеристик, вивчених під час аналізу, можна зробити висновок про ефективність та потенційні можливості інформаційного забезпечення даної системи. Розглянувши структуру веб-застосунку, варто зауважити, що його основними складовими є HTML, CSS та JavaScript, що забезпечують інтерактивність та зручність взаємодії з користувачем.

Особлива увага приділялася створенню інтерфейсу, який би був якісним та зручним для кінцевого користувача. Реалізація таких функцій, як калькулятор страхових виплат, система лояльності та вивчення типів страхування, вимагала впровадження високоякісної логіки та алгоритмів обробки даних.

Звернувши увагу на аспекти безпеки, слід відзначити, що застосунок має механізми захисту від неправомірного доступу до конфіденційної інформації користувачів. Це досягнуто за допомогою шифрування даних та застосування сучасних методів аутентифікації.

Підсумовуючи вищевикладене, можна стверджувати, що розроблений веб-застосунок є важливим інструментом для забезпечення інформаційних потреб у сфері страхування банківських операцій. Його інноваційні можливості та зручний інтерфейс відкривають нові перспективи для користувачів у здійсненні фінансових операцій та забезпеченні їх стабільності.

Важливим аспектом є також можливість масштабування системи та її пристосування до змін у вимогах ринку та розвитку технологій. Для досягнення цього були використані сучасні підходи до програмування та архітектурні рішення, що дозволяють швидко впроваджувати нові функції та покращувати ефективність системи.

Окрім того, важливо відзначити значення аналізу та збору даних для підтримки прийняття рішень у сфері страхування. Веб-застосунок забезпечує

зручний інтерфейс для збору та обробки інформації про користувачів, що дозволяє здійснювати аналітичні операції та прогнозування на основі цих даних.

Необхідно також зазначити, що розроблений веб-застосунок є лише першим кроком у напрямку розвитку інформаційного забезпечення в сфері страхування банківських операцій. Майбутній розвиток системи передбачає впровадження нових функціональних можливостей, розширення спектру послуг та підвищення якості обслуговування користувачів.

Додатковою перевагою розробленого веб-застосунку є його гнучкість та адаптивність до різних типів користувачів. Інтерфейс системи розроблений з урахуванням сучасних стандартів дизайну та може ефективно працювати на різних пристроях, включаючи комп'ютери, планшети та смартфони. Це дозволяє користувачам отримувати доступ до інформації про страхування у будь-який час та з будь-якого місця, забезпечуючи максимальний комфорт та зручність взаємодії.

Крім того, важливим аспектом є підтримка міжнародних стандартів безпеки та конфіденційності даних. Розроблений веб-застосунок відповідає вимогам GDPR та інших регулятивних актів, що забезпечує користувачам відчуття захищеності та конфіденційності їх особистих даних. Такий підхід сприяє підвищенню довіри користувачів до системи та підтримує стабільність функціонування у контексті суворих вимог до обробки особистої інформації.

З огляду на високий рівень функціональності та технічної складності розробленого веб-застосунку, важливим є постійний моніторинг та підтримка його роботи. Забезпечення безперебійної роботи системи та вчасне виявлення та усунення можливих проблем вимагає наявності кваліфікованого технічного персоналу та впровадження сучасних інструментів моніторингу та аналізу даних. Тільки за умови постійного удосконалення та підтримки система зможе ефективно виконувати свої функції та задовольняти потреби користувачів.

Загалом, розроблений веб-застосунок відкриває нові можливості для забезпечення інформаційних потреб у сфері страхування банківських операцій, створюючи зручні умови для користувачів та сприяючи розвитку фінансової сфери.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глєбова Н. В. Облік у банках : навч. посіб. Харків : ХНЕУ, 2009. 308 с.

2. Петрук О. М. Банківська справа : навч. посіб. Київ : Кондор, 2009. 466 с.

3. Ткаченко Н. В. Страхування : навч. посіб. для самост. роботи студентів вищ. навч. закл. Київ : Ліра-К, 2007. 376 с.

4. Ткаченко Н. В. Страхування : практикум : навч. посіб. для студ. вищ. навч. закл. Київ : Ліра-К, 2009. 270 с.

5. Banking, Insurance and Finance Union. New technology in banking, insurance and finance. BIFU.

6. Черевко О. В. Вдосконалення методології моніторингу ризиків легалізації доходів у банківській системі.

7. Стратегічні пріоритети детінізації економіки України у системі економічної безпеки: макро та мікро вимір : монографія / за ред. Черевка О. В. Черкаси: ПП Чабаненко Ю. А., 2014. 442 с.

8. Чайкін І.Б. Правове регулювання страхування ризиків на ринках фінансових послуг: монографія. Х.: Юрайт, 2012. 184 с.

9. Стрижиченко К.А. Формування концепції регулювання фінансового ринку України. Бізнес-Інформ. 2014. № 10. С. 324–330.

10. Рисін В. Критерії оцінки ризиків, пов’язаних з відмиванням грошей, у процесі формування ресурсної політики банку. Вісник Львівського університету. (Серія :Економіка). 2008. Вип. 39. С. 473–475.

11. Про фінансові послуги та державне регулювання ринків фінансових послуг: Закон України від 12.07.2001 р. № 2664-III. Відомості Верховної Ради України.2002. № 1. Ст. 1.

12. Acharya V. V., Pedersen L. H., Philippon T., Richardson M. P. Measuring systemic risk. AFA. Denver Meetings Paper. 2010. URL: http://dx.doi.org/10.2139/ssrn.1573171.

13. Achkasova S. Ensuring financial security of non-governmental pension funds in Ukraine. Economic Studies. 2018. No 1. pp. 152–172.

14. Achkasova S. Preventing and counteraction corruption risks in introducing and implementing riskbased approach in financial monitoring system. Organizationaleconomic mechanism of management innovative development of economic entities : collective monograph / edited by M. Bezpartochnyi Vol. 3. Higher School of Social and Economic. – Przeworsk : WSSG, 2019. Vol. 2. p. 82–91.

15. Amir R., Brander J., Zott C. Why do Venture Capital Firms Exist? Theory and Canadian Evidence of Business Venturing. 1998. vol. 13. pp.441–466.

16. Andriichenko Zh., Vnukova N., Chmutova I. Risk-based approach in the regulation of supervisory authorities in Ukraine : part in the monograph : Contemporary issues of sustainable development. Contemporary issues of sustainable development: Monograph. Opole: The Academy of Management and Administration in Opole, 2019. p. 8–17.

17. Pukała R., Kvasnytska R., Vnukova N., Achkasova S. The Scale Measurement of the Main Indicators of Capitalization of the Insurance Market (on the Example of Ukraine) Advances in Economics, Business and Management Research, Strategies, Models and Technologies of Economic Systems Management (SMTESM) 2019. Vol. 95. Atlantis Press, pp. 103–107.

18. Small and medium-sized enterprises' access to finance. URL: https://ec.europa.eu/ info/sites/info/files/file_import/european-semester_thematic-factsheet_smallmedium-enterprises-access-finance_en.pdf.

19. Ukraine's measures to combat money laundering and the financing of terrorism and proliferation: fifth round mutual evaluation report. December 2017. URL:

http://www. fatf-gafi.org/publications/mutualevaluations/documents/mer-ukraine-2017.htm.

20. Vnukova N. M., Kavun S. V., Kolodiziev O. M., Achkasova S. A., Hontar D. D. Determining the level of connectivity banks for combating money laundering, terrorist financing and proliferation of weapons of mass destruction. Banks and Bank Systems. Vol. 14. 2019. Issue 4. pp. 42–54.

21. Vnukova N., Kavun S., Kolodiziev O., Achkasova S., Gontar D. Indicators-Markers for Assessment of Probability of Insurance Companies Relatedness in Implementation of Risk-Oriented Approach. Economic Studies (Ekonomicheski Izsledvania), 32 (1), 2020. pp. 151–173.

22. Акімова О. В. Дослідження можливостей адаптації та використання міжнародних типологій в системі фінансового моніторингу України. Технологический аудит и резервы производства. 2016. № 5/4 (31). С. 51–57.

23. Андрійченко Ж. О. Визначення напрямів інституційних змін для забезпечення ефективного функціонування ризик-орієнтованого підходу у сфері фінансового моніторингу. Глобальні та національні проблеми економіки. 2017. Вип. 17. URL: http://globalnational.in.ua/issue-17-2017.

24. Андрійченко Ж. О., Літвінова С. О. Статистичне обґрунтування необхідності запровадження ризик-орієнтованого підходу у сфері фінансового моніторингу в Україні. Проблеми і перспективи розвитку підприємництва: Збірник наукових праць ХНАДУ. № 2 (17). 2017. Харків : ХНАДУ, 2017. С. 49–55.

25. Ачкасова С. А. Критерії віднесення фінансових операцій до сумнівних або незвичних. Сучасні проблеми фінансового моніторингу: Збірник наукових праць за матеріалами ІІІ Всеукраїнської науково-практичної конференції. Харків, 2013. С. 18–23.

26. Балануца О. О. Місце і роль фінансового моніторингу в Україні як основоположного чинника ефективної боротьби держави з легалізацією (відмиванням коштів) та фінансуванням тероризму. Збірник наукових праць

Національного університету державної податкової служби України. 2011. № 1. С. 5–42.

27. Бачо Р. Й. Формування концепції розвитку ринків небанківських фінансових послуг України. Електронне наукове фахове видання «Глобальні та національні проблеми економіки» Миколаївського національного університету імені В.О. Сухомлинського. 2016. № 13. URL: http://global-national.in.ua/archive/ 13-2016/132.pdf.

28. Безродна О. С., Лесик В. О. Теоретико-методичні аспекти оцінювання фінансової стабільності банківської системи. Проблеми економіки. 2017. № 2. С. 251–262.

29. Береславська О. І. Перспективи імплементації рекомендацій Базельського комітету в практичну діяльність банків України. Наукові записки Херсонського національного університету. Серія «Економіка». Випуск 23. 2013. С. 262–266.

30. Бондар М. І., Бондар Т. А. Ідентифікація клієнтів аудиторами (аудиторськими фірмами) у процесі здійснення фінансового моніторингу. Незалежний аудитор. 2014. № 7. С. 2–10.

31. Вейц О. І. Обґрунтування поняття легалізації доходів клієнтів банку. Бізнес Інформ. 2019. № 1. С. 337–342.

32. Внукова Н. М. Перспективи впровадження ризик-орієнтованого підходу у систему фінансового моніторингу. Концептуальні засади менеджменту та фінансів в умовах глобальної нестабільності: збірник матеріалів VI Міжнародної науково-практичної інтернет-конференції «Актуальні проблеми менеджменту та фінансів в сучасних глобалізаційних процесах» (14 березня 2019 р.). Ірпінь. Університет ДФС України, 2019. С. 194–196.

33. Дмитров С. О., Меренкова О. В., Дмитров С. О., Медвідь Т. А., Ващенко О. М. Оцінка та управління ризиком використання послуг для легалізації

кримінальних доходів або фінансування тероризму в комерційному банку : монографія. / за заг. ред. О. М. Бережного. Суми : ДВНЗ «УАБС НБУ», 2010. 114 с.

34. Єгоричева С. Б. Організація фінансового моніторингу в банках : навч. посіб. Київ : Центр учбової літератури. 2014. 292 с.

35. Єфименко Т. І., Гасанов С. С., Користін О. Є. та ін. Розвиток національної системи фінансового моніторингу. Київ: ДННУ «Академія фінансового управління», 2013. 380 с.

36. Жорнокуй Ю. М. Правове регулювання венчурного підприємництва (цивільноправовий аспект): дис. … канд. юрид. наук: 12.00.03 / Нац. ун-т внутр. справ. Харків, 2003. 184 с.

37. Касаткіна Т., Плахота А. Аналіз бізнес-моделей банків у рамках Supervisory review and evaluation process (SREP). Департамент банківського нагляду Національного банку України, 2018. URL: https://bank.gov.ua/doccatalog/document?id=69900832.

38. Коваленко В. В., Дмитров С. О., Єжов А. В. Міжнародний досвід у сфері запобігання та протидії відмиванню доходів, одержаних злочинним шляхом, та фінансуванню тероризму : монографія. Суми: УАБС НБУ. 2007. С. 110–140.

39. Коваленко В. В. Фінансовий моніторинг банків : навч. посіб. Суми : Мрія. 2012. 120 с.

40. Козак Ю. Г., Мацкул В. М. Математичні методи та моделі для магістрів з економіки. Практичні застосування : навч. посіб. Київ: Центр учбової літератури, 2017. 254 с.

ДОДАТКИ

Додаток А

Index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SafeOps - Домашня сторінка</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<header class="header">
  <h1 class="site-title">SafeOps</h1>
  <nav class="navMenu">
    <a href="index.html">Головна</a>
    <a href="info.html">Інфо</a>
    <a href="#">Про нас</a>
    <div class="dot"></div>
  </nav>
</header>

<div class="welcome-block">
  <p class="text">
    <span class="text__first">
      <span class="text__word">Welcome</span>
      <span class="text__first-bg"></span>
```

```
        </span>
        <br>
        <span class="text__second">
          <span class="text__word">to</span>
          <span class="text__second-bg"></span>
        </span>
        <span class="text__word">SafeOps</span>
      </p>
    </div>

    <h2 class="section-title">Огляд страхових продуктів в банківських операціях</h2>

    <section class="hero-section">
      <div class="card-grid">
        <div class="card">
          <div class="card__background" style="background-image: url(images/car_insurance.jpg)"></div>
          <div class="card__content">
            <p class="card__category">Страхування автомобілів</p>
            <h3 class="card__heading">Автострахування</h3>
          </div>
        </div>
        <div class="card">
          <div class="card__background" style="background-image: url(images/property_insurance.jpg)"></div>
          <div class="card__content">
            <p class="card__category">Страхування нерухомості</p>
```

```
            <h3 class="card__heading">Страхування житла</h3>
          </div>
        </div>
        <div class="card">
          <div class="card__background" style="background-image:
url(images/medical_insurance.jpg)"></div>
          <div class="card__content">
            <p class="card__category">Медичне страхування</p>
            <h3 class="card__heading">Медичне страхування</h3>
          </div>
        </div>
        <div class="card">
          <div class="card__background" style="background-image:
url(images/liability_insurance.jpg)"></div>
          <div class="card__content">
            <p class="card__category">Страхування відповідальності</p>
            <h3 class="card__heading">Цивільне страхування</h3>
          </div>
        </div>
      </div>
    </section>

    <div style="text-align: center; animation: fadeInUp 1s;">
      <h2 style="color: #f5f7fa; font-size: 36px; margin-bottom: 36px;">Переваги
страхування в банках</h2>
      <div style="margin: 20px auto; max-width: 800px; display: flex; justify-content:
space-between;">
        <div style="width: 30%; color: white; text-align: left;">
```

Захист від ризиків
Страхові продукти, надані банками, дозволяють клієнтам захистити свої фінансові активи від різноманітних ризиків, таких як непередбачені події або фінансові збитки.

Покращення фінансової стабільності
Страхування в банках дозволяє клієнтам уникнути великих фінансових втрат у випадку непередбачених обставин, забезпечуючи більшу фінансову стабільність та спокій.

Широкий вибір страхових продуктів
Банки пропонують різноманітні страхові продукти, включаючи страхування життя, медичне страхування, страхування автомобілів та інші, що відповідають різним потребам клієнтів.

```
<div style="text-align: center; animation: fadeInUp 1s;">
 <h2 style="color: #f5f7fa; margin-bottom: 24px; margin-top: 42px;">Статистика оформлення страхування в банку за 5 років</h2>
 <table style="margin: 20px auto; border-collapse: collapse; width: 80%; max-width: 800px;">
  <thead>
   <tr style="background-color: transparent; color: white;">
    <th style="border: none;">Рік</th>
    <th style="border: none;">Кількість оформлених страхувань</th>
    <th style="border: none;">Загальна кількість клієнтів</th>
    <th style="border: none;">Відсоток клієнтів, що оформили страхування</th>
   </tr>
  </thead>
  <tbody>
   <tr>
    <td style="color: white;">2019</td>
    <td style="color: white;">300</td>
    <td style="color: white;">1200</td>
    <td style="color: yellow;">25%</td>
   </tr>
   <tr>
    <td style="color: white;">2020</td>
    <td style="color: white;">400</td>
    <td style="color: white;">1400</td>
    <td style="color: yellow;">28.5%</td>
   </tr>
   <tr>
```

```
            <td style="color: white;">2021</td>
            <td style="color: white;">500</td>
            <td style="color: white;">1500</td>
            <td style="color: yellow;">33.3%</td>
          </tr>
          <tr>
            <td style="color: white;">2022</td>
            <td style="color: white;">550</td>
            <td style="color: white;">1600</td>
            <td style="color: yellow;">34.4%</td>
          </tr>
          <tr>
            <td style="color: white;">2023</td>
            <td style="color: white;">600</td>
            <td style="color: white;">1800</td>
            <td style="color: yellow;">33.3%</td>
          </tr>
        </tbody>
      </table>
    </div>

    <div style="text-align: center; margin-bottom: 100px;">
      <h2 style="color: #f5f7fa; margin-bottom: 24px; margin-top: 42px;">Чому варто обрати страхування в банку?</h2>
      <div style="margin: 20px auto; max-width: 800px; color: white; text-align: left;">
        <p style="font-size: 18px; line-height: 1.6;">
          <strong>Гарантована надійність:</strong><br>
```

Банки мають довгу історію стабільності та надійності, що робить їх відмінними партнерами у справі захисту вашого майна та фінансів.

Широкий вибір продуктів:

У банках ви знайдете різноманітні страхові продукти, які можуть відповідати вашим потребам, включаючи страхування майна, автомобілів, життя та багато інших.

Зручність та доступність:

Завдяки своєму присутності в багатьох регіонах, банки забезпечують зручний доступ до страхових послуг для клієнтів будь-якого рівня.

<!-- Анімовані крапки -->

<!-- Генерація крапок за допомогою JavaScript -->

<script>

for (var i = 0; i < 128; i++) {

var bubble = document.createElement("div");

bubble.className = "bubble";

bubble.style.setProperty("--size", 2 + Math.random() * 4 + "rem");

bubble.style.setProperty("--distance", 6 + Math.random() * 4 + "rem");

bubble.style.setProperty("--position", -5 + Math.random() * 110 + "%");

```
            bubble.style.setProperty("--time", 2 + Math.random() * 2 + "s");
            bubble.style.setProperty("--delay", -1 * (2 + Math.random() * 2) + "s");
            document.querySelector(".bubbles").appendChild(bubble);
          }
        </script>
      </div>
      <div class="content">
        <div>

        </div>
      </div>
      <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
      <script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"></script>


      <script>
            window.onload = function(){
        var tl = new TimelineLite({delay: 1}),
          firstBg = document.querySelectorAll('.text__first-bg'),
          secBg = document.querySelectorAll('.text__second-bg'),
          word  = document.querySelectorAll('.text__word');

        tl
          .to(firstBg, 0.2, {scaleX:1})
          .to(secBg, 0.2, {scaleX:1})
          .to(word, 0.1, {opacity:1}, "-=0.1")
          .to(firstBg, 0.2, {scaleX:0})
```

```
            .to(secBg, 0.2, {scaleX:0});

          document.querySelector('.restart').onclick = function() {tl.restart()};

        }

        </script>

        </body>
</html>
```

Додаток Б

Info.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SafeOps - Домашня сторінка</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<header class="header">
  <h1 class="site-title">SafeOps</h1>
  <nav class="navMenu">
    <a href="index.html">Головна</a>
    <a href="info.html">Інфо</a>
    <a href="#">Про нас</a>
    <div class="dot"></div>
  </nav>
</header>

<div style="text-align: center; animation: slideInUp 1s; max-width: 800px; margin: 0 auto; margin-bottom: 36px;">
  <h2 style="color: #f6f4e6; font-size: 46px; margin-bottom: 36px;">Страхові продукти в банківських операціях</h2>
  <div style="display: flex; flex-wrap: wrap; justify-content: center; gap: 20px; margin-top: 20px;">
```

<div style="width: calc(50% - 10px); background-color: #272727; border: 2px solid #f1c40f; border-radius: 10px; cursor: pointer; overflow: hidden;" onmouseover="this.style.backgroundColor='#f1c40f'; this.style.color='white'; this.children[0].style.transform='translateY(-10px)';" onmouseout="this.style.backgroundColor='#272727'; this.style.color=''; this.children[0].style.transform='translateY(0px)';">

<h3>Страхування вкладів</h3>

Страхування вкладів - це захист вкладників від втрат коштів, вкладених у банк. У разі банкрутства банку або інших фінансових проблем, страхова компанія компенсує втрати вкладників до певної суми.

<div style="width: calc(50% - 10px); background-color: #272727; border: 2px solid #f1c40f; border-radius: 10px; cursor: pointer; overflow: hidden;" onmouseover="this.style.backgroundColor='#f1c40f'; this.style.color='white'; this.children[0].style.transform='translateY(-10px)';" onmouseout="this.style.backgroundColor='#272727'; this.style.color=''; this.children[0].style.transform='translateY(0px)';">

<h3>Страхування кредитів</h3>

Страхування кредитів надає захист боржника та банку. Для боржника, це може бути захист від непередбачуваних обставин, таких як втрата роботи або непрогнозовані медичні витрати.

```
</div>

<div style="width: calc(50% - 10px); background-color: #272727; border: 2px solid #f1c40f; border-radius: 10px; margin-top: 20px; cursor: pointer; overflow: hidden;" onmouseover="this.style.backgroundColor='#f1c40f'; this.style.color='white'; this.children[0].style.transform='translateY(-10px)';" onmouseout="this.style.backgroundColor='#272727'; this.style.color=''; this.children[0].style.transform='translateY(0px)';">
    <div style="background-color: #272727; padding: 20px; border-radius: 8px; transition: transform 0.3s;">
        <h3>Страхування платіжних карток</h3>
        <p>Страхування платіжних карток може включати захист від крадіжки або незаконного використання картки. Деякі політики також можуть забезпечувати захист від шахрайства під час онлайн-покупок або гарантувати повернення грошей у випадку неудачних транзакцій.</p>
    </div>
</div>

<div style="width: calc(50% - 10px); background-color: #272727; border: 2px solid #f1c40f; border-radius: 10px; margin-top: 20px; cursor: pointer; overflow: hidden;" onmouseover="this.style.backgroundColor='#f1c40f'; this.style.color='white'; this.children[0].style.transform='translateY(-10px)';" onmouseout="this.style.backgroundColor='#272727'; this.style.color=''; this.children[0].style.transform='translateY(0px)';">
    <div style="background-color: #272727; padding: 20px; border-radius: 8px; transition: transform 0.3s;">
        <h3>Страхування життя</h3>
```

Страхування життя у банківських операціях може бути пропоноване як додаткова послуга для клієнтів. Воно забезпечує фінансовий захист для сім'ї в разі смерті основного годувальника сім'ї або іншої важливої особи.

<h2 style="color: #f6f4e6; font-size: 46px; margin-bottom: 36px;">Поради щодо вибору страхових продуктів</h2>

<div style="flex: 1; background-color: #f1c40f; padding: 20px; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); cursor: pointer; transition: background-color 0.3s;" onmouseover="this.style.backgroundColor='#f6f4e6';" onmouseout="this.style.backgroundColor='#f1c40f';">
<h3 style="color: #34495e;">Підбирайте продукти, які відповідають вашим потребам</h3>
Перш ніж купувати страховий продукт, з'ясуйте, які саме ризики ви хочете застрахувати. Ретельно оцініть свої потреби та обережно дослідіть всі варіанти перед прийняттям рішення.

<div style="flex: 1; background-color: #f1c40f; padding: 20px; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); cursor: pointer; transition: background-color 0.3s;" onmouseover="this.style.backgroundColor='#f6f4e6';" onmouseout="this.style.backgroundColor='#f1c40f';">

<h3 style="color: #34495e;">Розумійте умови та обмеження</h3>

Перед укладанням угоди ознайомтесь з умовами страхового поліса. Вивчіть умови покриття, виключення та обмеження, щоб уникнути неприємних сюрпризів у разі настання страхового випадку.

<div style="flex: 1; background-color: #f1c40f; padding: 20px; border-radius: 10px; box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1); cursor: pointer; transition: background-color 0.3s;" onmouseover="this.style.backgroundColor='#f6f4e6';" onmouseout="this.style.backgroundColor='#f1c40f';">

<h3 style="color: #34495e;">Звертайтеся до фахівців</h3>

Не соромтеся звертатися до страхового агента або консультанта, якщо у вас виникають питання щодо обраного страхового продукту. Вони можуть допомогти зрозуміти всі нюанси та підібрати оптимальний варіант.

<h2 style="text-align: center; color: #f6f4e6; font-size: 46px; margin-bottom: 36px;">Страховий калькулятор</h2>

<label for="amount" style="color: #f6f4e6">Сума страхування:</label>

<input type="number" id="amount" min="0" step="1000" placeholder="Введіть суму страхування">

```
        <label for="percentage" style="color: #f6f4e6">Відсоток виплати:</label>
        <input type="number" id="percentage" min="0" max="100" placeholder="Введіть відсоток виплати">

        <button onclick="calculate()">Розрахувати</button>

        <div id="result"></div>

    </div>

    <div class="loyalty-program">
        <h2>Програми лояльності</h2>
        <p>При оформленні страхового полісу в нашому банку ви отримуєте можливість приєднатися до наших програм лояльності. Це дозволить вам отримувати додаткові переваги та бонуси при зверненні до нас.</p>
        <ul>
            <li>Збільшена кількість балів за кожну виплачену премію</li>
            <li>Ексклюзивні пропозиції для учасників програми</li>
            <li>Додаткові знижки на інші послуги банку</li>
            <li>Можливість обміну балів на подарункові сертифікати та інші призи</li>
        </ul>
        <p>Не пропустіть шанс стати учасником програми лояльності та отримувати ще більше вигод!</p>
    </div>
```

```
<div class="footer">
  <div class="bubbles">
    <!-- Анімовані крапки -->
    <!-- Генерація крапок за допомогою JavaScript -->
    <script>
function calculate() {
  var amount = parseFloat(document.getElementById('amount').value);
  var percentage = parseFloat(document.getElementById('percentage').value);

  if (isNaN(amount) || isNaN(percentage)) {
    document.getElementById('result').innerText = 'Будь ласка, введіть коректні значення.';
    return;
  }

  var payout = amount * (percentage / 100);
  document.getElementById('result').innerText = 'Ваша виплата складе ' + payout.toFixed(2) + ' грн.';
}
      for (var i = 0; i < 128; i++) {
        var bubble = document.createElement("div");
        bubble.className = "bubble";
        bubble.style.setProperty("--size", 2 + Math.random() * 4 + "rem");
        bubble.style.setProperty("--distance", 6 + Math.random() * 4 + "rem");
        bubble.style.setProperty("--position", -5 + Math.random() * 110 + "%");
        bubble.style.setProperty("--time", 2 + Math.random() * 2 + "s");
        bubble.style.setProperty("--delay", -1 * (2 + Math.random() * 2) + "s");
```

```
            document.querySelector(".bubbles").appendChild(bubble);
          }
        </script>
      </div>
      <div class="content">
        <div>

        </div>
      </div>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"></script>


    <script>
          window.onload = function(){
      var tl = new TimelineLite({delay: 1}),
        firstBg = document.querySelectorAll('.text__first-bg'),
        secBg = document.querySelectorAll('.text__second-bg'),
        word  = document.querySelectorAll('.text__word');

      tl
        .to(firstBg, 0.2, {scaleX:1})
        .to(secBg, 0.2, {scaleX:1})
        .to(word, 0.1, {opacity:1}, "-=0.1")
        .to(firstBg, 0.2, {scaleX:0})
        .to(secBg, 0.2, {scaleX:0});
```

```
        document.querySelector('.restart').onclick = function() {tl.restart()};

    }

    </script>

    </body>
</html>
```

Додаток В

Styles.css

```
@import
url("https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600;700&dis
play=swap");
* {
  margin: 0;
  padding: 0;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}

body {
  background: #272727;
  font-family: "Montserrat", sans-serif;
}


.header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 20px;
}

.section-title {
  text-align: center; /* Центруємо текст */
  font-size: 2em; /* Збільшуємо розмір шрифту */
```

```
  color: white; /* Білий колір тексту */
}

.site-title {
  color: #f6f4e6;
  font-size: 1.5em;
  font-weight: 600;
  text-transform: uppercase;
}

.navMenu {
  display: flex;
  gap: 20px;
  align-items: center;
}

.navMenu a {
  color: #f6f4e6;
  text-decoration: none;
  font-size: 1.2em;
  text-transform: uppercase;
  font-weight: 500;
  position: relative;
}

.navMenu a:hover {
  color: #fddb3a;
}
```

```
.navMenu .dot {
  width: 6px;
  height: 6px;
  background: #fddb3a;
  border-radius: 50%;
  opacity: 0;
  position: absolute;
  bottom: -10px; /* Збільшено відступ */
  left: 50%;
  transform: translateX(-50%);
  transition: opacity 0.2s ease-in-out;
}

.navMenu a:hover .dot {
  opacity: 1;
}

.footer {
  z-index: 1;
  --footer-background: #FADB5F; /* Жовтий колір */
  display: grid;
  position: relative;
  grid-area: footer;
  min-height: 12rem;
}

.bubbles {
```

```
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  height: 1rem;
  background: var(--footer-background);
}

.bubble {
  position: absolute;
  left: var(--position, 50%);
  background: var(--footer-background);
  border-radius: 100%;
  animation: bubble-size var(--time, 4s) ease-in infinite var(--delay, 0s),
  bubble-move var(--time, 4s) ease-in infinite var(--delay, 0s);
  transform: translate(-50%, 100%);
}

.content {
  z-index: 2;
  display: grid;
  grid-template-columns: 1fr auto;
  grid-gap: 4rem;
  padding: 2rem;
  background: var(--footer-background);
}

.content a,
```

```
.content p {
  color: #f5f7fa;
  text-decoration: none;
}

.content b {
  color: white;
}

.content p {
  margin: 0;
  font-size: 0.75rem;
}

.content > div {
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.content > div > div {
  margin: 0.25rem 0;
}

.content > div > div > * {
  margin-right: 0.5rem;
}
```

```
@keyframes bubble-size {
  0%, 75% {
    width: var(--size, 4rem);
    height: var(--size, 4rem);
  }
  100% {
    width: 0rem;
    height: 0rem;
  }
}

@keyframes bubble-move {
  0% {
    bottom: -4rem;
  }
  100% {
    bottom: var(--distance, 10rem);
  }
}


.welcome-block {
  text-align: center;
  margin-bottom: 50px;
}

.text {
```

```
  display: inline-block;
  font-size: 6vw; /* Зменшено розмір шрифту */
  line-height: 1.205;
  color: #ffffff; /* Білий колір тексту */
}

.text__first,
.text__second {
  position: relative;
}

.text__word {
  opacity: 0;
}

.text__first-bg,
.text__second-bg {
  display: block;
  width: 100%;
  height: 100%;
  position: absolute;
  left: 0;
  top: 0;
  z-index: 100;
  transform-origin: left;
  transform: scaleX(0);
}
```

```
.text__first-bg {
  background-color: #FADB5F;
}

.text__second-bg {
  background-color: #F06543;
}

.text__second {
  margin-left: 10px; /* Зменшена відстань від першого блоку */
}

:root{
  --background-dark: #272727;
  --text-light: rgba(255,255,255,0.6);
  --text-lighter: rgba(255,255,255,0.9);
  --spacing-s: 8px;
  --spacing-m: 16px;
  --spacing-l: 24px;
  --spacing-xl: 32px;
  --spacing-xxl: 64px;
  --width-container: 1200px;
}


.hero-section{
  align-items: flex-start;
  background-image: linear-gradient(15deg, #272727 0%, #272727 150%);
```

```
  display: flex;
  min-height: 100%;
  justify-content: center;
  padding: var(--spacing-xxl) var(--spacing-l);
}

.card-grid{
  display: grid;
  grid-template-columns: repeat(1, 1fr);
  grid-column-gap: var(--spacing-l);
  grid-row-gap: var(--spacing-l);
  max-width: var(--width-container);
  width: 100%;
}

@media(min-width: 540px){
  .card-grid{
    grid-template-columns: repeat(2, 1fr);
  }
}

@media(min-width: 960px){
  .card-grid{
    grid-template-columns: repeat(4, 1fr);
  }
}

.card{
```

```
  list-style: none;
  position: relative;
}

.card:before{
  content: '';
  display: block;
  padding-bottom: 150%;
  width: 100%;
}

.card__background{
  background-size: cover;
  background-position: center;
  border-radius: var(--spacing-l);
  bottom: 0;
  filter: brightness(0.75) saturate(1.2) contrast(0.85);
  left: 0;
  position: absolute;
  right: 0;
  top: 0;
  transform-origin: center;
  transform: scale(1) translateZ(0);
  transition:
    filter 200ms linear,
    transform 200ms linear;
}
```

```
.card:hover .card__background{
  transform: scale(1.05) translateZ(0);
}

.card-grid:hover > .card:not(:hover) .card__background{
  filter: brightness(0.5) saturate(0) contrast(1.2) blur(20px);
}

.card__content{
  left: 0;
  padding: var(--spacing-l);
  position: absolute;
  top: 0;
}

.card__category{
  color: white; /* Білий колір тексту категорії */
  font-size: 0.9rem;
  margin-bottom: var(--spacing-s);
  text-transform: uppercase;
}
.card__heading{
  color: white; /* Білий колір заголовку */
  font-size: 1.6rem; /* Зменшено розмір шрифту */
  text-shadow: 2px 2px 20px rgba(0,0,0,0.2);
  line-height: 1.4;
  word-spacing: 100vw;
}
```

```
.information-block {
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  gap: 20px;
  margin-top: 50px;
}
.info-block {
  background-color: #f6f4e6; /* Сірий колір фону */
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /* Тінь */
}
.info-title {
  font-size: 1.2em; /* Збільшено розмір шрифту */
  color: #272727; /* Чорний колір тексту */
  margin-bottom: 10px;
}
.info-text {
  font-size: 0.9em; /* Зменшено розмір шрифту */
  color: #4e4e4e; /* Темно-сірий колір тексту */
}
    .color-container {
      width: 1200px !important;
      padding: 0 !important;
      margin-right: auto;
      margin-left: auto;
      margin: 20px auto;
    }
```

```
.color-container-title {
  font-size: 36px;
  color: white;
  text-align: center;
  margin-bottom: 50px;
}
.gradient-cards {
  display: flex;
  justify-content: space-between;
}
.color-card {
  width: 30%;
  background-color: transparent;
}
.color-container-card {
  padding: 30px;
  border-radius: 24px;
  box-shadow: 0px 8px 24px rgba(0, 0, 0, 0.1);
}
.color-card-title {
  font-size: 24px;
  color: white;
  margin-top: 20px;
  margin-bottom: 10px;
}
.color-card-description {
  font-size: 18px;
  color: white;
```

```
    opacity: 0.8;
  }
  .bg-green-box {
    background-color: #54E8A9;
  }
  .bg-white-box {
    background-color: #2E3042;
  }
  .bg-yellow-box {
    background-color: #FFEE24;
  }
  .bg-blue-box {
    background-color: #87A1FF;
  }
    @keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translate3d(0, 100%, 0);
  }
  to {
    opacity: 1;
    transform: none;
  }
}
table, th, td {
  border: 1px solid white;
  padding: 10px;
  text-align: center;
```

```
  }
   @keyframes scaleIn {
   from {
     transform: scale(0.9);
     opacity: 0;
   }
   to {
     transform: scale(1);
     opacity: 1;
   }
  }
.calculator {
  width: 300px; /* Встановіть бажану ширину калькулятора */
  margin: 0 auto; /* Центрування калькулятора */
  background-color: #272727; /* Чорний фон */
  padding: 20px; /* Відступи від країв */
  border-radius: 10px; /* Закруглені кути */
  margin-bottom: 24px;
}
.calculator input[type="number"] {
  width: 100%; /* Ширина 100% */
  padding: 10px; /* Відступи всередині */
  margin: 10px 0; /* Відступи між елементами */
  border: 1px solid #ccc; /* Обведення з гарною каркасом */
  border-radius: 5px; /* Закруглені кути */
  box-sizing: border-box; /* Забезпечення відповідної ширини */
  background-color: #f9f9f9; /* Світлий сірий фон */
}
```

```
.calculator button {
  width: 100%; /* Ширина 100% */
  padding: 10px 20px; /* Відступи всередині */
  border: none; /* Відсутність меж */
  background-color: #f1c40f; /* Жовтий колір */
  color: white; /* Білий колір тексту */
  border-radius: 5px; /* Закруглені кути */
  cursor: pointer; /* Показувати вказівник руки */
  transition: background-color 0.3s; /* Плавний перехід */
}
.calculator button:hover {
  background-color: #d4ac0d; /* Темніший колір при наведенні */
}
.calculator #result {
  margin-top: 20px; /* Відступ зверху */
  text-align: center; /* Вирівнювання по центру */
  font-weight: bold; /* Жирний шрифт */
  color: white; /* Білий колір тексту */
}
.loyalty-program {
    background-color: #272727;
    color: white;
    padding: 20px;
    border-radius: 10px;
    max-width: 600px;
    margin: 0 auto;
    text-align: center;
    box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
```

```
        margin-bottom: 76px;
    }
    .loyalty-program h2 {
        font-size: 24px;
        margin-bottom: 10px;
    }
    .loyalty-program p {
        font-size: 16px;
        margin-bottom: 20px;
    }
    .loyalty-program ul {
        list-style-type: none;
        padding: 0;
        margin-bottom: 20px;
    }

    .loyalty-program ul li {
        font-size: 16px;
        margin-bottom: 10px;
    }

    .loyalty-program p:last-child {
        margin-bottom: 0;
}
```