Форма № Н-9.02

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
(повне найменування закладу вищої освіти)
Навчально-науковий інститут інформаційних технологій і робототехніки
(повне найменування інституту, назва факультету (відділення))
Кафедра автоматики, електроніки та телекомунікацій
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до кваліфікаційної роботи

бакалавра
(ступінь вищої освіти)

на тему «**Розроблення комплексного методу захисту інформації в сучасних бездротових мережах»**

Виконав: студент 3 курсу, групи 301-пТТ
спеціальності
172 «Телекомунікації та радіотехніка»
(шифр і назва напряму підготовки, спеціальності

Мирошниченко М. В.
(прізвище та ініціали)
Керівник Косенко В.В.
(прізвище та ініціали)
Рецензент Шефер О.В.
(прізвище та ініціали)

Полтава - 2023 рік

Національний університет «Полтавська політехніка імені Юрія Кондратюка»
Інститут Навчально-науковий інститут інформаційних технологій і робототехніки
Кафедра Автоматики, електроніки та телекомунікацій
Освітній рівень бакалавр
Спеціальність 172 «Телекомунікації та радіотехніка»

ЗАТВЕРДЖУЮ
завідувач кафедри автоматики,
електроніки та телекомунікацій

_____ д.т.н., проф. О.В. Шефер
“ __ ” ___ 2023 р.

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Мирошниченку Максиму Віталійовичу

1. Тема проекту (роботи) **«РОЗРОБЛЕННЯ КОМПЛЕКСНОГО МЕТОДУ ЗАХИСТУ ІНФОРМАЦІЇ В СУЧАСНИХ БЕЗДРОТОВИХ МЕРЕЖАХ»**

керівник проекту (роботи) Косенко Віктор Васильович, д.т.н., професор
затверджена наказом вищого навчального закладу від “20” 03 2023 року № 236-фа

2. Строк подання студентом проекту (роботи) 14.06.2023 р.

3. Вихідні дані до проекту (роботи). Вихідними даними кваліфікаційної роботи є матеріали отримані під час переддипломної практики.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області існуючих бездротових мереж з метою захисту інформації. На основі проведеного аналітичного огляду, формування вимог до захисту інформації та працездатності сайтів користувачів. Розроблення комплексу заходів та засобів захисту інформації. Вибір технології розробки та середовища програмування. Вибір засобів реалізації конфіденційності інформації користувача. Практична реалізація розробленого комплексного методу захисту інформації. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових плакатів):
Автономна архітектура мережі. Архітектура скоординованої бездротової мережі. Статистичний аналіз кібератак. Аналіз моніторингу доступності та перевірки роботи сервісів. Функціонал Host tracker, UptimeRobot, Monitis, Site24x7, Winginx nf OpenServer. Статична архітектура додатку в фреймворкові Yii2. Діаграма варіантів використання системи безпеки. Сайти користувача для моніторингу системи безпеки.

6. Консультанти розділів роботи

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

7. Дата видачі завдання 10.04.23 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Термін виконання етапів роботи		**Приміт ка**
1.	Аналіз існуючих бездротових мереж з метою захисту інформації. Формування вимог до захисту інформації та працездатності сайтів користувачів.	26.04.23	25%	Пл. 1
2.	Розроблення комплексу заходів та засобів захисту інформації. Вибір технології розробки та середовища програмування.	10.05.23	50%	Пл. 2
4.	Вибір засобів реалізації конфіденційності інформації користувача. Практична реалізація розробленого комплексного методу захисту інформації.	24.05.23	70%	Пл. 4
5.	Висновки. Підготовка графічних матеріалів.	07.06.23		Пл. 5
6.	Оформлення кваліфікаційної роботи.	14.06.23	100%	Пл. 6

Мирошниченко М.В.

Студент _____ _____
(підпис) (прізвище та ініціали)

Керівник роботи _____ Косенко В. В.
(підпис) (прізвище та ініціали)

ЗМІСТ

Вступ……...5
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТі……7
1.1 Аналіз існуючих бездротових мереж в розрізі захисту інформації……..7
1.2 Аналітичний огляд моніторингу сайтів для забезпечення працездатності під впливом кібератак……..……19
1.3 Аналіз існуючих систем моніторингу захисту інформації на сайтах та їх порівняння……....28
1.4 Формування вимог до захисту інформації та працездатності сайтів користувачів……36
2 РОЗРОБЛЕННЯ КОМПЛЕКСНОГО ЗАХИСТУ ІНФОРМАЦІЇ……..37
2.1 Вибір локального серверу безпекового рівня……...37
2.2 Вибір технології розробки ……...….43
2.3 Вибір засобу реалізації бази даних з метою забезпечення конфіденційності інформації користувача ……49
2.4 Вибір середовища програмування……...51
2.5 Проектування бази даних для забезпечення безпеки інформації користувача ……57
2.6 Діаграма прецедентів……61
3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОГО КОМПЛЕКСНОГО МЕТОДУ……..…70
Висновки……83
Список використаних джерел……84
Додатки……86

ВСТУП

Бездротові технології, поза сумнівом, стали проривом у світі телекомунікацій. Легкий монтаж, низька вартість і можливість підключення роблять бездротові мережі популярними. Високий попит для бездротових мереж і незахищений зв'язок пристроїв IoT викликає різні категорії загроз безпеці інформації [2, 3]. Це створює загрозу конфіденційності даних, таких як особиста інформація, фінансові дані та медичні дані та інформація військової таємниці. Взаємозв'язок між проблемами безпеки та конфіденційності пов'язаний один з одним, й багато рішень були запропоновані дослідниками у формі схем автентифікації, використання штучного інтелекту та техніки машинного навчання для подолання атак [4, 5]. Безпека – це загальний термін, який включає основні атрибути конфіденційності, які стосуються шифрування та захист конфіденційної інформації від несанкціонованого доступу з боку інших сторін. Слід відзначити, що інформація може бути доступною лише для авторизованих осіб. Другий атрибут - це цілісність інформації, що гарантує факт захисту інформації, цілісність та оригінальність інформації, а також унеможливлення втручання зловмисними користувачами. Це забезпечує точність і узгодженість даних мережі. Останнім атрибутом є доступність, що означає вільне надання користувачам системи авторизаційного доступу. Поширеною проблемою доступності є DoS-атаки, які порушують доступ до інформації та системи, вцілому. Отже, безпека в бездротовій мережі, а також в Інтернеті речей полягає у збереженні даних від несанкціонованого доступу, або доступ за допомогою кодування.

Бездротові локальні мережі забезпечують високі характеристики передачі всередині та за межами офісів і робочих місць. Користувачі таких мереж зазвичай використовують ПК з процесорами та великими екранами, здатними запускати ноутбуки, персональні комп'ютери та додатки, що потребують

великих ресурсів. Співробітник може скористатися послугами мережі в конференц-залі або в інших приміщеннях будівлі. Це дозволяє працівнику ефективно виконувати свої обов'язки. Бездротові локальні мережі можуть задовольнити вимоги всіх офісних або домашніх програм на швидкість до 54 Мбіт/с. За характеристиками, компонентами, вартістю та продуктивністю такі мережі схожі на традиційні дротові локальні мережі типу Ethernet.

Бездротові регіональні мережі обслуговують територію всього міста. У більшості випадків додатки вимагають виділеного з’єднання, а іноді, потрібна мобільність. Наприклад, у лікарні така мережа забезпечує передачу даних між головним корпусом і віддаленими клініками. ПрАТ «Полтавобленерго» використовує таку мережу в масштабах міста й області, щоб забезпечити доступ до роботи з різних районів.

У результаті бездротові регіональні мережі об’єднують існуючу мережеву інфраструктуру або дозволяють мобільним користувачам підключатися до існуючої мережевої інфраструктури. Постачальники послуг бездротового Інтернету (WISP) надають бездротові регіональні мережі в містах і сільській місцевості для забезпечення регулярних бездротових з’єднань для домашніх користувачів і компаній. Такі мережі часто більш ефективні, ніж звичайні дротові з'єднання, які мають обмеження, пов'язані з прокладкою дротових з'єднань. Характеристики бездротової регіональної мережі відрізняються, особливо, собівартістю. Використання інфрачервоної технології в комунікації забезпечує швидкість передачі даних 100 Гбіт/с і вище. Бездротові глобальні мережі дозволяють використовувати мобільні програми в різних країнах або навіть на континентах. Виходячи з економічних міркувань, телекомунікаційні компанії створюють відносно дорогу інфраструктуру для бездротової глобальної мережі, яка забезпечує підключення на великій відстані для багатьох користувачів. Вартість такого рішення розподіляється між усіма користувачами, тому абонентська плата не дуже висока.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз існуючих бездротових мереж в розрізі захисту інформації

Наступні категорії бездротових мереж відрізняються залежно від розміру фізичної області, до якої надається з’єднання:

- Бездротова персональна мережа Personal Area Network (PAN);
- Бездротова локальна мережа Local Area Network (LAN);
- Бездротова міська мережа Mine Area Network) (MAN):
- Бездротова глобальна мережа Wide Area Network (WAN).

Бездротові приватні мережі характеризуються малою дальністю передачі (до 17 метрів) і використовуються в невеликій будівлі. Характеристики таких мереж середні, а швидкість передачі зазвичай не перевищує 5 Мбіт/с. Така мережа, наприклад, може забезпечувати бездротову синхронізацію даних користувача. Аналогічно передбачено бездротове з'єднання з принтером. З’єднання стаціонарного комп’ютера із зовнішніми пристроями, є значною перевагою, оскільки це значно полегшує початкове встановлення зовнішніх пристроїв, а потім, якщо необхідно, їх переміщення.

Оскільки бездротова мережа обслуговує користувача, його можна розглядати як важливу частину бездротової мережі. Користувач починає процес використання бездротової мережі і завершує його самостійно. Тому його допустимо називати «кінцевим користувачем», або «споживачем». Як правило, користувач взаємодіє з ПК, який виконує інші завдання, пов’язані з певними програмами, крім взаємодії з бездротовою мережею.

Використання бездротових технологій має багато переваг. Хоча ця технологія дає користувачам відчуття, що вони можуть пересуватися, не

втрачаючи зв’язку, вона надає чудову можливість розробникам мереж створювати зв’язки. Це також дозволяє створювати багато нових пристроїв для використання мережі.

Однак, слід зазначити, що бездротові технології становлять більше загроз, ніж звичайні дротові мережі. Щоб створити захищену бездротову програму, потрібно визначити всі маршрути, якими можуть передаватися бездротові «атаки». На жаль, програми ніколи не бувають повністю безпечними, але ретельне вивчення ризиків бездротових технологій може допомогти підвищити рівень захисту в будь-якому випадку. Це означає аналіз потенційних загроз і побудову мережі таким чином, щоб вона була здатна запобігати атакам і бути готовою до захисту від нестандартних «атак».

Основна відмінність між дротовими та бездротовими мережами полягає в абсолютно неконтрольованій зоні між кінцевими точками мережі. У досить широкій зоні стільникових мереж бездротове середовище ніколи не контролюється. Сучасні бездротові технології пропонують обмежений набір засобів управління мережевим простором. Це дозволяє зловмисникам поблизу бездротових структур здійснювати атаки, які неможливі при дротовому сполученні.

Бездротова мережа – це комбінація різних мереж, які дозволяють одному комп’ютеру отримувати доступ до іншого без засоби проводового з'єднання фізично [6].

У наші дні найбільш поширені бездротові канали зв'язку є радіосигнали у вигляді частотного діапазону. Найпоширенішим прикладом цього є з’єднання Wi-Fi мережі. Бездротові мережі реалізуються у фіз рівень, який є рівнем 1 еталонної моделі OSI [7]. Точки доступу використовуються для встановлення підключення до мережі у бездротовому середовищі. Ця точка доступу є типом апаратний пристрій, який виявляє та дозволяє мережу віддалений доступ. Наступним апаратним пристроєм вважається найважливішою частиною

бездротових мереж є маршрутизатор пристрій, який забезпечує фізичний спосіб спілкування різн мережі [8]. Бездротові точки доступу ефективно працюють із радіоприймачем для розробки з’єднання, яке дозволяє як передачу та прийом радіосигналів [9]. Ці сигнали приймаються клієнтськими пристроями, які ідентифікують сигнали, і після підтвердження каналів зв'язку він надає далі доступ до мережі. Бездротові точки доступу приймають загальний стандарт бездротового зв'язку.

Існує два основних типи архітектури бездротової мережі, які обговорюються нижче: Автономна архітектура (також відома як режим Ad-hoc): У архітектурі Ad-hoc усі пристрої підключаються напряму спілкування так само, як і однорангове з’єднання [8]. для налаштування в режимі Ad-hoc, потрібна ручна настройка замість автоматизованого процесу та відсутності точки доступу, наприклад для зв'язку потрібен маршрутизатор/комутатор. Такий тип архітектура використовується в невеликому середовищі, напр. централізований домен бізнесу [9]. Спеціальна архітектура бездротової мережі ілюстрація наведена на рис. 1.1 [10].

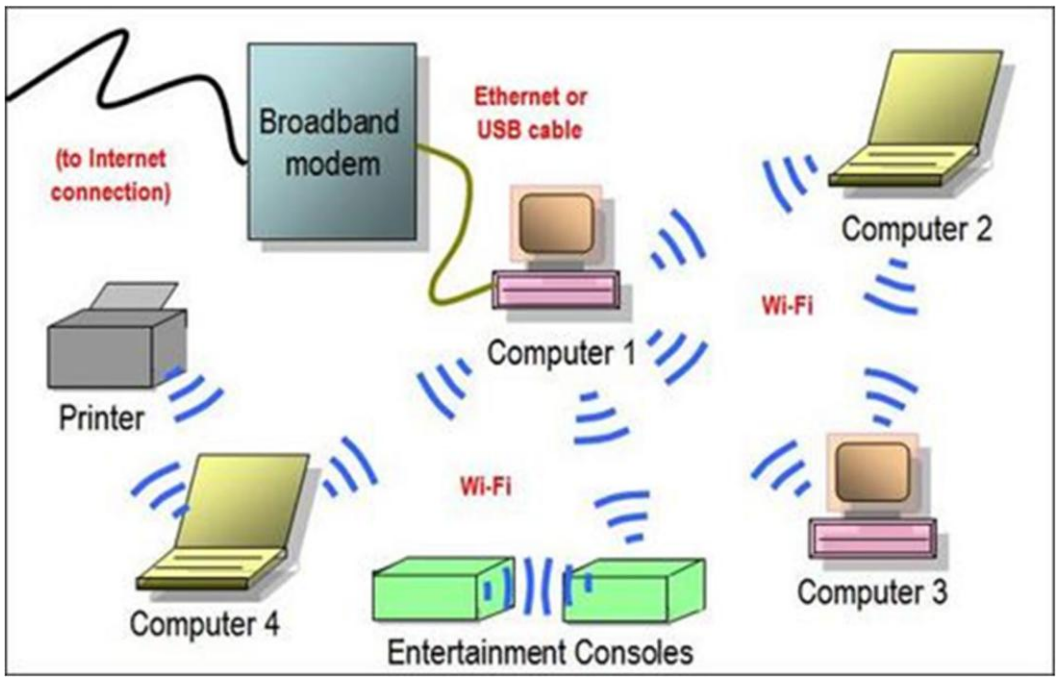


Рисунок 1.1 – Автономна архітектура мережі (Ad-hoc)

Централізовано скоординована архітектура (також відома як інфраструктура режиму). Пристрої підключаються за допомогою точки доступу, це означає, що для зв’язку потрібен маршрутизатор/комутатор. Такий тип архітектури використовується у великому середовищі, наприклад, розподілений бізнес-домен [10]. Централізовано Архітектура скоординованої бездротової мережі наведена на рис. 1.2 [10].

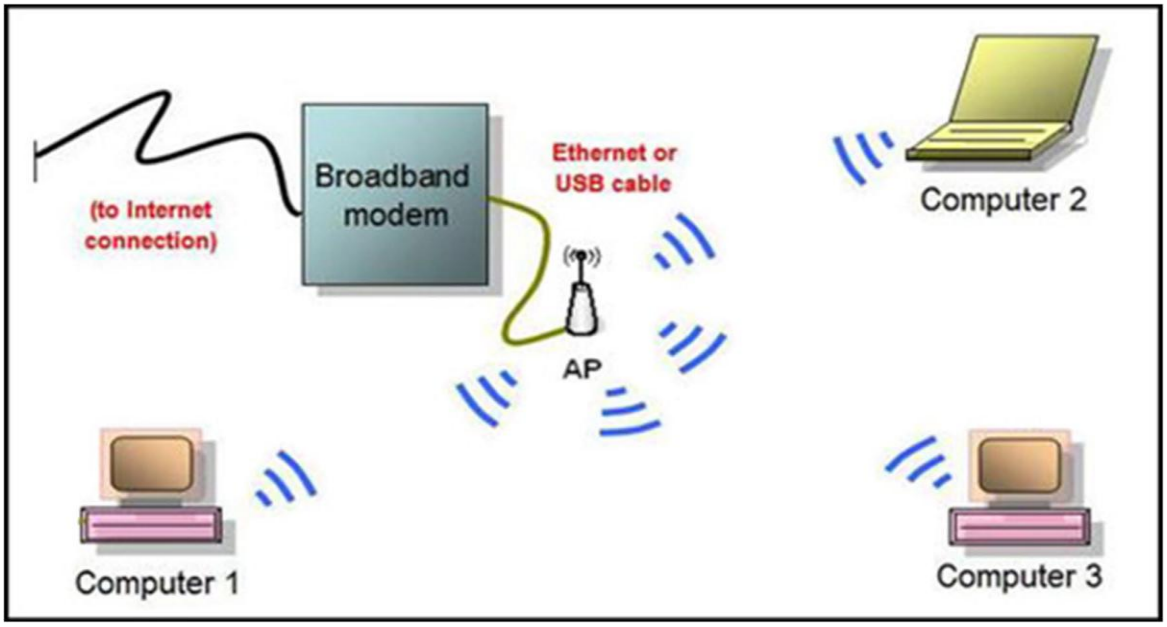


Рисунок 1.2 – Архітектура скоординованої бездротової мережі

Найбільш поширеною проблемою у відкритому та некерованому середовищі, такому як бездротові мережі, є можливість анонімних атак. Збої в мережі виникають, коли навмисне чи ненавмисне втручання перевищує можливості відправника та одержувача в каналі зв’язку. У результаті цей канал часто блокується. Зловмисник може використовувати різні методи. Відмова від надання послуг. Така атака, як DoS (Denial of Service), може повністю вимкнути мережу. По всій мережі, включаючи базові станції та клієнтські термінали, існує така сильна перешкода, що станції не можуть спілкуватися одна з одною. Ця атака блокує всі комунікації в межах певного діапазону. Важко запобігти

або зупинити атаку DoS на бездротову мережу. Більшість технологій бездротових мереж використовують неліцензовані частоти, що означає, що перешкоди можуть створюватися кількома електронними пристроями.

Блокування клієнтської станції дозволяє шахраю поставити себе на місце клієнта. Він також використовується для відмови в обслуговуванні клієнта, щоб завадити йому встановити з'єднання. Намір дуже вмілих атак розширює існуюче з'єднання, щоб з'єднати пошкоджену станцію користувача з базовою станцією. Заблокувати клієнтську станцію протокол WLTS. Протокол WLTS на основі SSL/TLS використовується в пристроях WAP (Wireless Application Protocol), таких як мобільні телефони та КПК. SSL і WLTS відрізняються один від одного рівнем трафіку. SSL покладається на TCP для перенаправлення втрачених пакетів або передачі нестандартних пакетів. Користувачі WLTS, які використовують WLTS, не можуть використовувати TCP для виконання своїх функцій, оскільки вони використовують лише UDP (протокол датаграм користувача).

Протокол UDP не призначений для підключення, тому ці функції повинні бути включені в WLTS. Протокол 802.1x. Основною функцією цього протоколу є аутентифікація; у деяких випадках протокол можна використовувати для встановлення ключів шифрування. Після підключення лише 802.1x. протокол конфігурації DHCP (Dynamic Host Configuration Protocol), IP і h. такі протоколи не допускаються. Розширюваний протокол автентифікації (EAP) (RFC 2284) використовується для автентифікації користувача.

Використання бездротових технологій має багато переваг. Хоча ця технологія дає користувачам відчуття, що вони можуть пересуватися, не втрачаючи зв'язку, вона надає чудову можливість розробникам мереж створювати зв'язки. Це також дозволяє створювати багато нових пристроїв для використання мережі.

Але бездротові технології становлять більше загроз, ніж звичайні дротові мережі. Щоб створити захищену бездротову мережу, потрібно визначити всі маршрути, якими можуть передаватися бездротові «атаки». На жаль, мережі ніколи не бувають повністю безпечними, але ретельне вивчення ризиків бездротових технологій може допомогти підвищити рівень захисту в будь-якому випадку. Це означає аналіз потенційних загроз і побудову мережі таким чином, щоб вона була здатна запобігати атакам і бути готовою до захисту від нестандартних «атак».

Основна відмінність між дротовими та бездротовими мережами полягає в абсолютно неконтрольованій зоні між кінцевими точками мережі. У досить широкій зоні стільникових мереж бездротове середовище ніколи не контролюється. Сучасні бездротові технології пропонують обмежений набір засобів управління мережевим простором. Це дозволяє зловмисникам поблизу бездротових структур здійснювати атаки, які неможливі в дротовому світі.

Найбільш поширеною проблемою у відкритому та некерованому середовищі, такому як бездротові мережі, є можливість анонімних атак.

Збої в мережі виникають, коли навмисне чи ненавмисне втручання перевищує можливості відправника та одержувача в каналі зв’язку. У результаті цей канал вимкнено. Зловмисник може використовувати різні методи.

Така атака, як DoS (Denial of Service), може повністю вимкнути мережу. По всій мережі, включаючи базові станції та клієнтські термінали, існує така сильна перешкода, що станції не можуть спілкуватися одна з одною. Ця атака блокує всі комунікації у межах певного діапазону. Важко запобігти або зупинити атаку DoS на бездротову мережу. Більшість технологій бездротових мереж використовують неліцензовані частоти, це означає, що перешкоди можуть створюватися кількома електронними пристроями.

Під конфіденційністю розуміємо право користувача запобігти розкриттю особистої інформації конкретним особам, або організаціям. Порушення конфіденційності зазвичай починається з порушення безпеки, але не обов'язково. Програми, яким потрібно збирати особисту інформацію від користувачів формально задовільнятимуть критеріям політики конфіденційності, які визнаються користувачами. Однак, насправді, це може призводять до порушення конфіденційності. Питання конфіденційності [10] стосуються того, як веб-сайт і організація обробляють, зберігання та використовують інформацію. Загалом, конфіденційність стосується збереження персональних даних від несанкціонованого доступу організацій або окремих осіб.

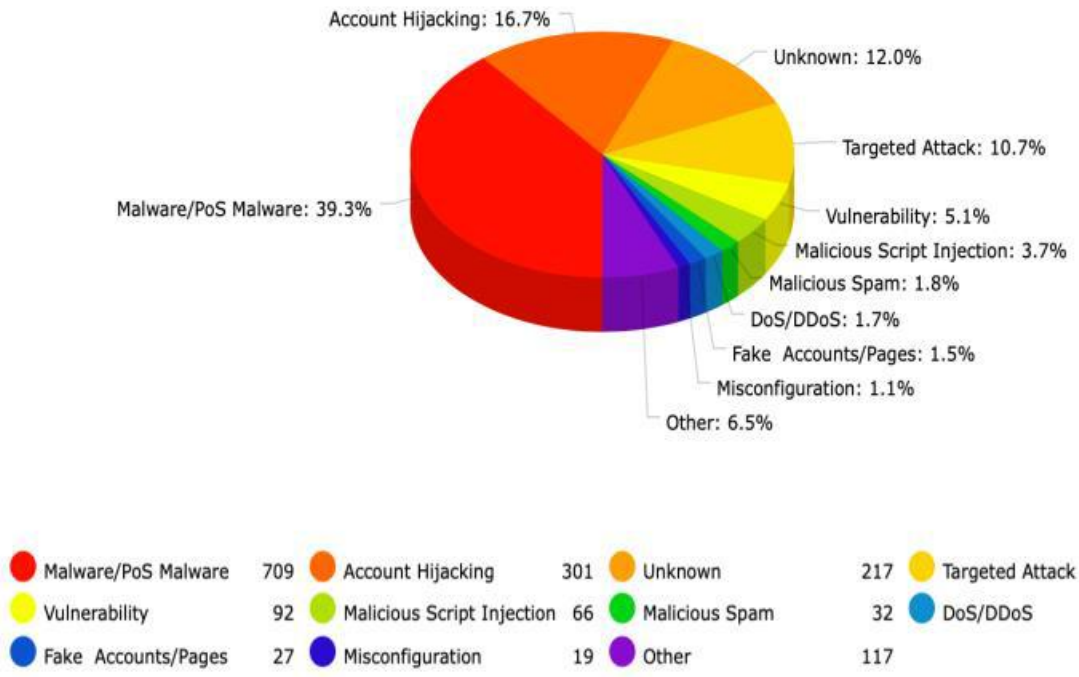


Рисунок 1.3 – Статистика різних кібератак у 2022 році

На рисунку 1.3, [11] показано, що найпопулярніша кібератака належить шкідливому програмному забезпеченню, що доводить, що громадськості рекомендується не нехтувати та не ігнорувати ці тривожні проблеми. У цьому

аналізі критично досліджується питання безпеки та конфіденційності в бездротовій мережі. Є надія, що вміст зможе показати важливість захисту користувачів і водночас підвищення обізнаності серед суспільства.

Інформація на тему захисту даних також викликала у громадськості критичне мислення дозволяючи їм виконувати свою активну роль у бездротовій мережі. Ці критерії прагнуть вплинути на громадськість шляхом застосування якісних та надійних ресурсів [11] з різноманітних дослідницьких баз даних, таких як IEEE [12] та ResearchGate [13].

У будь-якому випадку, дана аналітика робить внесок у суспільство, оскільки він представлений у ефективний спосіб, щоб повідомити користувачам, що безпека та питання конфіденційності не обмежується їхніми думками.

Безпека полягає у запобіганні потрапляння небажаного трафіку в мережу, а конфіденційність — у запобіганні пошуку інформацію від виходу з мережі. Отже, питання безпеки та проблеми конфіденційності — це дві різні теми, хоча вони передбачають певний вид несанкціонованого доступу для досягнення цілей зловмисника.

Проблеми безпеки [13] виникають, коли зловмисник отримує несанкціонований доступ до веб-сайту або системи. З іншого боку, проблеми конфіденційності включають необґрунтований доступ до конфіденційної та особистої інформації, яка не на 100% містить критерії порушення безпеки.

Відмова в обслуговуванні (DoS) і розподілені DoS (DDoS) атаки. Атака на відмову в обслуговуванні — це явна спроба перешкодити законним користувачам використовувати потрібні ресурси. Ан зловмисник може здійснити цю атаку, переповнюючи мережу трафіком, таким чином запобігаючи законному мережевому трафіку. DoS атаки включають порушення з’єднань між кількома машинами, запобігання доступу до певної служби. Це також може порушувати роботу системи або окремих осіб.

Крім того, DDoS [13] здійснює цей тип атаки ще важче запобігти. Хакерські атаки впливають не лише на цільовий комп'ютер, а й на хост-комп'ютер. Для розгортання цих шкідливих програм потрібно зловмисникам отримати доступ і проникнути на головний комп'ютер. Головна програма управління використовується для координації атаки і дозволяє нападнику залишатися прихованим під час атаки.

Кроки під час DDoS:

1. Справжній зловмисник надсилає повідомлення головній програмі керування для виконання.

2. Головна програма керування передає команду доменам атаки після отримання виконання повідомлення.

3. Потім алгоритм цих шкідливих програм почнає атаку на цільову жертву після отримання команди атаки.

Атаки спуфінгу [14] — це випадки, коли зловмисники маскують частину інформації з невідомого джерела як надійне джерело. Існує багато типів спуфінгу таких, як IP-спуфінг і DNS-спуфінг.

IP-спуфінг IP-спуфінг [14] — це створення IP-пакетів із підробленими вихідними IP-адресами, щоб приховати особу відправника, або систему. IP використовується під час передачі даних між машинами через Інтернет і кожного пакета, який надсилає власну IP-адресу, яка ідентифікує джерело інформації. Ця атака використовується для скоєння кіберзлочинів онлайн або для порушення безпеки мережі.

IP-спуфінг не дозволяє зловмисникам приховати джерело повідомлень, також не неможливо визначити через підроблену IP-адресу. З іншого боку, зловмисники порушують безпеку мережі використовуйте IP-адресу, яка збігається з однією з IP-адрес, дійсних у мережі.

Категорія безпеки — це класифікація вразливостей або загрози, з якими може зіткнутися система інформації процеси в реальному часі. Ці категорії

базуються на різних таких факторів, як потенційний вплив будь-якої події або її може бути результатом будь-якої неправильної практики маніпуляції.

Компанії класифікують свої проблеми безпеки відповідно до свого сприйняття моделями або іноді їх функціональними практиками.

Серед них можуть бути різні форми слабких місць безпеки всі найпоширеніші - це дезінформація, фальсифікація статистики інформація, крадіжка або пошкодження даних, на основі Інтернету хакерство та внутрішні маніпуляції співробітників.

Як показав аналіз, атаки DDOS [15] є однією з найпоширеніших типів активних атак, які дозволяють порушникам закону викрасти конфіденційну інформацію шляхом перехоплення системи.

Зловмисники використовують стратегію під назвою кешування пам’яті, де підроблений пакет надсилається потенційній жертві, а потім заливається сервер з агресивним трафіком.

Переходячи до проблем конфіденційності, щоб глибше заглибитися в висновки, фішинг викликає дедалі більше занепокоєння у громадськості. Коли люди залишаються вдома та використовують бездротові мережі, щоб залишатися на зв’язку з іншими, є можливість, коли люди, швидше за все, не лише спілкуватимуться, але й працюватимуть, поза звичайним захистом конфіденційності вони вразливі та більш сприйнятливі до фішингових атак.

Незалежно від того, які проблеми, є рішення, запропоновані дослідниками та вони продовжують розвивати їх відповідно до вимог. Наприклад, у [12-18] автори запропонували виявлення атак на протоколи зв’язку, що використовуються в мережах Інтернету речей, тоді як у [19] технологію RFID пропонується використовувати для безпечного моніторингу відвідуваності. У роботі [16-18] автори провели комплексний аналіз питання захисту конфіденційності в програмах хмарних обчислень, які можна використовувати як основу для розробки безпечних рішення в контексті

мобільних хмарних обчислень. Подібним чином було проведено багато інших досліджень спеціалізовані контексти, які здатні забезпечити обізнаність про проблеми безпеки та конфіденційності в поточному бездротовому зв’язку мережевих технологій, Інтернеті речей і комунікаційні протоколи для цих технологій.

Ефективна безпекова класифікація необхідна для розуміння та ідентифікації загроз та їх потенційний вплив. Можна спостерігати загрози безпеці і класифікуцію різними способами, враховуючи різні критерії.

Наведені нижче принципи повинні бути прийняті для класифікації питань безпеки та розробки конкретних моделей.

Модель загроз відрізняється від компанії до компанії, але загальні ефекти залишаються винятковими у функціональності.

На сьогоднішній день, зважаючи на велику інформатизацію суспільства, потребу у мобільності користувачів мережі Internet все більшою і більшою популярністю користуються бездротові мережі. Сучасний розвиток бездротових мереж дозволяє встановлювати з'єднання такої ж якості, як і мережі з використанням фізичного середовища передачі даних, але із значно більшою кількістю користувачів. Але при цьому виникає необхідність захисту переданої інформації в таких типах мереж, тобто використання механізмів захисту і шифрування даних. Так як загрози інформаційним ресурсам, в деяких випадках, можуть бути великими і катастрофічними. Тому актуальним є дослідження методів підвищення ефективності захисту інформації в бездротових комп’ютерних мережах.

Для проведення досліджень необхідно провести аналіз можливих загроз, визначити результати їх впливу та здійснити системний аналіз методів захисту від них.

Базовим стандартом, який визначає набір протоколів для передачі даних в бездротових мережах є IEEE 802.11. Цей стандарт постійно доповнюється та оновлюється.

Існують основні та допоміжні методи захисту бездротових мереж. Основними протоколами, які використовуються на даний час в бездротових мережах є: WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access), WPA2 (Wi-Fi Protected Access II), а також стандарт IEEE 802.1X, який описує процес інкапсуляції даних EAP (Extensible Authentication Protocol).

Найпростішим способом захисту від криптографічних атак є використання WPA2. WPA / WPA2, що значно підвищує безпеку бездротової мережі. Але додатковий захист відбувається за рахунок додаткової складності протоколу. На високому рівні, WPA атаки можна розділити на дві категорії: атаки аутентифікації та атаки шифрування.

Реалізація протоколу безпеки WPA2, безумовно, є кращим вирішенням проблем безпеки бездротової мережі. Поряд з тим, доповненням до основних рекомендацій можуть бути такі: приховування паролів, використання MAC фільтрації для підключених пристроїв та побудова складних паролів.

Для управління мережевою структурою і типами пакетів, які передаються бездротовими мережами, були проаналізовані поняття і методи протоколів захисту за допомогою документації, представленої IEEE, основних протоколів, таких як: WEP, WPA, WPA2 і основних типів шифрування, які вони використовують.

Отже, істотне збільшення використання бездротових мереж призвело до розробки механізмів безпеки, які спочатку були подолані зловмисниками, тому необхідне комплексне рішення для захисту мережі. Для цього доцільно впровадити технологію, яка включає в себе: створення пароля, створення плану забезпечення безпеки і захисту програмного забезпечення, що дозволяє більшу складність і безпеку бездротової мережі.

1.3 Аналітичний огляд моніторингу сайтів для забезпечення їх працездатності під впливом кібератак

Спектр соціальних груп, що підключаються до мережі Інтернет і що шукають інформацію в WWW, весь час розширюється за рахунок користувачів, що не відносяться до категорії фахівців в області інформаційних технологій: лікарі, будівельники, історики, юристи, фінансисти, спортсмени, мандрівники, священнослужителі, артисти, письменники, художники. Список можна продовжувати нескінченно. Той, хто відчув корисність і незамінність Мережі для своєї професійної діяльності або захоплень, приєднується до величезної армії споживачів інформації у «Всесвітньому Павутинні».

Web-технології повністю перевернули наші уявлення про роботу з інформацією і з комп'ютером. Виявилося, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна здатність, ємність запам'ятовуючих пристроїв, не враховували головного: «вузького місця» системи -– інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І тільки коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, відбувся безпрецедентний вибух інтересу до можливостей обчислювальної техніки.

З розвитком технологій гіпертекстової розмітки в Інтернеті стало з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів великих компаній, що оповідають про успіхи компанії і її провали, до сайтів маленьких фірм, що пропонують відвідати їх офіси в межах одного міста.

Розвиток Інтернет технологій став поштовхом до появи нової гілки в Інтернеті – Інтернет форумів. Стали з'являтися сайти, і навіть цілі портали, на

яких люди з усіх куточків планети можуть спілкуватися, отримувати відповіді на різні питання і, навіть, укладати ділові угоди.

Web-сайти використовуються як механізм спілкування між власниками сайту і його користувачами, а, іноді – між самими користувачами. Власники сайтів зазвичай ставлять завдання і визначають основні правила взаємодії, в той час як користувачі – це ті люди, що відвідують сайт і намагаються користуватися представленим на ньому вмістом або його можливостями. Канал зв'язку між власником сайту і його відвідувачем може змінюватися. Найчастіше власники сайтів надають користувачам інформацію для її споживання, роблячи з цього частково односторонню взаємодію.

Web-сайти можна розсортувати за кількома широкими категоріями.

Інформаційні сайти. На таких сайтах представлена інформація щодо конкретної теми або про певну організацію. Це самі поширені в мережі Internet Web-сайти.

Операційні сайти. Сайтом такого типу можна скористатися з метою виконання будь-якої операції або завдання. У цю категорію входять сайти, зайняті в електронній комерції.

Сайти спільнот. На цих сайтах представлена інформація або кошти, пов'язані із здійсненням операцій, але упор робиться на взаємодію між відвідувачами. Сайти, засновані на спільнотах, мають тенденцію до фокусування на конкретній темі або людині; вони заохочують взаємодію між схоже мислячими особистостями.

Розважальні сайти. Ці сайти створюються для ігор або якогось цікавої взаємодії, для якої можуть вживатися елементи операційного, інформаційного типів і сайтів спільнот.

Інші сайти. У цю категорію входять художні або експериментальні сайти, особисті Web-простори, такі як Web-журнали, а також сайти, які можуть не

слідувати загальноприйнятим Web-угодам або не мати чітко певного економічного призначення.

Крім цього, можна згрупувати сайти на основі організацій, які підтримують або в якомусь сенсі платять за сайт. В рамках цього типу класифікації визначимо п'ять основних груп, що перераховані нижче.

Корпоративні сайти. Сайт з цієї групи створюється і підтримується організацією або індивідуумом для отримання комерційної вигоди - або безпосередньо за допомогою електронної комерції, або побічно через стимулювання придбання товарів або послуг поза Internet.

Урядові сайти. Вищим органом по відношенню до такого сайту є урядова організація, а призначенням сайту є задоволення будь-якої суспільної або правової потреби.

Освітні сайти. Сайт такого типу курирує якась освітня установа (можливо, що має відношення до урядових органів).

Філантропічні сайти. Філантропічний сайт існує з метою просування цілей некомерційної організації або благодійної діяльності приватної особи або організації.

Персональні сайти. Такий сайт існує виключно на розсуд якоїсь людини або групи людей по будь-яким причинам, зазвичай будучи плодом виплеску творчої енергії або формою самовираження особистості;.

Класифікація може виявитися складним завданням. Наприклад, освітні сайти насправді можуть потрапляти в категорію урядових. Деякі сайти з категорії персональних можуть, ймовірно, належати до групи філантропічних або комерційних - в залежності від причини, за якою людина береться за створення Інтернет ресурсу.

Сьогоднішні підприємства залежні від IT. Кожна частина бізнесу тепер потребує підключення до Інтернету для роботи, а не тільки для фільмів під час перерв; від спілкування електронною поштою, обміну миттєвими

повідомленнями та VoIP, до бек-офісу ERP та CRM - не кажучи вже про важливість каналів цифрового маркетингу та електронної комерції.

Застосування та навіть послуги, що постають перед споживачами, тепер переходять на хмарні та гібридні моделі, оскільки компанії усвідомлюють, що внутрішня IT, коли вона виконує завдання з керування сервером Exchange і підтримує настільні комп'ютери, не має знань про інструменти та спеціалістів, щоб вони працювали.

Постійна залежність від Інтернету для залучення бізнесу призводить до загрози з можливістю трансформування конкурентних переваг нового світу бізнесу в хмарі в бізнес-вбивцю.

Простоювання сайту призводить до втрати близько 900 000 доларів США на тиждень для компанії з приблизно 10 000 працівників. Компанія Dun & Bradstreet продемонструвала, що більше половини компаній Fortune 500 мають щонайменше 1,6 години простою кожен тиждень. Прямі збитки є суттєвими, а також означають серйозний ризик втрати довіри клієнтів. Це набагато складніше для кількісного визначення, але тим не менш це є складним завданням для утримання клієнтів і є серйозним бар'єром для зростання на вищому рівні. Таким чином, будь-яка форма незапланованого простою програми є токсичною для успіху в бізнесі незалежно від ролі, яку ця програма відіграє у бізнесі.

Найкращим способом запобігання простою сайтів та усунення втрат підприємствами є прийняття ряду запобіжних заходів, які допоможуть досягти високої доступності для послуги або програми. Методики високої доступності використовуються для підтримки безперебійного обслуговування як можна довше – зазвичай лише допускається пробій 0,001% (приблизно 5 хвилин на рік).

Звичайний хостинг-провайдер може забезпечити лише 99% доступності послуги, тобто 87 годин (3,62 дні) простоїв в рік. Навіть обіцянка 99.9%

безвідмовної роботи дозволяє приблизно дев'ять годин простою в рік. Незважаючи на це, бізнес все ще може відчувати значну продуктивність і втрати клієнта протягом такого періоду часу; особливо, якщо зупинки простою в пікові періоди.

Щоб зрозуміти, як безвідмовна робота сприяє здоровому бізнесу, розглянемо те, що відбувається в простою, протилежному випадку.

Довіра клієнтів і лояльність. Забезпечення надійного та безперебійного обслуговування сприяє зміцненню відносин з клієнтами. Ніщо не погіршує імідж бренда швидше, ніж веб-сайт компанії, котрий недоступний клієнтам. Навіть якщо компанія не має нічого спільного з технологією, невдачі ставлять під загрозу лояльність ваших клієнтів. Ми живемо в епоху, коли підключення до Інтернету та розширення, доступність, вважається схожим на утиліту, яка завжди існує, як електрика чи вода.

Немає даремного часу. Час відмови від обслуговування. У випадку простою клієнти втрачають час. Також втрачається час на обмін даними з провайдером, доки проблема не буде вирішена, і, нарешті, додатковий час витрачається на відстеження замовлень та клієнтів. Врешті-решт, втрачений час призводить до втрати продуктивності. Якщо веб-сайт недоступний протягом 2 годин, а спеціалісти компанії не можуть отримати роботу, фактично втрачаються гроші, що дорівнюють погодинній ставці кожного співробітника. Веб-сайт, який безперервно працює, немає таких проблем.

Немає втраченого доходу. Це, мабуть, найважливіший момент. Відключення веб-сайту робить продукцію, послуги та бренд недоступним усьому світі, поки він перебуває в недоступному стані. Клієнти не можуть придбати продукцію, компанія не може рекламувати свої товари та послуги, а власник втрачаєте гроші. Однак це залежить від того, наскільки бізнес генерує прибуток за допомогою онлайн-засобів. Сайт електронної комерції, який

залежить від часу безвідмовної роботи, для отримання доходу страждає більше, ніж магазин, наприклад, канцелярії.

Кращий рейтинг SEO. Нарешті, але не менш важливо те, що Google відслідковує веб-сайт і може покарати його, якщо він неодноразово знаходиться в автономному режимі. Хоча інциденти простою, навіть ті, що в один день, напевно не завдадуть шкоди рейтингу веб-сайту, висока вартість роботи в режимі реального часу сприяє досягненню загального довгострокового показника SEO.

Мати власний сайт або сервер – це половина справи. Необхідно постійно стежити за його працездатністю і розуміти, що означають ті чи інші абревіатури із заголовних англійських букв. Така робота вимагає постійної присутності поруч з комп'ютером і тісного контакту зі спеціалізованими програмами.

Система моніторингу представляє собою спеціальний сервіс, щохвилини здійснює перевірку працездатності серверів і сайтів, що відслідковує їх стан, який збирає і обробляє необхідні для цього дані. Перевірка роботи сайту відбувається в будь-який час дня і ночі.

Моніторинг інтернет-сайтів включає в себе перевірку зміни контенту, в тому числі під впливом вірусів, його захист від негативних впливів і перевірку протоколів HTTP, HTTPS. Моніторинг інтернет-серверів здійснює перевірку їх доступності за допомогою інструменту ping, працездатності служб SMTP, IMAP, POP, FTP, MySQL, PostgreSQL і TCP порту. Якщо в роботі сайту відбувається збій, виникає несправність, сервіс повинен автоматично повідомляє про це власника або адміністратору ресурсу декількома способами (через електронну пошту, ICQ, за допомогою SMS тощо).

Існують такі методи перевірки доступності, працездатності основних систем і підсистем сайту:

– регулярна перевірка працездатності сайту;

– моніторинг доступності;

– моніторинг проблем.

Регулярна перевірка працездатності сайту. Цей спосіб перевірки досить простий і зробити це може кожен самостійно. Відкрити потрібний сайт, якщо працює, все перевірили, закривайте. Але це не дуже зручно і неефективно. Все питання полягає в тому, що складнощі з доступністю виникають непередбачено. А ось знати про них краще відразу. Користувачі, повідомляючи про неполадки з сайтом, вносять в вирішення проблеми свою лепту. Основний показник того, що сайт не працює, – відсутність клієнтів. У підсумку проблему доводиться вирішувати негайно, в той час, як її можна уникнути.

Моніторинг доступності. Фахівцям, які відповідають за роботу сайту, відомо, що за сайтом потрібно стежити постійно. Саме це завдання і стоїть перед моніторингом доступності. Від правильного вибору частоти моніторингу залежить успіх цієї операції і надійна робота всього ресурсу. Один раз на добу перевірити не складно, але ніхто не дає гарантії, що після перевірки сайт не може зламатися. В результаті простій в цей час принесе збитки.

Ідеальним варіантом буде перевірка доступності кожні десять хвилин. Найчастіше відвідувачі повертаються на сайт з проміжком одна – дві години. В цьому випадку буде достатньо часу для пошуку проблеми і рішення її.

Не дає гарантії швидкого вирішення проблеми навіть її виявлення в найкоротші терміни. Не складно знайти проблему,а складно з'ясувати причини її виникнення та визначити швидкий і ефективний спосіб усунення неполадок.

Моніторинг проблем. Під час критичного рівня доступності сайту, коли виникають різні проблеми з доступністю, як наслідок різних причин і обставин, доводиться перейти на новий рівень і відмовитися від звичайного моніторингу. Тепер уже виникає необхідність контролю не одного, а декількох параметрів сайту. При цьому потрібно стежити за сайтом з частотою один раз в хвилину. Для того, щоб максимально відстежувати хвилинний інтервал, краще це робити з різних географічних адрес. Це дасть низку переваг: максимальне покриття

інтервалу, можна помітити проблеми, які можуть бути пов'язані з географією користувачів.

Основні параметри перевірки:

– виникають проблеми з DNS-сервером. Сайт доступний, але в якийсь момент адреса сайту не визначається;

– довге очікування відповіді. Це може бути пов'язано з: оновленням кешу, складними завданнями на сервері;

– виконання планових завдань. Знову-таки результатом цього буде тимчасова недоступність сайту;

– тривалий час завантаження статичних файлів. Ця проблема може виникнути внаслідок проблем з мережевою інфраструктурою або фізичним носієм;

– виникають проблеми при з'єднанні з базою даних.

Вирішити ці проблему можна за допомогою щохвилинного моніторингу, для перевірки потрібно використовувати кілька точок. Як альтернативу можна використовувати незалежні сервіси або точки для перевірки доступу. Моніторинг періодичний і недовговічний. Періодичний моніторинг використовується в основному для профілактики. Недовговічний моніторинг використовується, якщо потрібно знайти і виправити неполадки.

Як варіант можна користуватися внутрішньою перевіркою, коли сайт займається самоперевіркою доступності. Але зовнішня перевірка все таки краще. Зовнішні сервіси можуть відразу надавати інформацію про проблему. Якщо зробити всі налаштування коректно, то завжди можливо бути в курсі всіх помилок, як з боку сервера, так і з боку сайту [2].

Спосіб зовнішньої діагностики доступності, добре зарекомендував себе, в тих випадках, коли потрібно виявити «плаваючу» помилку. Якщо в налаштуваннях увімкнені детальні логи, то з’являються помилки, навіть при відсутності деталей помилки на стороні сервера, буде можливість відстежити і

виправити її. Так само ефективність моніторингу доступності з різних точок, піднімає частоту перевірки до декількох разів на хвилину. І в цьому випадку гарантія своєчасного виявлення і усунення проблем виростає в декілька разів.

Моніторинг роботи сайту. Для Інтернет магазинів головне – можливість замовлень і прийом запитів. Тому це питання стосується їх більшою мірою. Сюди можна віднести і інші сайти, які відрізняються своєю складною функціональністю. Одним з наочних прикладів є кабінет Інтернет банкінгу, наприклад сервіс Приват 24. У таких ситуаціях потрібно розробляти складні ланцюги перевірок або як варіант, встановлювати складні параметри для перевірок.

У цьому випадку буде дуже до речі грамотний фахівець в цій області, послугами таких і користуються серйозні організації. У разі відсутності можливості оплатити такого роду послуги можна скористатися сервісами, вони дають можливість автоматизувати процес моніторингу працездатності фактично повністю, як би це робив виділений спеціаліст.

Для наочності позитивного впливу моніторингу, потрібно просто порівняти витрати на нього і втрати доходу при недоступності сайту. Оцінивши ці показники, потім можливо визначити ефективність моніторингу. Після цього можна зробити висновки, потрібен фахівець чи ні.

1.3Аналіз існуючих систем моніторингу захисту інформації на сайтах та їх порівняння

Ресурс для власників сайтів і системних адміністраторів ping-admin.ru (рис. 1.4) здійснює цілодобовий моніторинг доступності та перевірку робот на сервері таких сервісів [3]:

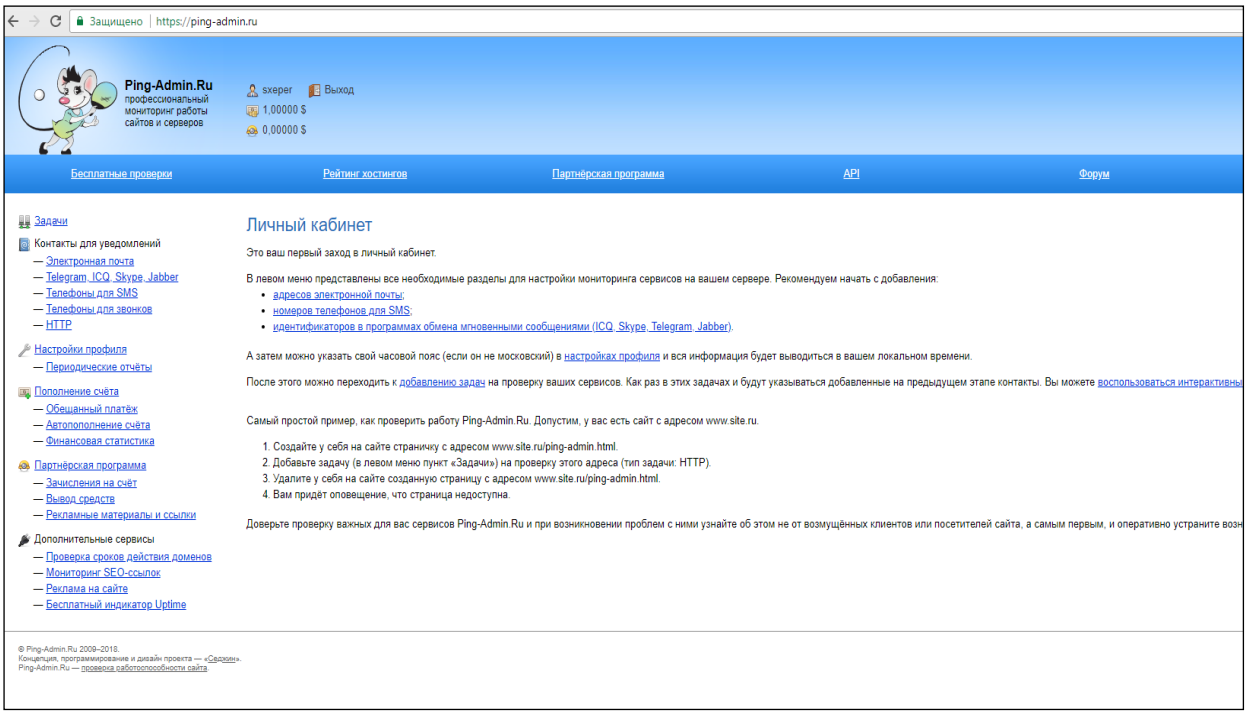


Рисунок 1.4 – Зовнішній вигляд Ping-admin

Специфікація Ping-admin:

– HTTP, HTTPS (перевірка працездатності сайту);

– FTP;

– MySQL, PostgreSQL (перевірка працездатності бази даних);

– POP3, SMTP, IMAP (перевірка працездатності пошти);

– DNS; – Ping; – Telnet.

При відмові сервісу, відповідне повідомлення оперативно відправляється адміністратору на зазначені адреси електронної пошти, на телефон у вигляді SMS або дзвінком, RSS або за допомогою запиту HTTP GET до скрипту.

Сервіс моніторингу офіційно підтримує нижченаведені регулярні перевірки:

– перевірка сайту на віруси (антивірусна перевірка);

– регулярна перевірка SEO-посилань (noindex,nofollow, robots. txt і т.д.);

– внутрішні ресурси сервера: навантаження, жорсткий диск, uptime в днях і будь-які інші;

– контроль терміну дії SSL-сертифікатів.

Перевірка доступності сайту через моніторинг працездатності його сервісів забезпечує повноцінний контроль над проектом.

Система Host-tracker.com (рис. 1.5) має функціонал, що дозволяє сканувати вказаний ресурс з необхідною періодичністю, при виникненні проблем з доступом, сайт буде перевірятися всім точками моніторингу, і якщо доступ до ресурсу неможливий, системою буде відправлено повідомлення за допомогою e-mail або SMS-повідомлення [4].

Використання декількох точок моніторингу дозволяє виключити помилкові спрацьовування, коли проблеми з доступом виникають, з причин не пов'язаних з працездатністю ресурсу, а через проблеми, обумовлених відсутністю зв'язку між сервером, яка здійснює перевірку і сайтом користувача.

Після виявлення помилки, система продовжує відстежувати статус ресурсу, і як тільки він стане знову доступний, висилає повідомлення про відновлення його працездатності, із зазначенням часу простою. Це досить зручно. Системний адміністратор або власник ресурсу, знає скільки не працював його сайт.

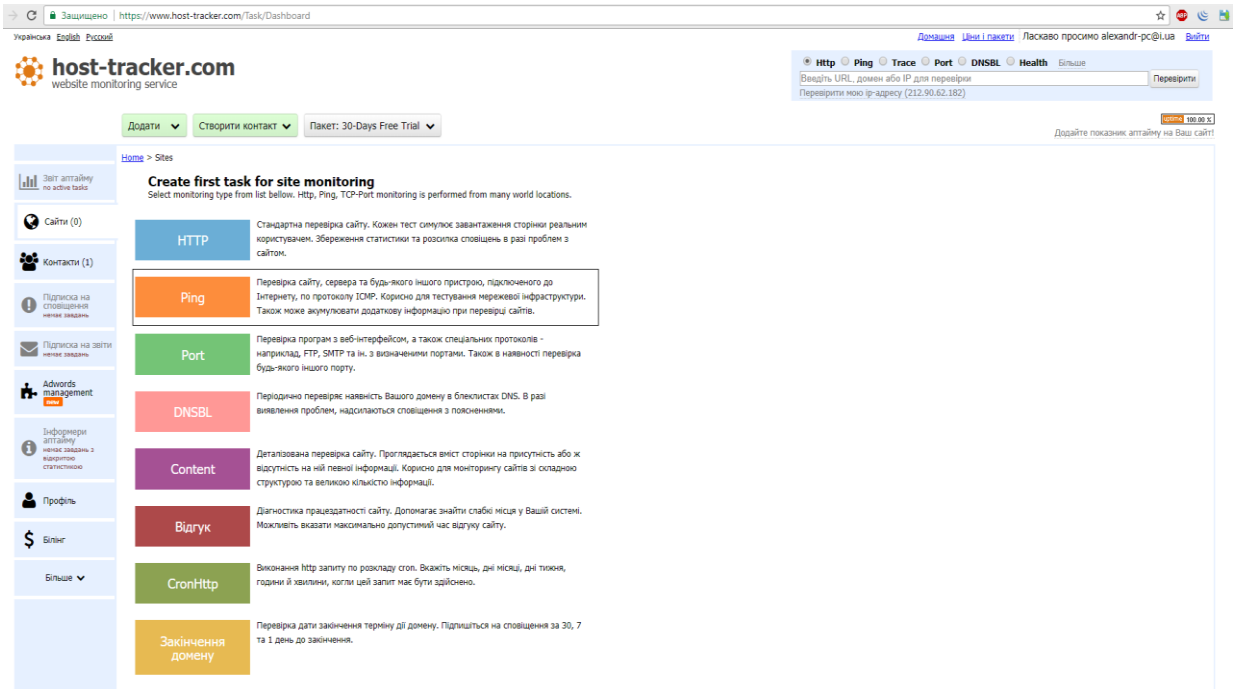


Рисунок 1.5 – Зовнішній вигляд Host tracker

Основні можливості системи Host-tracker.com:

– моніторинг довільної кількості ресурсів;

– розподілений моніторинг;

– період моніторингу кожні 1/5/15/30/60 хвилин;

– можливість моніторингу роботи CGI скриптів;

– підтримуються HTTP методи HEAD / POST / GET;

– можливість задавати передані CGI скрипту параметри;

– контроль наявності потрібних ключових слів на сторінці;

– можливість встановити довільну кількість адрес для повідомлення про збої сервера;

– зберігання статистики та подальше формування звітів;

– можливість «відкрити» звіти по одному або декілька ресурсів для вільного доступу;

– зберігання звітів без обмеження в часі;

– можливість налаштування автоматичного відсилання обраних звітів щоденних, щотижневих, щомісячних, квартальних, річних на вказані електронні адреси;

– прив'язка до часової зони;

– деталізація звітів з точністю до днів;

– можливість зберігання протоколів перевірки;

– можливість відсилання повідомлень про збої як на електронну адресу, так і на мобільний телефону, за допомогою SMS повідомлень;

– моментальна перевірка доступності ресурсу.

Функціонал системи Monitis.com (рис. 1.6) включає в себе такі можливості моніторингу [5]:

– відстеження наявності сайту і продуктивності в в більшості країн;

– перевірка працездатності з використанням HTTP, HTTPS, PING, DNS;

– перевірка працездатності баз даних (MySQL);

– e-mail перевірки з використанням протоколів SMTP, POP3, IMAP;

– перевірка IP з використанням протоколів TCP,UDP, SSH і IMCP;

– VoIP перевірки з використанням SIP протоколу;

– перевірка змісту сторінок;

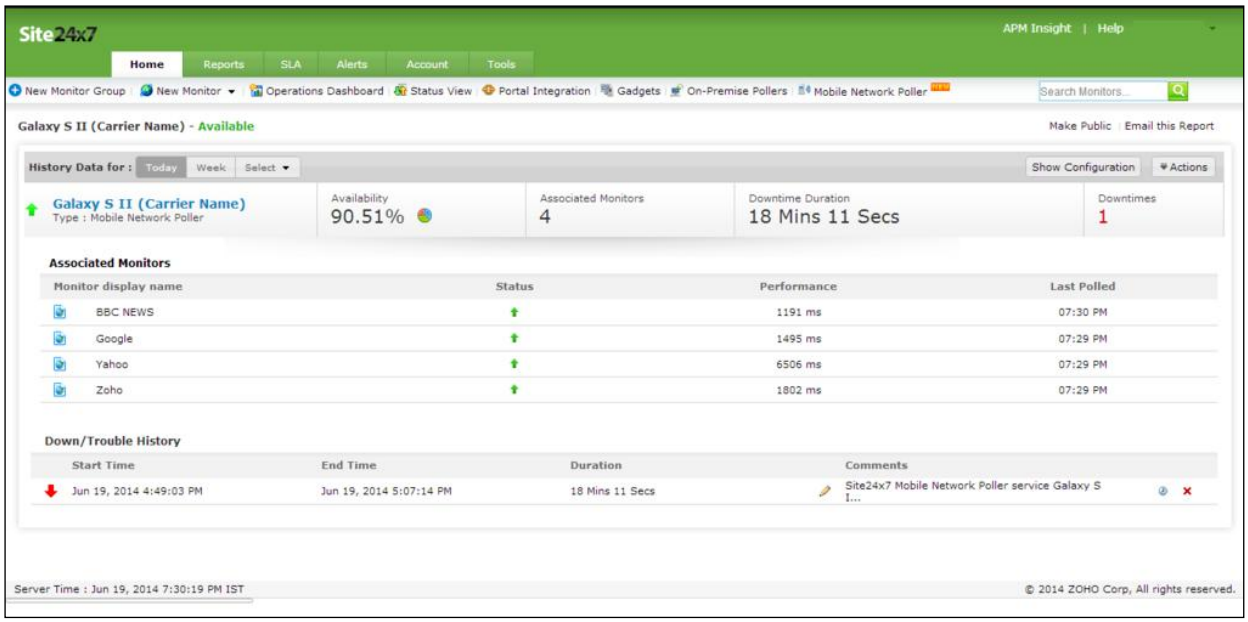
– миттєве сповіщення про відмову на електрону пошту, IM, SMS;

– SLA звітність - докладні звіти.

Також є дуже зручна функція, створити свій тарифний план, увімкнути тільки ті функції, які потрібні, що сприяє зниженню ціни.

Рисунок 1.6 – Зовнішній вигляд Monitis

Site24x7.com (рис. 1.7) надасть точні відомості про продуктивність вашої сторінки та про те, як користувачі ведуть себе під час перегляду сторінок. Інструмент дозволяє контролювати тільки один сайт. Автоматичні перевірки працездатності виконуються кожної години. Крім того, інструмент запускає моніторинг сервера, мереж та користувачів [6].



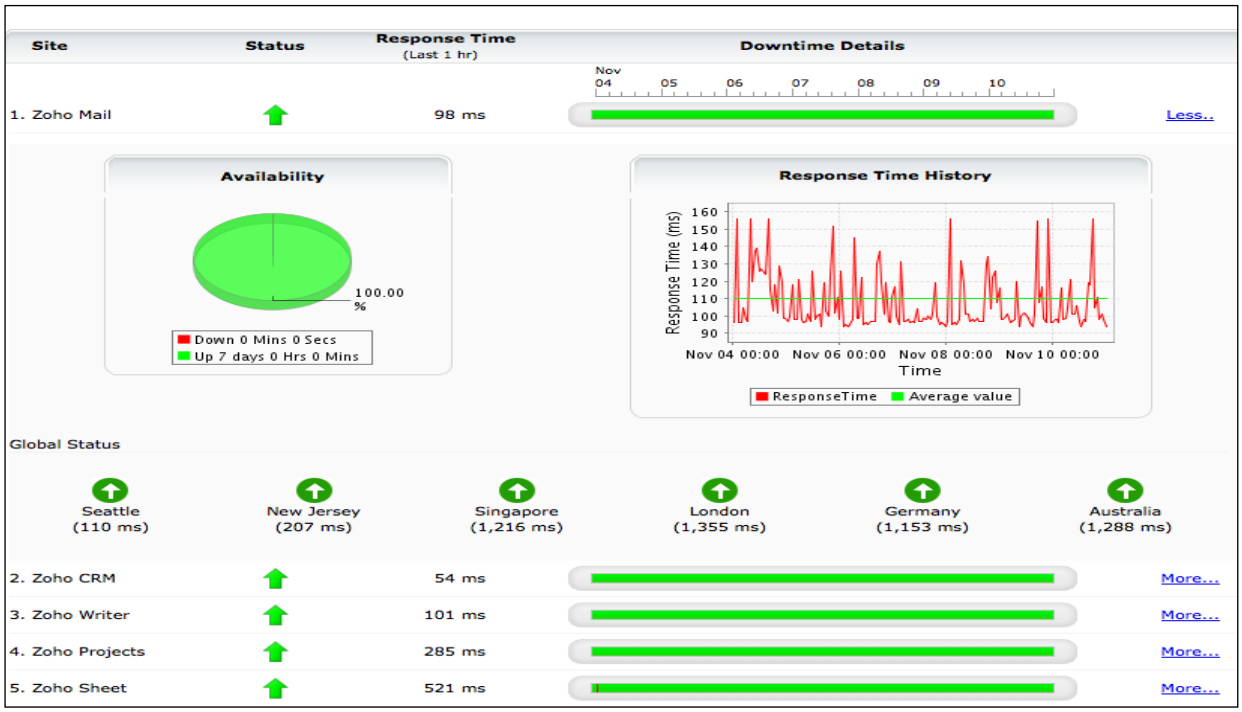


Рисунок 1.7 – Зовнішній вигляд Site24x7

Функціонал Site24x7:

– безперервний моніторинг 24 · 7 · 365;

– підтримка більшості протоколів - HTTP, HTTPS, FTP, SSH, SMTP, DNS, POP3, IMAP, MySQL, а також інших протоколів, що використовують TCP / IP;

– повідомлення про неполадки за допомогою e-mail і SMS;

– моніторинг з декількох точок світу за допомогою глобальної мережі моніторингу. контроль здійснюється з кількох точок, надається інформація про працездатність сайту і доступності користувача по всьому світу;

– можливість виведення звіту про доступність серверів;

– можливість відправки провайдеру запиту на перевантаження системи;

– можливість автоматичної відправки повідомлень в технічну підтримку хостингу;

– звіти про доступність і продуктивність;

– використовується сучасні технології взаємних перевірок для запобігання помилкових тривог;

– планування профілактичних робіт;

– безкоштовне перенесення даних;

– простота у використанні - потрібно тільки браузер.

Головною особливістю сервісу UptimeRobot.com є повна безкоштовність (рис. 1.7). Є можливість як звичайної перевірки сторінки, так і перевірки наявності або відсутності заданої фрази на сторінці, а також можливість простого звернення на сервер і перевірки сервісу на обраному порту. Інтервал перевірок можна налаштувати в діапазоні від 5 до 120 хвилин [6].

Отримувати повідомлення про недоступність сайтів можливе декількома способами: на email, через sms, push повідомлення на телефон і іншими (можна використовувати кілька одночасно).

У безкоштовної версії є до 50 перевірок чотирьох різних типів, наведених нижче:

– HTTP (s)-запит з перевіркою коду відповіді сервера;

– HTTP (s)-запит з пошуком заданого ключового слова у відповіді;

– PING ресурсу;

– TCP-SYN перевірка на доступність TCP-порту.

Серед недоліків цього сервісу можна виділити недостатність regexp-перевірки та перевірок UDP-портів.

Рисунок 1.8 – Зовнішній вигляд UptimeRobot

Зробимо порівняльну характеристику цих систем в табл. 1.1, де 1 – це мінімальний показник, а 10 – найкращий.

Таблиця 1.1 – Порівняльна характеристика систем моніторингу сайтів

	Функ-ціонал	Зручність користування	Наявність тестового періоду	Наявність API
Ping-admin	9	4	Так (7 днів)	Так
Host-tracker	10	8	Так (30 днів)	Так (обмежений функціонал
Monitis	9	7	Так (15 днів)	Так
Site24x7	8	9	Так (30 днів)	Так
UptimeRobot	8	10	Є безкоштовний пакет, але з обмеженими функціями	Так

1.4 Формування вимог до захисту інформації та працездатності сайтів користувачів

Заходи, що розроблюються та розроблене програмне забезпечення відноситься до сервісів, які дають змогу підвищити безпеку інформації користувача та перевірити доступність сайту. Користування сервісом не повинно обмежуватися геолокацією користувача і різновидами доменами сайтів.

Сервіс має працювати таким чином: користувач повинен зареєструватися в сервісі, підтвердити свою пошту або телефон вказаний при реєстрації (для того щоб запобігти використання «фейкових» облікових записів). Після підтвердження необхідно успішно пройти процедуру авторизації, далі користувач потрапляє в панель адміністрування, де є змога додати один або декілька доменів для відстеження їх працездатності ресурсом (в безкоштовному тарифному плані, буде доступне додавання тільки одного домену). Також, в кожному доданому домені повинні бути: налаштування сповіщень, а саме: по e-mail або по sms, статистика непрацездатності ресурсу в відсотках і т.д.

Розроблюваний сервіс буде виконувати наступні функції:

- реєстрація користувача;
- авторизація користувача;
- редагування даних користувача;
- додавання доменів користувача для моніторингу;
- налаштування кожного домену;
- відображення статистики працездатності;
- сповіщення користувача про непрацездатний сайт.

2 РОЗРОБЛЕННЯ КОМПЛЕКСНОГО ЗАХИСТУ ІНФОРМАЦІЇ

На даний момент в світі існує безліч різних технологій, для того щоб створити серверну частину веб-додатку, але в кожної є свої мінуси и плюси.

Виділимо важливі критерії при виборі технологій:

– розмір і тип проекту;

– важкість проекту;

– швидкість розробки;

– доступні інструменти розробки;

– наявність готових рішень;

– гнучкість рішення;

– наявність великої спільноти;

– відмово стійкість рішення;

– тренд його розвитку;

– наявність детальної документації;

– вимоги до навантажень;

– вимоги до безпеки;

– кросплатформеність.

2.1 Вибір локального серверу безпекового рівня

Локальний сервер – програма, яка створює на персональному комп’ютері середовище повноцінного веб хостингу. Тобто на вашому домашньому комп'ютері створюється міні-хостинг, на якому будуть успішно функціонувати всі серверні движки, скрипти, CMS (WP, Joomla та інші). Вам навіть не потрібно буде підключатися до Інтернету – у вас буде свій міні-інтернет з одним або декількома сайтами. Так що за допомогою локального сервера можна успішно займатися веб-розробкою і потім переносити свої скрипти на реальний веб

хостинг в інтернеті. Велика частина сайтів сучасного інтернету динамічні і працюють в основному на PHP. PHP також часто використовується для навчання програмуванню. Але браузери розуміють тільки HTML і CSS, а PHP - немає. Тому що PHP це серверний мова програмування і сервер якраз перетворює і обробляє PHP-код (або результат його виконання) в вид, зрозумілий браузеру. І такі обробники стоять на кожному сервері / хостингу в інтернеті (без них нікуди), але не на вашому домашньому комп'ютері [8].

Потреба в локальних серверах постійно росла при розробці динамічних сайтів на PHP, Perl та іншими мовами програмування. Спочатку це було обумовлено поганим і дорогим Інтернетом, потім люди зрозуміли необхідність тестування скриптів в спеціальному середовищі, та й взагалі виріс аматорський і професійний інтерес до програмування.

Для повноцінної імітації веб-сервера і вирішення всіх вищеозначених завдань і був створений локальний сервер.

Зазвичай ці проблеми вирішувалися, та й до сих пір вирішуються, засобами FTP-клієнта. Файл завантажується з веб-хостингу, редагується та завантажується назад.

Це як мінімум незручно – треба бути постійно підключеним до стабільного Інтернету, потрібно чекати поки завантажиться назад (файл адже може бути великим, їх може бути кілька), потрібно постійно редагувати файли коли «щось йде не так». Локальний сервер, налаштований ідентично якби він був би на хостингу веб-сервера, спрощує цей процес. На веб-хостинг завантажується тільки підсумкова, фінальна версія веб-додатку.

Локальний сервер Denwer. Денвер (Denwer) – один з найбільш популярних локальних серверів [9]. Розшифровується як «джентельменський набір веб-розробника» – набір дистрибутивів і програмного забеспечення для веб-розробки на локальному ПК. Денвер є одним з найстаріших локальних

серверів широко відомих в рунеті, одним з основних переваг якого в момент появи була можливість роботи з USB-флеш-накопичувача.

Зараз вже є і інші локальні сервера, які не поступаються за функціоналом.

Локальний сервер Winginx – локальний веб-сервер для розробки на мовах програмування PHP і навіть Node.js. У Winginx вбудовані БД – MongoDB, Redis, memcached, MySQL. Особливістю Winginx є вбудований сервер nginx, а не Apache як на інших локальних серверах [10].

Відмінність комплексу Winginx:

– швидкий і простий запуск локального сервера nginx на ОС Windows;

– зручна локальна розробка сайтів і сервісів на Node.js і PHP;

– мультипроектна система для розробки, що має універсальні і гнучкі налаштування, легко оновлюються компоненти;

– середовище для ведення проекту: можна створювати завдання і враховувати час на їх виконання;

– середовище для локального тестування і запуску веб-додатків, сайтів і браузерних сервісів. Зовнішній вигляд серверу представлено на рисунку 2.1.

– Особливості Winginx в порівнянні з іншими локальними серверами: єдиний центр управління сервером і оновленнями компонентів, одночасна мультипроектного робота з декількома сайтами (в т.ч. використовуючи різні версії PHP), управління завданнями і проектами, облік часу на виконання завдань, завантаження безкоштовних CMS з магазина Winginx і їх установка «в 1 клік». Winginx має вбудований серверний менеджер і центр поновлення. Winginx не навантажує локальний комп'ютер, непомітно працюючи в треї.

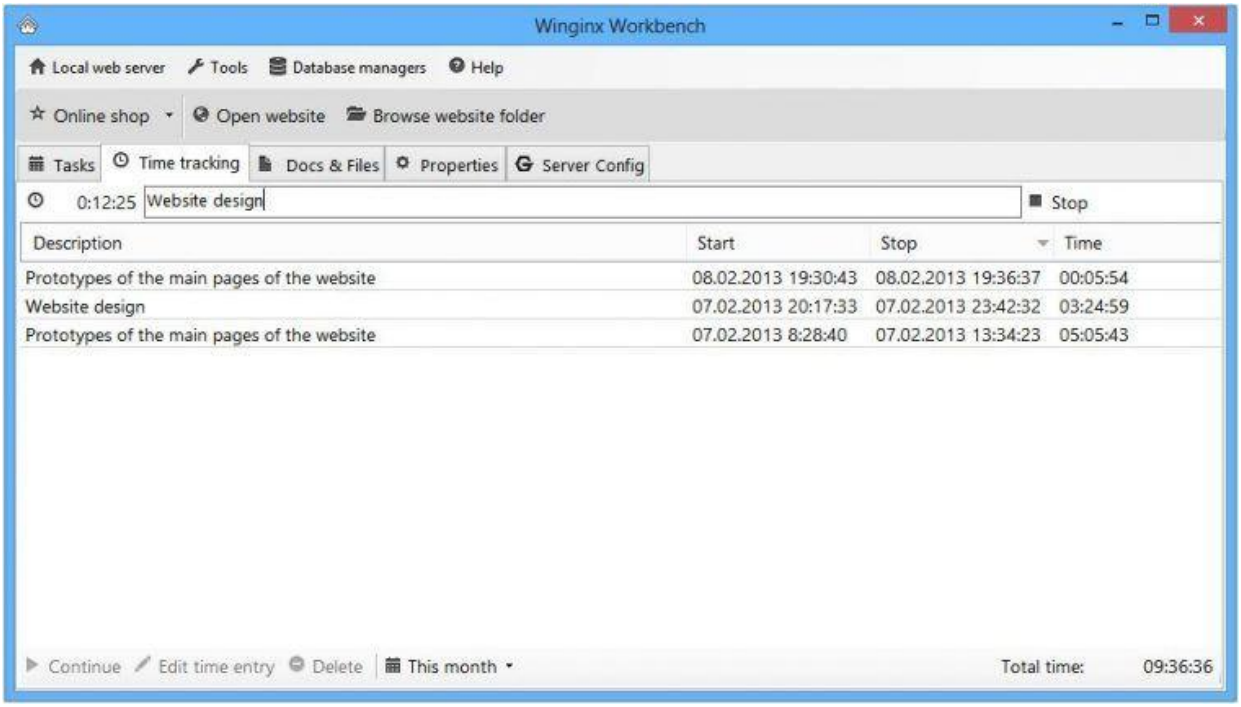


Рисунок 2.1 – Зовнішній вигляд Winginx

Локальний сервер Open Server – програмне середовище, що створює портативну локальну серверну платформу [11]. Open Server створений спеціально для веб-розробників і враховує всі отримані рекомендації і побажання по роботі середовища. Завдяки цьому, Open Server широко використовується в різних країнах світу для тестування, налагодження і розробки з нуля різних веб-проектів і створення повнофункціональних веб-серверів в локальних корпоративних і домашніх мережах. Зовнішній вигляд зображений на рисунку 2.2.

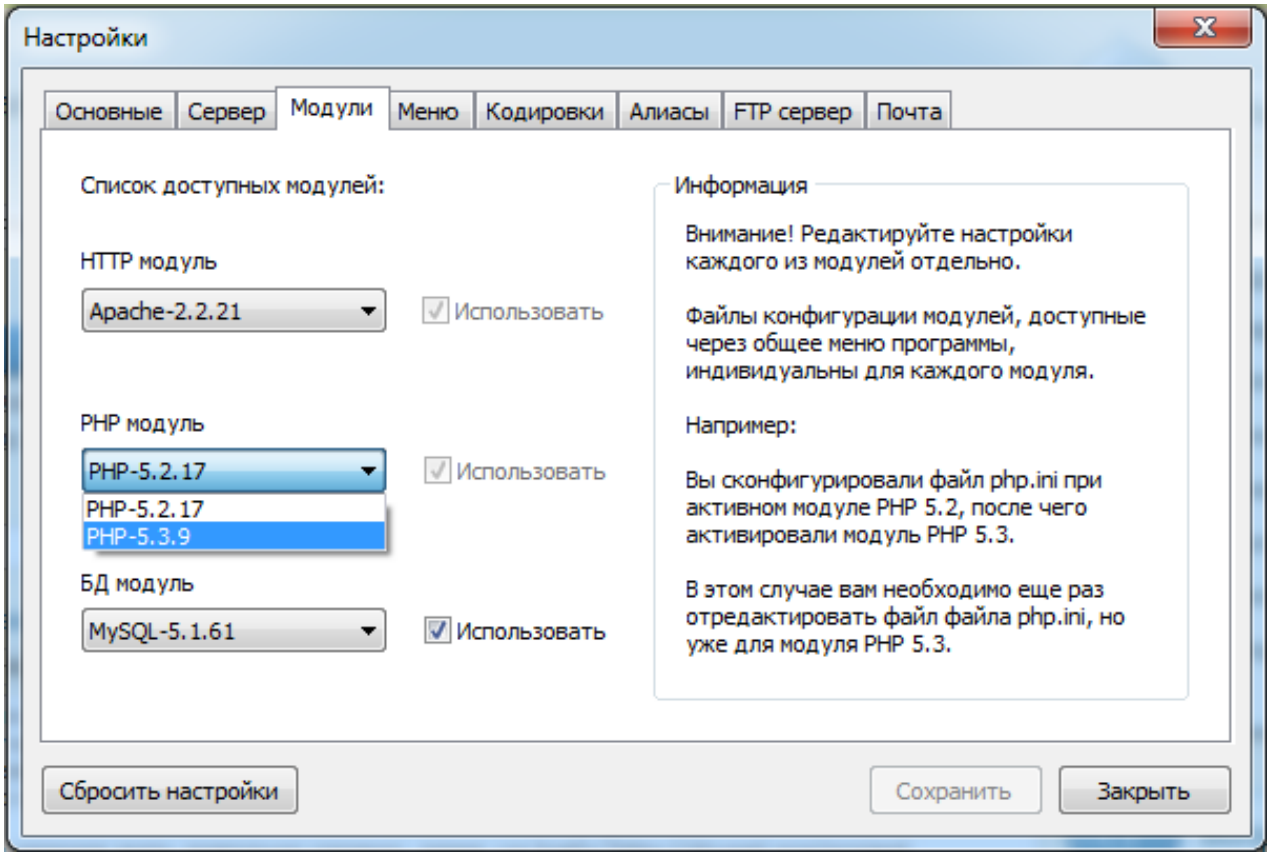


Рисунок 2.2 – Зовнішній вигляд OpenServer

Особливості комплексу OpenServer:

– не вимагає установки (портативність);

– можливість роботи з USB накопичувача;

– одночасна робота з Denwer, Xampp і т.д .;

– робота на локальному / мережевому / зовнішньому IP адресу;

– підтримка SSL без всяких додаткових налаштувань;

– створення домену шляхом створення звичайної папки;

– підтримка кириличних доменів;

– підтримка аліасів (доменних покажчиків);

– захист сервера від зовнішнього доступу;

– punycode конвертер доменних імен;

– пакет з понад 40 портативних програм;

– планувальник завдань (cron);

– створення локального піддомена без втрати видимості основного домену в мережі Інтернет;

В ролі локального серверу був вибраний саме OpenServer.

2.2 Вибір технології розробки

Сучасні технології створення і підтримки веб-сайтів орієнтовані на платформи, що дозволяють ефективно управляти інформаційним наповнюванням і даними, які надходять від відвідувачів сайту. Як правило, такі рішення базуються на серверних технологіях типу ASP, ASP.NET, JSP, PHP або використовують готові потужні засоби (наприклад, фреймворки) для створення корпоративних сайтів, орієнтованих на впровадження зазначених технологій.

Створення веб-сторінок фрагментами серверного коду є технологією ASP, ASP.NET (Active Server Pages). Це розроблення Microsoft комерційно доступна технологія, за допомогою якої веб-майстер може самостійно формувати динамічно поновлювані веб-сторінки. Характерною особливістю такої технології є можливість відділення функціональної частини розробки від процесів створення дизайну. ASP-сторінки можуть містити HTML-текст, змішаний зі сценаріями як JavaScript і VBScript. В процесі обробки запиту нової сторінки його виконує сервер і динамічно генерує браузеру потік HTML-тексту відображається на екрані монітора. ASP-технологія Microsoft отримала подальший розвиток в технологіях JSP, PHP та ін.

Технологія JSP (Java Server Pages) – це технологія створення серверних сторінок Java. Специфікація JSP є розширенням Java Servlet API для генерації динамічних веб-сторінок на веб-сервері. Така крос-платформа є альтернативою технології ASP корпорації Microsoft.

Специфікація Sun під назвою JSF (Java Server Faces) реалізує технологію JSP, що описує правила створення веб-додатків зі зручним для користувача інтерфейсом і орієнтована на розробку серверних компонентів створення інтерфейсу.

Однією з перших технологій створення веб-додатків, які виконують-ються сервером, була Common Gateway Interface (CGI) технологія.

Вона дозволила розробку і виконання серверних додатків, обіг яких відбувається за допомогою зазначеного в URL імені (і параметрів). Залежно від обраного протоколу вхідної інформації таких веб-додатків вважають безпосередньо код HTTP-заголовка або запит пошукової системи. CGI-додатки - це консольні додатки, які генерують HTML-код, переданий браузера. Подібні додатки можуть являти собою код на скриптових мовах, який інтерпретується на сервері. Крім того, CGI- застосування представляють робочий файл, який можна створити за допомогою будь-якого засобу розробки, генерує консольні додатки для операційної системи, під управлінням якої працює веб-сервер.

Серед інших популярних технологій, що реалізують створення веб сторінок з фрагментами коду, що виконується на сервері, виділимо некомерційну, вільно поширювану технологію PHP (Personal HomePages). Ця технологія заснована на використанні CGI-додатків, інтерпретують впроваджений в HTML-сторінку код скриптовій мовою. Головною особливістю мови PHP є її практичність. PHP надає програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Вона відрізняється винятковою гнучкістю до потреб розробника. хоча PHP традиційно рекомендують використовувати в поєднанні з HTML-кодом, проте PHP з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови програмування.

Результати порівняння технологій розроблення і впровадження веб-ресурсів зведені в табл. 2.1. Розглянуті технології забезпечують одночасну

функціональність, ефективний супровід процесів створення сайтів і їх наповнення інформаційними ресурсами.

Таблиця 2.1 – Порівняння сучасних технологій розробки сайту

	PHP	JSP	ASP.NET
Кросплатформеність	+	+	-
Продуктивність	+/-	+/-	+
Простота використання	+	+/-	+/-
Доступні програмні бібліотеки	+	+	+
Поділ дизайну і логіки	+	+/-	+
Безкоштовність	+	+	-

Отже, в цього дипломному проекті буде використовуватися мова програмування PHP, однак робити веб-сервіс на чистому PHP, дуже довго, и не зручно, тому треба вибрати сучасний фреймворк, який би ідеально підійшов для цього. Тому, порівняємо найпопулярніші PHP фреймворки.

Фреймворк – це програмне забезпечення, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. На відміну від бібліотеки функцій, фреймворк накладає обмеження на структуру і логіку програмного продукту [12].

Веб-фреймворк призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів і ресурсів. Веб-фреймворки містять логіку обробки HTTP-запитів, спрощують доступ до баз даних та інше. Фреймворк є надбудовою над мовою програмування і дозволяє конструювати програми з сторонніх модулів, легко їх розширювати і модифікувати.

З одного боку, фреймворк вводить обмеження на структуру файлів, стиль оформлення коду, правила з розділення логіки і має функції, які можуть зовсім не використовуватися в готовому рішенні. З іншого боку, фреймворки

скорочують час проектування і розробки додатків, виключають дублювання коду, а також спрощують супровід проектів.

Крім того, фреймворки супроводжуються документацією, навколо фреймворків утворюються співтовариства, що містять приклади і базу знань.

При виборі фреймворку для розробки необхідно враховувати патерни, які використовує фреймворк, наявність документації та розмір спільноти навколо фреймворка, а також можливість розширення функціоналу проектованої системи за рахунок власних і сторонніх розширень.

Для порівняння були обрані PHP-фреймворки Yii 2 як самий популярний фреймворк в країнах СНД і Laravel - як найпопулярніший фреймворк в світі. У таблиці 2.2 представлено порівняння PHP- фреймворків Yii 2 і Laravel.

Завдяки зручності, розвиненій спільноті, а також високої популярності для розробки веб-додатку був обраний PHP-фреймворк Yii 2.

Таблиц 2.2 – Порівняння PHP-фреймворків Yii 2 і Laravel

Критерії	Фреймворки	
	Yii 2	Laravel
Вимоги	PHP 5.4.0 і вище	PHP 5.5.9 і вище. Розширення PHP: OpenSSL; PDO; Mbstring; Tokenizer
Патерн	MVC	MVC
Розширення	Підтримуються (composer)	Підтримуються (composer)
Міграції	Підтримуються	– Підтримуються – Інструмент наповнювання

		даних
ORM (об'єктно реляційне відображення)	Active Record	Eloquent ORM (Active Record)
Валідація форм	Підтримується	Підтримується
Локалізація	Підтримується	Підтримується
Англомовне співтовариство	Документація, форуми, блоги	Документація, відео-уроки (Laracast), форуми, блоги
Україномовне співтовариство	Документація, форуми, блоги	Форуми, блоги

Як і більшість інших PHP-фреймворків, Yii – це MVC-фреймворк. Перевага Yii над іншими фреймворками полягає в ефективності, широких можливостях і якісної документації. Yii спочатку спроектований дуже ретельно для відповідності всім вимогам при розробці серйозних веб-додатків. Yii не є ні побічним продуктом будь-якого проекту, ні складанням сторонніх рішень. Він є результатом великого досвіду авторів у розробці веб-додатків, а також їх досліджень найбільш популярних веб-фреймворків і додатків. Статичну архітектуру фреймворку можна розглянути на рисунку 2.3.

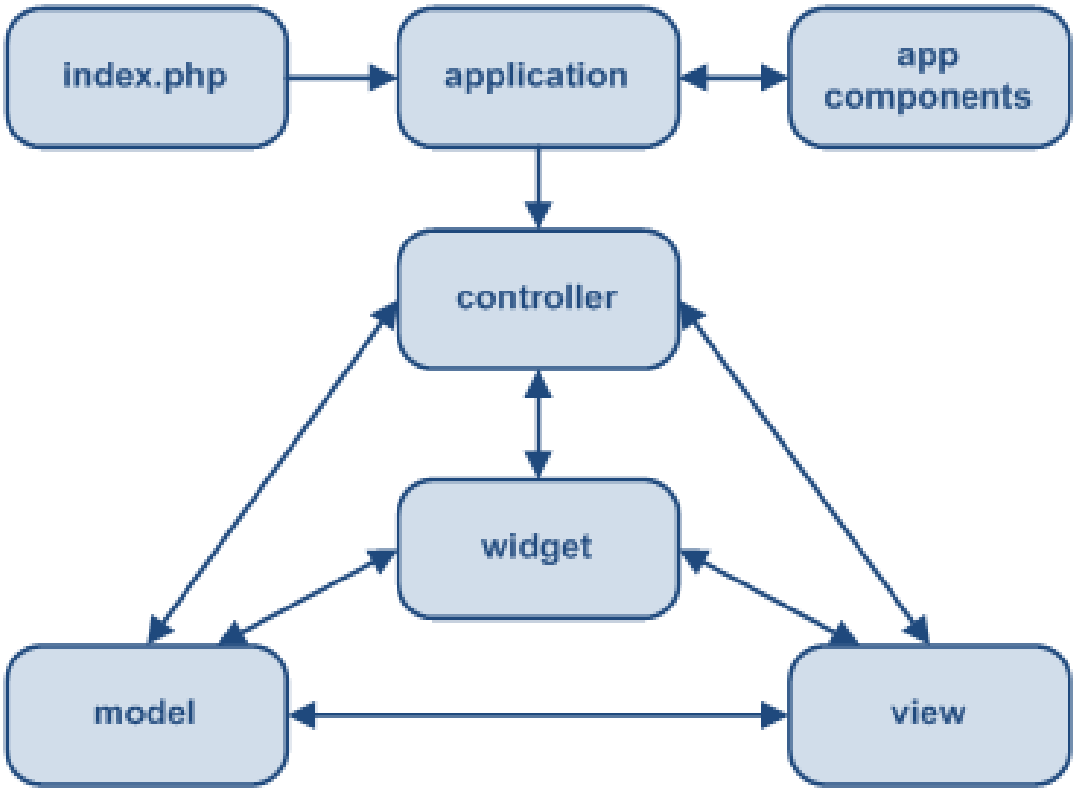


Рисунок 2.3 – Статична архітектура додатку в фреймворкі Yii2

Особливості:

– Висока продуктивність;

– Інтерфейси DAO і ActiveRecord для роботи з базами даних (PDO);

– Підтримка інтернаціоналізації;

– Кешування сторінок та окремих фрагментів;

– Перехоплення і обробка помилок;

– Введення і валідація форм;

– Аутентифікація і авторизація;

– Використання AJAX і інтеграція з jQuery;

– Генерація базового PHP-коду для CRUD-операцій (скаффолдінг);

– Підтримка тем оформлення для їх легкої зміни;

– Можливість підключення сторонніх бібліотек;

– Міграції бази даних;

– Автоматичне тестування;

– Підтримка REST.

2.3 Вибір засобу реалізації бази даних з метою забезпечення конфіденційності інформації користувача

Робота з базами даних є однією з головних складових процесу програмування сайту динамічного типу. Бази даних для сайтів використовують з метою зберігання різнопланової інформації. База даних представляє з себе певний набір взаємопов'язаних таблиць. Розміри таблиць в базах різні, а їх кількість – довільна. Бази даних на сервері накопичують необхідну для роботи сайту інформацію статистичного характеру.

До найпопулярніших мережевих баз даних відносять SQL, MySQL, Oracle Database та інші. Вибір потрібної системи управління базою даних (СКБД) обумовлюється вимогами до інформаційних характеристик і функціональних можливостей системи.

Однією з найпоширеніших систем управління базами даних в наш час вважається MySQL, яка є альтернативою комерційним системам [13].

MySQL — вільна система керування реляційними базами даних. MySQL був розроблений компанією «TcX» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має підтримку з боку різноманітних мов програмування[16]. MySQL має подвійне ліцензування. MySQL може розповсюджуватися відповідно до

умов ліцензії GPL. Але за умовами GPL, якщо якась програма використовує бібліотеки MySQL, то вона теж повинна розповсюджуватися за ліцензією GPL. Проте це може розходитися з планами розробників, які не бажають відкривати сирцеві тексти своїх програм. Для таких випадків передбачена комерційна ліцензія компанії Oracle, яка також забезпечує якісну сервісну підтримку. В разі використання та розповсюдження програмного забезпечення з іншими вільними ліцензіями, такими як BSD, Apache, MIT та інші, MySQL дозволяє використання бібліотек MySQL за ліцензією GP [14].

Об'єктно-реляційна система управління базами даних компанії Oracle (Oracle Database) орієнтована під операційні системи Windows, Unix, Linux і MacOS. Oracle Database, на відміну від MySQL, має більш широку сферу застосування. СКБД Oracle широко відома як в нашій країні, так і в світі. На ній базується безліч сучасних інформаційних систем [15].

SQL (Structured Query Language – мова структурованих запитів) – декларативна мова програмування засобів взаємодії користувача з базами даних, реалізує процеси формування запитів, оновлення і керування базами даних, створення схеми бази даних і її модифікації, систему контролю доступу до інформаційних ресурсів. SQL може формувати інтерактивні запити або, будучи вбудованою в додатки, виступати в якості інструкцій для керування даними. Крім того, стандарт SQL містить функції визначення процесів зміни, перевірки і захисту даних [16]. У табл. 2.3 представлено порівняння сучасних СКБД.

Проаналізувавши базові характеристики розглянутих СКБД, можна зробити висновок про зручність використання MySQL для створення динамічних веб-сторінок, оскільки сервер MySQL по всім параметрам виявився найкращий, тому в цьому дипломному проекті будемо використовувати саме його.

Таблиця 2.3 – Порівняльна характеристика СКБД

	SQL	Oracle	MySQL
Надійність	+	+	+
Швидкість	–	+	+
Простота	–	–	+
Зручність користування	+/–	+	+
Безкоштовність	+/–	–	+

2.4 Вибір середовища програмування

JetBrains PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA [17]. Зовнішній вигляд зображений на рисунку 2.4.

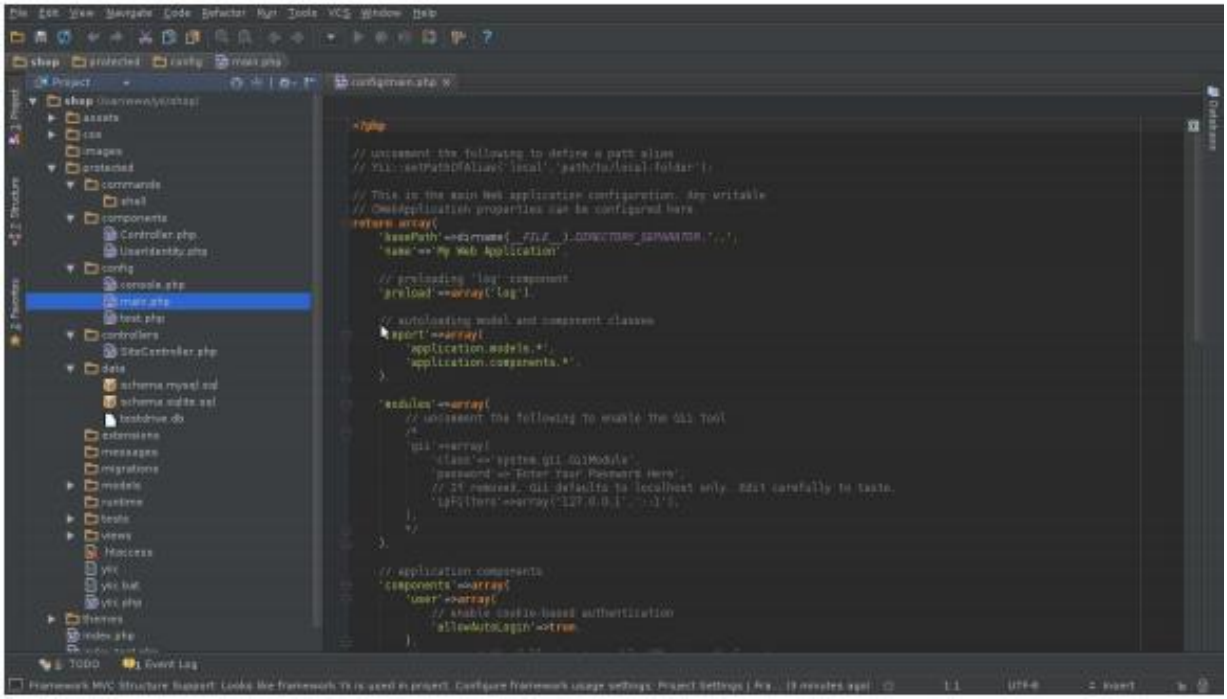


Рисунок 2.4 – Зовнішній вигляд JetBrains PhpStorm

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3 – 7 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів. PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плаґінів, розроблених для платформи IntelliJ, або написавши власні плаґіни. Вся функціональність WebStorm включена в PhpStorm.

Особливості. Основна задача IDE — спростити розробку програми з допомогою PHP на різних платформах. Тому серед можливостей, які властиві будь-якому середовищу розробки, є і специфічні.

Редактор коду PHP. PhpStorm надає багатий і інтелектуальний редактор коду для PHP з підсвічуванням коду, розширеною конфігурацією форматування коду, перевіркою на наявність помилок на льоту і розумним авто доповненням. Можливості редактору коду:

– підтримка PHP 5.3, 5.4 та 5.5, включаючи генератори, співпрограми, простори імен, замикання, типажі, синтаксис коротких масивів, доступ до члена класу при інстанціюванні, розіменування масиву при виклику функції, бінарні літерали, вираження в статичних виклики тощо. PhpStorm може використовуватися як для сучасних, так і для традиційних проектів на PHP;

– автодоповнення коду фіналізують класи, методи, імена змінних, ключові слова PHP, а також широко використовувані імена полів і змінних залежно від їхнього типу;

– підтримка стандартів оформлення коду (PSR1/PSR2, Drupal, Symfony2, Zend);

– підтримка PHPDoc. PhpStorm надає відповідне автодоповнення коду, засноване на анотаціях @property, @method і @var.

– детектор дубльованого коду;

– PHP Code Sniffer (phpcs), котрий перевіряє код на льоту;

– рефакторинги (перейменування, введення змінної/константи/поля, вбудовування змінної);

– підтримка редагування шаблонів Smarty (підсвічування синтаксичних помилок, автодоповнення функцій і атрибутів Smarty, автоматична вставка парних дужок, лапок і закриваючих тегів тощо);

– MVC подання для фреймворків Symfony2 і Yii;

– розпізнавання коду, запакованого в PHAR-архіви.

Середовище розробки в інструменті JetBrains PHP Shorm підтримує дуже багато різних можливостей. Серед них:

– підтримка SQL і баз даних (Рефакторинг схеми бази даних, генерація скриптів міграції схеми, експорт результатів виконання запиту у файл або буфер обміну, редагування збережених процедур і багато іншого;

– віддалене розгортання додатків і автоматична синхронізація з використанням FTP , SFTP , FTPS та ін протоколів;

– інтеграція з системами управління версіями (Git - включаючи спеціальний функціонал для роботи з GitHub , Subversion , Mercurial , Perforce , CVS , TFS), що дозволяє робити багато дій, наприклад commit, merge, diff та інші, прямо з PhpStorm;

– локальна історія (Local History) (локально відстежує будь-які зміни в коді);

– PHP UML (Діаграми класів UML для PHP коду з рефакторингом, що викликаються прямо з діаграми);

– підтримка Phing (надає автодоповнення, перевірку стандартних тегів, властивостей, імен цілей, значень атрибутів шляху в компонувальних файлах (build files);

– інтеграція з системами відстеження помилок;

– підтримка Vagrant, SSH консолі і віддалених інструментів;

2.5 Проектування бази даних для забезпечення безпеки інформації користувача

Концептуальне проектування. Мета етапу концептуального проектування – створення концептуальної моделі даних виходячи з уявлень користувачів про предметну область. Для її досягнення виконується ряд послідовних процедур:

1) Визначення сутностей та їх документування. Для ідентифікації сутностей визначаються об'єкти, які існують незалежно від інших. Такі об'єкти є сутностями. Кожній сутності присвоюється осмислене ім'я зрозуміле користувачам. Імена та опису сутностей заносяться в словник даних. Якщо можливо, то встановлюється очікувана кількість примірників кожної сутності.

2) Визначення зв'язків між сутностями і їх документування. Визначаються тільки ті зв'язки між сутностями, які необхідні для задоволення вимог до проекту бази даних. Встановлюється тип кожної з них. Виявляється клас приналежності сутностей. Зв'язках присвоюються осмислені імена, виражені дієсловами. Розгорнутий опис кожної зв'язку із зазначенням її типу і класу приналежності сутностей, що беруть участь в зв'язку, заноситься в словник даних.

3) Створення ER-моделі предметної області. Для уявлення сутності і зв'язків між ними використовуються ER-діаграми. На їх основі створює-ся єдиний наочний образ моделюється предметної області - ER-модель предметної області.

4) Визначення атрибутів і їх документування. Виявляються всі атрибути, що описують сутність створеної ER-моделі. Кожному атрибуту при-привласнювати осмислене ім'я, зрозуміле користувачам. Про кожному атрибуті в словник даних збожеволіють такі відомості:

– ім'я атрибута і його опис;

– тип і розмірність значень;

– значення, яке приймається для атрибута за замовчуванням (якщо є);

– чи може атрибут мати Null-значення;

– чи є атрибут складовим, і якщо це так, то з яких простих атрибутів він складається;

– чи є атрибут розрахунковим, і якщо це так, то як обчислюються його значення.

5) Визначення значень атрибутів і їх документування. Для кожного атрибута сутності, що бере участь в ER-моделі, визначається набір допустимих значень і йому присвоюється ім'я.

6) Визначення первинних ключів для сутностей та їх документування. На цьому кроці керуються визначенням первинного ключа - як атрибуту або набору атрибутів сутності, що дозволяє унікальним чином ідентифікувати її екземпляри. Відомості про первинні ключи поміщаються в словник даних.

7) Обговорення концептуальної моделі даних з кінцевими користувачами. Концептуальна модель даних представляється ER-моделлю з супровідною документацією, що містить опис розробленої моделі даних. Якщо будуть виявлені невідповідності предметної області, то в модель вносяться зміни до тих пір, поки користувачі не підтвердять, що запропонована ним модель адекватно відображає їх особисті уявлення.

Логічне проектування БД. Логічне проектування баз даних - це процес створення моделі, що використовується в підприємстві інформації, на основі вибраної моделі організації даних, але без урахування типу цільової СКБД та

інших фізичних аспектів реалізації. Мета полягає в створенні логічної моделі даних для досліджуваної частини підприємства. Логічна модель даних враховує особливості вибраної моделі організації даних у цільовій СКБД (наприклад, реляційна модель).

Якщо концептуальна модель даних не залежить від будь-яких фізичних аспектів реалізації, то логічна модель даних створюється на основі вибраної моделі організації даних цільової СКБД. На цьому етапі вже повинно бути відомо, яка СКБД буде використовуватися як цільова - реляційна, мережева, ієрархічна або об'єктно-орієнтована. Однак на цьому етапі ігноруються всі інші характеристики вибраної СУБД, наприклад, будь-які особливості фізичної організації її структур зберігання даних та побудови індексів.

В процесі розробки логічна модель даних постійно тестується і перевіряється на відповідність вимогам користувачів.

Створена логічна модель даних є джерелом інформації для етапу фізичного проектування та забезпечує розробників фізичних баз даних засобами пошуку компромісів, необхідних для досягнення поставлених цілей, що дуже важливо для ефективного проектування.

Мета етапу логічного проектування – перетворення концептуальної моделі на основі вибраної моделі даних у логічну модель, не залежно від особливостей використовуваної в подальшому СУБД для фізичної реалізації баз даних. Для її досягнення виконуються наступні процедури:

– вибір моделі даних;

– визначення набору таблиць і їх документування;

– нормалізація таблиць;

– визначення вимог підтримки цілісності даних та їх документування.

Логічна модель даних також грає важливу роль на етапі експлуатації та супроводу вже готової системи. При правильно організованому супроводі підтримується в актуальному стані моделі даних, що дозволяє точно та наочно

представляти будь-які внесені в базу даних зміни, а також оцінити їх вплив на прикладні програми та використання даних, вже наявних в базі.

Опис сутностей:

1) Сутність «user» містить інформацію про користувачів.

2) Сутність «user_contact» містить додаткову інформацію користувачів

3) Сутність «site» містить інформацію про сайти.

4) Сутність «user_site» містить інформації які сайти належать користувачу

5) Сутність «plan» містить інформації про тарифи.

6) Сутність «user_purchase» містить данні про придбання тарифного плану користувачем.

7) Сутність «event» містить всі події які відбуваються с сайтами користувачів.

Фізичне проектування БД. Фізичне проектування баз даних – це процес підготовки описання реалізації баз даних на вторинних запам'ятовуючих пристроях. На цьому етапі розглядаються основні відносини, організація файлів та індексів, призначених для забезпечення ефективного доступу до даних, а також всі пов'язані з цим обмеження цілісності та засоби захисту.

При виконанні етапу, розробник приймає рішення про способи реалізації розроблюваної бази даних. Приступаючи до фізичного проектування баз даних, перш за все необхідно вибрати конкретну ціль-вую СКБД. Тому фізичне проектування нерозривно пов'язане з конкретною СКБД. Між логічним та фізичним проектуванням існує постійний зворотній зв'язок, так як рішення, прийняті на етапі фізичного проектування з метою підвищення продуктивності системи, здатні впливати на структуру логічної моделі даних.

Як правило, основним завданням фізичного проектування баз даних є опис способу фізичної реалізації логічного проекту бази даних. На рисунку 2.5 представлена фізична модель бази даних.

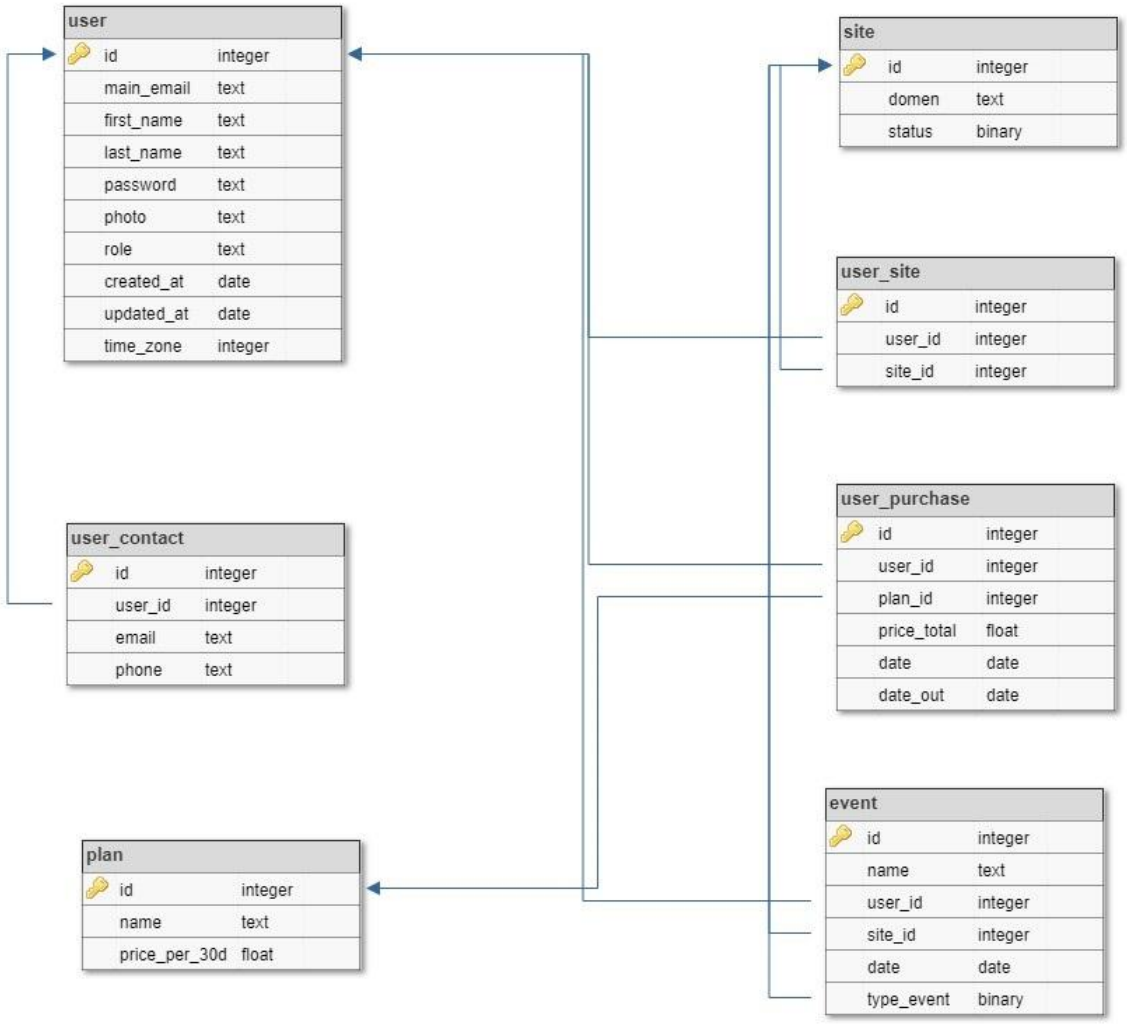


Рисунок 2.5 – Фізична модель бази даних

Результатом фізичного проектування логічної схеми вище на мові SQL являються наступні приклади скриптів:

Скрипт створення таблиці «user»:

```
CREATE TABLE user (
    id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
main_email VARCHAR(255) NOT NULL,
first_name VARCHAR(255) NOT NULL,
last_name VARCHAR(255) NOT NULL,
```

```
password VARCHAR(255) NOT NULL,
photo VARCHAR(255) NULL DEFAULT NULL,
role VARCHAR(255) NOT NULL,
timezone INT(10) NOT NULL,
created_at DATE NOT NULL,
created_at DATE NOT NULL,
UNIQUE INDEX `main_email_UNIQUE` (`main_email` ASC),
```

Скрипт створення таблиці «event»:

```
CREATE TABLE event (
    id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
      name VARCHAR(255) NULL DEFAULT NULL,
    user_id INT(11) NOT NULL,
      site_id INT(11) NOT NULL,
date DATE NOT NULL,
type_event INT(1) NOT NULL
    city VARCHAR(255) NULL DEFAULT NULL,
    FOREIGN KEY (user_id) REFERENCES user(id) ON DELETE CASCADE,
  FOREIGN KEY (site_id) REFERENCES site(id) ON DELETE CASCADE);
```

Цілісність бази даних. Під цілістю бази даних розуміється узгодженість (несуперечність) даних. Звичайно, СКБД не може контролювати правильність кожного окремого значення, введеного в базу даних. Цілісність баз даних може бути порушена внаслідок збою обладнання, помилок користувача або програмної помилки. В системах з багатьма користувачами цілість може бути порушена при одночасному зверненні до одного й того самого фрагмента даних.

Цілісність забезпечується шляхом задання обмежень. В залежності від джерела можна виділити інструментальні обмеження, структурні обмеження та бізнес-правила.

До інструментальних обмежень відносять перевірку правильності даних при введенні. Наприклад, поле числа не може містити текстові символи, а поле дати має містити допустиме значення дати. Середні рівні реалізації інструментальних обмежень вбудовані в СКБД.

До структурних обмеженням відносять унікальність первинного ключа та унікальність можливих ключових слів. В відношенні будь-яких двох кортежах не можуть мати одне і те ж значення атрибута (або агрегату атрибутів), об’явленої як первинний або можливий ключ. Крім того, в первинному або в можливому ключі не може бути компонент з нульовим знанням. Система управління повинна відхиляти будь-яку спробу (при введенні або оновленні) вставляти в базу даних кортеж, ключ якого або пустий (нуль), або має значення, вже введене в базу даних.

Бізнес-правила представляють собою умови, що дані відповідають-предметній області. Бізнес-правила можна розділити на елементарні та розширені. Елементарні правила обмежують значення конкретного атрибута або агрегату атрибутів через обмеження. Розширені правила виражаються у вигляді деякої залежності між атрибутами.

При проектуванні бази даних були введені такі бізнес-правила:

– в одного користувача може буде додано декілька електронних адрес і телефонів для сповіщень (при цьому є одна основна пошта, яка використовується для входу на сайт);

– тарифний план можна придбати лише на 30 днів і більше (ціна вказана за 30 календарних днів);

– користувач може додати декілька доменів для моніторингу (при цьому другий користувач має можливість також додати цей домен);

– події можуть бути двох типів (0 і 1, де 0 це подія яка вказує, що сталась якась помилка на сайті, і 1 це якщо сайт знову перейшов в робочий стан.

2.6 Діаграма прецедентів

Прецеденти (варіанти використання) – це технологія визначення функціональних вимог до системи. Робота прецедентів полягає в описі типових взаємодій між користувачами системи і самою системою і надання опису процесу її функціонування. Ключем до прецедентів є мета користувача: прецедент являє собою безліч сценаріїв, об'єднаних деякою загальною метою користувача.

Сценарій (scenario) – це послідовність кроків, що описують взаємовідносини користувача і системи.

У термінах прецеденту користувачі називаються акторами. Актор (actor) являє собою якусь роль, яку користувач грає по відношенні до системи. Актори діють в рамках прецедентів. Один актор може виконувати кілька прецедентів; і навпаки, відповідно до одним прецедентом можуть діяти кілька акторів.

Не існує стандартного способу опису вмісту прецеденту, в різних випадках застосовуються різні формати. Загальний стиль використання описаний нижче.

Вибір одного з сценаріїв в якості головного успішного сценарію (main success scenario). Спочатку потрібно описати тіло прецеденту, в якому головний успішний сценарій представлений послідовністю нумерованих кроків. Потім потрібно вибрати інший сценарій і вставити його в вигляді розширення (extension), описуючи його в термінах змін головного успішного сценарію. Розширення можуть бути успішними - користувач досяг своєї мети, або невдалими.

У кожному прецеденті є провідний актор, який посилає системі запит на обслуговування. Провідний актор – це актор, бажання якого намагається задовольнити прецедент і який зазвичай, але не завжди, є ініціатором прецеденту. Одночасно можуть бути і інші актори, з якими система також взаємодіє під час виконання прецеденту. Вони називаються другорядними акторами.

Кожен крок у прецеденті – це елемент взаємодії актора з системою. Кожен крок повинен бути простим твердженням і повинен чітко вказувати, хто виконує цей крок. Крок повинен показувати намір актора, а не механіку його дій. Отже, в прецеденті інтерфейс актора не описується.

Прецеденти є цінний інструмент для розуміння функціональних вимог до системи. Перший варіант прецедентів повинен складатися на ранній стадії виконання проекту. Більш докладні версії прецедентів повинні з'являтися безпосередньо перед реалізацією даного прецеденту. Важливо розуміти, що прецеденти представляють погляд на систему з боку. Тому, відповідності між прецедентами і класами всередині системи може і не бути. Незважаючи на те що в мові UML нічого не говориться про текст прецедентів, саме текстовий зміст прецедентів є основною цінністю цієї технології. Діаграма варіантів використання веб-додатка представлена на рисунку 2.6.

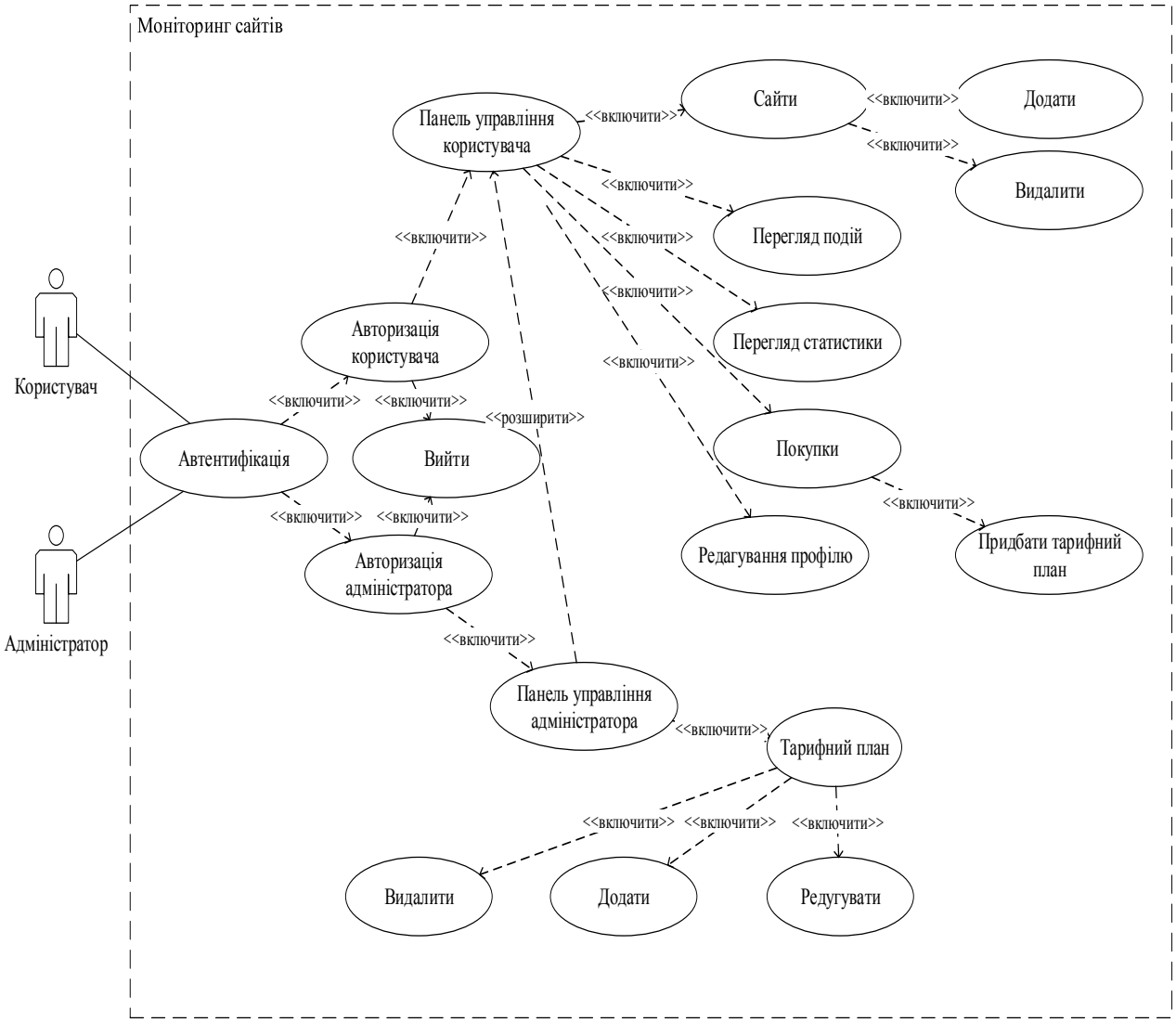


Рисунок 2.6 – Діаграма варіантів використання

В результаті аналізу поставленого завдання була виявлена роль користувача і роль адміністратора – які являються фізичною особою та працюють з даним веб додатком.

Найкраще деталізувати діаграму прецедентів за допомогою графічної таблиці (специфікації), яка б показала їх вміст (таблиця 2.4 – 2.14). Деталізація прецеденту – це точне визначення кожного прецеденту.

Таблиця 2.4 – Специфікація прецеденту «Автентифікація»

Ім’я прецеденту	Автентифікація
Ідентифікатор прецеденту	ID 1
Короткий опис	Процес розпізнавання користувача системи і надання йому певних прав та повноважень
Дійсні актори	Користувач, адміністратор
Передумова	Користувач, адміністратор повинен бути зареєстрованим в системі або пройти процедуру реєстрації
Вихідні умови	У разі успішної автентифікації виконується вхід користувача в систему, в іншому випадку виводиться помилка автентифікації.
Порядок дій	1) Натиснути кнопку «Вхід» 2) Заповнити всі обов’язкові поля (e-mail і пароль) 3) Натиснути кнопку підтвердження

Таблиця 2.5 – Специфікація прецеденту «Авторизація користувача»

Ім’я прецеденту	Авторизація користувача
Ідентифікатор прецеденту	ID 2
Короткий опис	привласнення прав користувачу на вчинення дій в системі які йому дозволені
Дійсні актори	Користувач
Передумова	Користувач повинен успішно пройти процедуру автентифікації
Вихідні умови	Попадання в систему керування
Порядок дій	-

Таблиця 2.6 – Специфікація прецеденту «Авторизація користувача»

Ім’я прецеденту	Авторизація адміністратора
Ідентифікатор прецеденту	ID 3
Короткий опис	привласнення прав адміністратора на вчинення будь-яких дій в системі
Дійсні актори	Адміністратор
Передумова	Адміністратор повинен успішно пройти процедуру автентифікації
Вихідні умови	Попадання в систему керування адміністратора
Порядок дій	-

Таблиця 2.7 – Специфікація прецеденту «Вийти»

Ім’я прецеденту	Вийти
Ідентифікатор прецеденту	ID 4
Короткий опис	Вихід с системи
Дійсні актори	Користувач, адміністратор
Передумова	Авторизований користувач або авторизований адміністратор
Вихідні умови	Вихід з веб додатку, завершення сесії, видалення всіх cookies файлів с пристрою користувача або адміністратора
Порядок дій	Натиснути кнопку «Вихід»

Таблиця 2.8 – Специфікація прецеденту «Панель управління користувача»

Ім’я прецеденту	Панель управління користувача
Ідентифікатор прецеденту	ID 5
Короткий опис	Панель, де можливе управління сайтами

Дійсні актори	Користувач
Передумова	Авторизований користувач
Вихідні умови	1. Керування сайтами 2. Перегляд подій 3. Перегляд статистики 4. Управління покупками 5. Редагування персональних даних
Порядок дій	-

Таблиця 2.9 – Специфікація прецеденту «Панель управління адміністратора»

Ім’я прецеденту	Панель управління адміністратора
Ідентифікатор прецеденту	ID 6
Короткий опис	Панель, де адміністратор може управляти тарифними планами
Дійсні актори	Адміністратор
Передумова	Авторизований адміністратор
Вихідні умови	Такі самі як у користувача і також можливість управління тарифними планами, виставлення цін
Порядок дій	–

Таблиця 2.10 – Специфікація прецеденту «Сайти»

Ім’я прецеденту	Сайти
Ідентифікатор прецеденту	ID 7
Короткий опис	Додавання, видалення сайту із моніторингу системи
Дійсні актори	Користувач, адміністратор

Передумова	Авторизація користувача і вхід в панель управління на сторінку управління сайтами
Вихідні умови	Можливість додавати, видаляти сайти із списку моніторингу, налаштування сповіщень для сайту, на які поштові адреси і телефони буде надходити сповіщення.
Порядок дій	1. Натиснути кнопку «Сайти» 2. Вибір дії 3. Зберегти

Таблиця 2.11 – Специфікація прецеденту «Перегляд подій»

Ім’я прецеденту	Перегляд подій
Ідентифікатор прецеденту	ID 8
Короткий опис	Перегляд всіх подій
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку перегляду подій
Вихідні умови	Можливість переглянути які події викликають с сайтом, збій, відновлення роботи, дата події.
Порядок дій	Натиснути кнопку «Події»

Таблиця 2.12 – Специфікація прецеденту «Перегляд статистики»

Ім’я прецеденту	Перегляд статистики
Ідентифікатор прецеденту	ID 9
Короткий опис	Перегляд статистики
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку перегляду статистики

Вихідні умови	Можливість переглянути скільки не робив сайт, коли не робив, коли був відновлений, процент uptime і downtime
Порядок дій	Натиснути кнопку «Перегляд статистики»

Таблиця 2.13 – Специфікація прецеденту «Покупки»

Ім’я прецеденту	Покупки
Ідентифікатор прецеденту	ID 10
Короткий опис	Перегляд, придбання тарифного плану
Дійсні актори	Користувач
Передумова	Авторизація користувача і вхід в панель управління на сторінку покупок
Вихідні умови	Можливість перегляду історії покупок, відображення дати придбання, строку закінчення дії тарифу, придбання підписки на тарифний план
Порядок дій	Натиснути кнопку «Покупки»

Таблиця 2.14 – Специфікація прецеденту «Редагування профілю»

Ім’я прецеденту	Редагування профілю
Ідентифікатор прецеденту	ID 11
Короткий опис	Редагування профілю користувача
Дійсні актори	Користувач, адміністратор
Передумова	Авторизація користувача і вхід в панель управління на сторінку редагування даних
Вихідні умови	Зміна персональних даних
Порядок дій	1. Змінити всі необхідні дані 2. Натиснути кнопку «Зберегти»

3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНОГО КОМПЛЕКСНОГО МЕТОДУ

Процес верифікації вимог є невід'ємною частиною всього процесу розробки. Верифікація тісно пов'язана з проектуванням, розробкою і супроводом сайту.

Верифікація – це процес переконання, що завдання було виконано в повній відповідності з вимогами, які викладені в технічному задані. Паралельно з цим також фіксується нові дефекти. Верифікація дозволяє гарантувати, що програмна система реалізована без непередбаченої функціональності, відповідає пред'явленим вимогам, специфікаціям і стандартам.

Верифікація перевіряє відповідність одних створюваних в ході розробки і супроводу ПЗ артефактів іншим, раніше створеним або використовуваним в якості вихідних даних, а також відповідність цих артефактів і процесів їх розробки правил і стандартів [11]. Зокрема, верифікація перевіряє відповідність між нормами стандартів, описом вимог (технічного завдання) до ПЗ, проектними рішеннями, вихідним кодом, призначеної для користувача документації та функціонуванням самого ПЗ. Крім того, перевіряється, що вимоги, проектні рішення, документація і код оформлені відповідно до норм і стандартів, прийнятих в даній країні, галузі та організації при розробці ПЗ, а також - що при їх створенні виконувалися всі зазначені в стандартах операції, в потрібній послідовності. Виявлені при верифікації помилки і дефекти є розбіжностями або протиріччями між декількома з перерахованих документів, між документами і реальною роботою програми, між нормами стандартів і реальним процесами розробки і супроводу ПЗ. При цьому прийняття рішення про те, який саме документ підлягає виправленню (може бути, і обидва) є окремим завданням [12].

Валідація перевіряє відповідність будь-яких створюваних або використовуваних під час розробки і супроводу ПЗ артефактів потреб і потреб користувачів і замовників цього ПЗ, з урахуванням законів предметної області і обмежень контексту використання ПЗ. Ці потреби і потреби частіше за все не зафіксовані документально - при фіксації вони перетворюються в опис вимог, один з артефактів процесу розробки ПЗ [13]. Тому валідація є менш формалізованої діяльністю, ніж верифікація. Вона завжди проводиться за участю представників замовників, користувачів, бізнес-аналітиків або експертів в предметній області - тих, чия думка можна вважати досить гарним виразом реальних потреб і потреб користувачів, замовників та інших зацікавлених осіб. Методи її виконання часто використовують специфічні техніки виявлення знань і дійсних потреб учасників.

Тестування – це процес, який полягає в перевірці відповідності програмного продукту або сайту заявленим характеристикам і вимогам.

Далі докладно описані етапи тестування сайту:

1) Починається все з підготовчих робіт – тестувальник вивчає отриману документацію (аналізує функціонал за технічним завданням, вивчає кінцеві макети сайту і становить план тесту для подальшого тестування).

2) Функціональне тестування - найбільш тривалий етап перевірки ресурсу. Суть цього процесу полягає в перевірці всього описаного функціоналу:

– перевірки роботи всіх обов'язкових функцій сайту;

– тестування працездатності призначених для користувача форм на сайті;

– перевірки роботи пошуку (включаючи релевантність результатів);

– перевірки гіперпосилань, пошук неробочих посилань;

– перевірки завантаження файлів на сервер;

– перевірки працездатності лічильників на сторінках сайту;

– перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданим замовником.

3) Тестування верстки - при перевірці верстки насамперед тестувальник перевіряє розташування елементів, відповідність їх позицій надавати пріоритет-ленним макетів, а також перевіряє оптимізацію зображень і графіки. Так-леї здійснюється перевірка валідності коду. У процесі верстки важливо зберегти коректну ієрархію об'єктів, і важливо упевнитися в її валідності за фактом завершення робіт.

4) Usability тестування – проводиться для оцінки зручності продукту у використанні, заснований на залученні користувачів в якості тестувальників і аналіз отриманих результатів.

Незважаючи на той факт, що опрацювання зручності використання ресурсу здійснюється в процесі складання технічного завдання, розробки макетів, бувають ситуації, коли отриманий результат не є оптимальним ним. Хоча таке і відбувається досить рідко, оптимальне рішення в даному випадку – виконати зміни в реалізований товар.

5) Тестування безпеки – на даній стадії тестування спеціаліст перевіряє – чи немає у користувачів доступу до службових / закритих сторінок, а також проводить перевірку захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу.

6) Тестування продуктивності сайту – проводиться з метою визначення швидкодії сайту або його частини під певним навантаженням. Тестування продуктивності включає в себе такі види тестування:

– тестування навантаження – найпростіша форма тестування продуктивності. Тестування навантаження зазвичай проводиться для того, щоб оцінити поведінку сайту (або додатки) під заданої очікуваної навантаженням. Цією навантаженням може бути, наприклад, очікувана кількість одночасно працюючих користувачів на сайті, що здійснюють вказану кількість транзакцій за інтервал часу. Такий тип тестування зазвичай дозволяє отримати час відгуку всіх найважливіших бізнес-функцій;

– тестування швидкодії – перевірка швидкості завантаження сайту для визначення швидкості відпрацювання скриптів, завантаження зображень і контенту. Цей тест проводиться з метою оптимізації процесу завантаження сайту, а також визначення оптимальності налаштувань сервера.

В даному розділі будемо використовувати функціональне тестування веб-додатку.

Завданням функціонального тестування є перевірка відповідності програми своїм специфікаціям. При даному підході текст програми не доступний, і програма розглядається як "чорної скриньки". Найпоширенішими видами функціонального тестування є методи випадкового тестування, еквівалентної розбивки й аналізу граничних умов.

Функціональне тестування призначене для перевірки відповідності програмного забезпечення його специфікації (завданню) або еталону, тобто перевірки, чи виконує програмна система або її модулі відповідні функції та поставлені вимоги. При цьому тестувальник перевіряє виконувані функції, а не реалізацію програмного забезпечення шляхом аналізу вхідних і вихідних даних. У науковій літературі цей метод називають ще методом тестування "чорної скриньки". Для реалізації цього методу повинні бути відомі функції програм програмного забезпечення. Результати функціонального тестування дають відповіді на запитання: як виконуються функції програм програмного забезпечення; як сприймаються вхідні дані;

Функціональне тестування застосовується тільки коли програмне забезпечення вже створене, тобто на останніх етапах життєвого циклу програмного забезпечення. Якщо функціональне тестування виявить погану якість створення програмного забезпечення, то доводиться повертатися на попередні етапи розробки, що спричиняє як фінансові збитки так і часові витрати.

Тестування в перспективі «вимоги» використовує специфікацію функціональних вимог до системи як основу для дизайну тестових випадків (Test Cases). У цьому випадку необхідно зробити список того, що буде тестуватися, а що ні, пріоритезувати вимоги на основі ризиків (якщо це не зроблено в документі з вимогами), а на основі цього пріоритезувати тестові сценарії (test cases). Це дозволить сфокусуватися і не упустити при тестуванні найбільш важливий функціонал. Тестові випадки представлені в таблицях 3.1 – 3.8.

Таблиця 3.1 – Тестовий випадок №1

Ідентифікатор тестового випадку		TC-1 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість реєстрації нового користувача і авторизацію користувача	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту (Рисунок 3.1)	+
2	Користувач натискає кнопку «Реєстрація»	З'являється модальне вікно, для заповнення полів для реєстрації	+
3	Користувач заповнює всі поля і натискає кнопку «Реєстрація»	Поля заповнені	+
4	Отримано повідомлення для	З’являється вікно з повідомленням “На Ваш е-	+

	підвердження реєстрації	mail відправлено підтвердження реєстрації”	

Продовження таблиці 3.1 – Тестовий випадок №1

Крок	Дія	Очікуваний результат	Відмітка про виконання
5	Користувач переходить по гіперпосиланню в письмі	З’являється вікно з повідомленням “Ваш e-mail підтверджений! Ви можете увійти у свій особистий кабінет ”	
5	Користувач натискає “Вхід”	Користувач натискає кнопку “Вхід”	+
6	Користувач заповнює всі поля	Поля заповнені	+
7	Користувач натискає кнопку “Увійти”	Відображається особистий кабінет користувача (Рисунок 3.2)	+
Результати тесту: тест виконаний успішно			

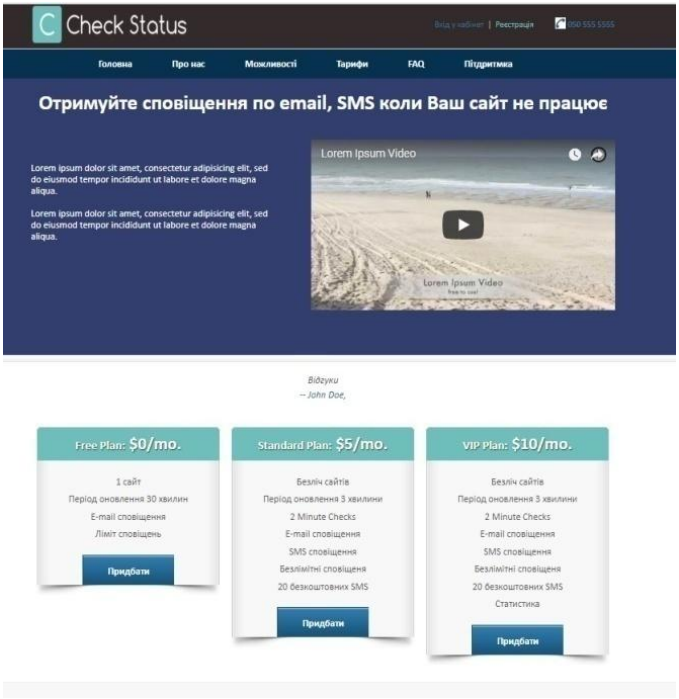


Рисунок 3.1 – Головна сторінка сайту (продукт для потенційного клієнта, не особистий кабінет користувача)

Таблиця 3.2 – Тестовий випадок №2

Ідентифікатор тестового випадку		TC-2 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість виходу із системи	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в	Користувач увійшов в	+

	систему	систему	
3	Користувач натискає на кнопку «Вихід»	Користувач вийшов з системи	+
Результати тесту: тест виконаний успішно			

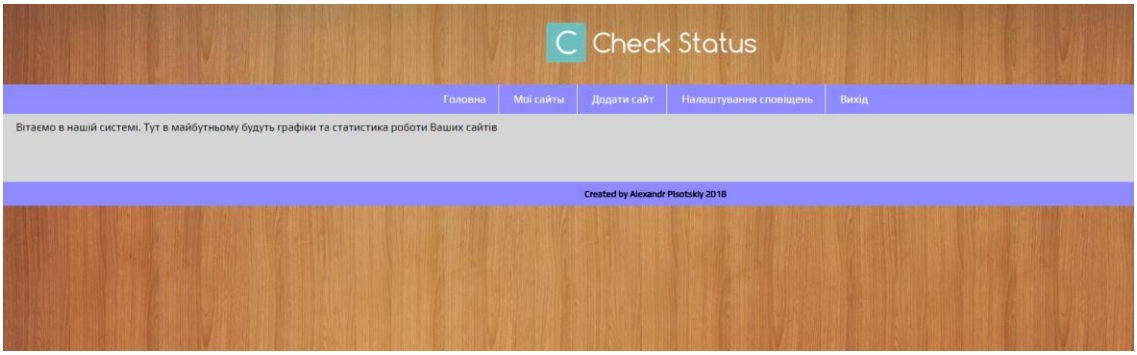


Рисунок 3.2 – Особистий кабінет користувача

Таблиця 3.3 – Тестовий випадок №3

Ідентифікатор тестового випадку		TC-3 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити можливість додати сайт в систему	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в	Користувач увійшов в	+

	систему	систему	
3	Користувач натискає на кнопку «Додати сайт»	Потрапляє на сторінку додавання сайту	+
4	Користувач вписує домен сайту у спеціальне поле для вводу і натискає кнопку «Додати»	Якщо домен коректний, відображається модальне вікно про успішне додавання домену(Рисунок 3.3) , якщо це не домен, то буде помилка, що дані введені невірні (Рисунок 3.4)	+
Результати тесту: тест виконаний успішно			

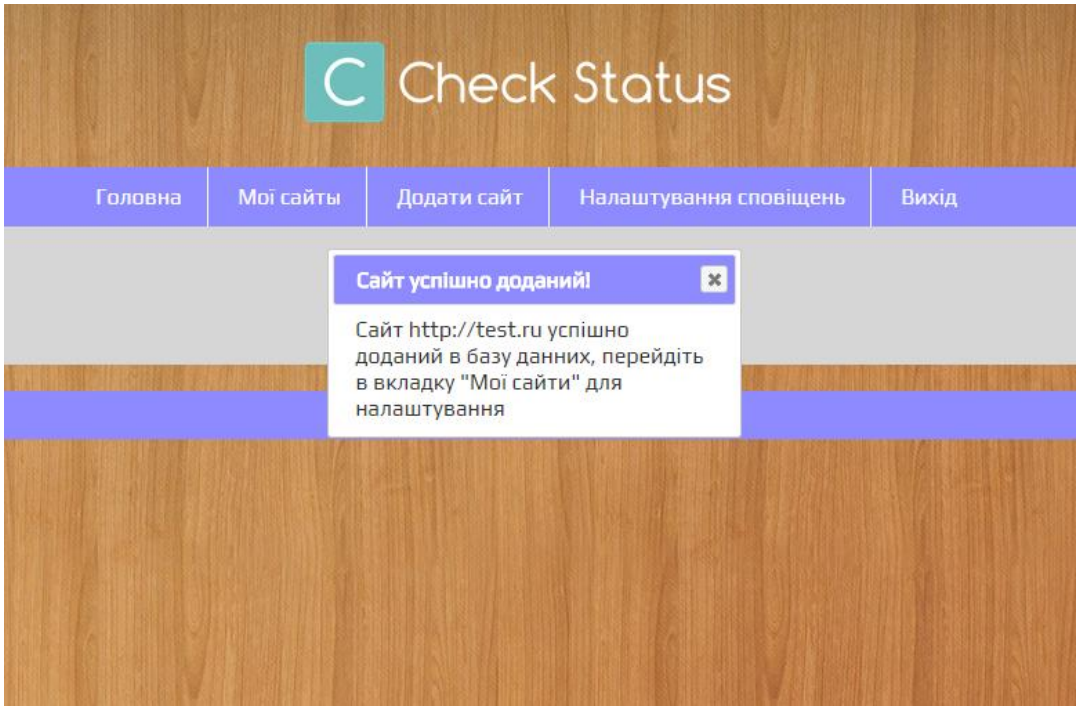


Рисунок 3.3 – Сповіщення що сайт доданий до системи

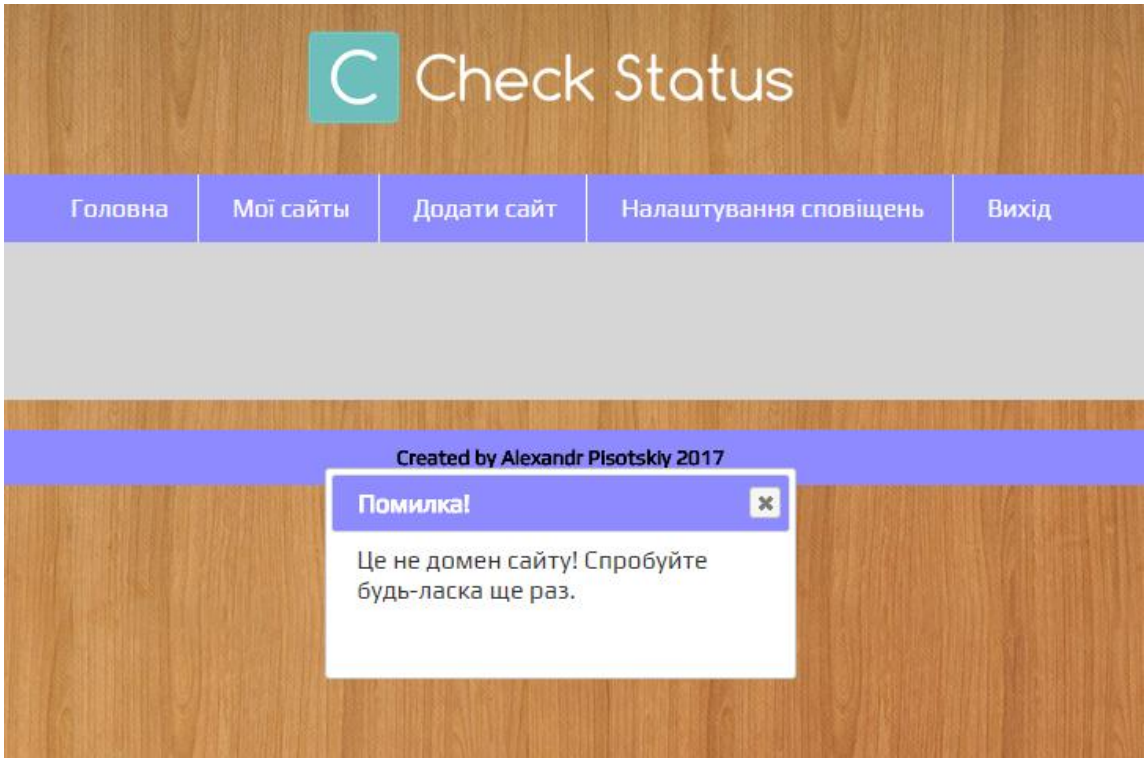


Рисунок 3.4 – Помилка додавання сайту

Таблиця 3.4 – Тестовий випадок №4

Ідентифікатор тестового випадку		ТС-4 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити які сайти додані користувачем до системи моніторингу	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в систему	Користувач увійшов в систему	+
3	Користувач натискає вкладку «Мої сайти»	Потрапляє на сторінку сайтів які були додані раніше до системи (Рисунок 3.5)	+

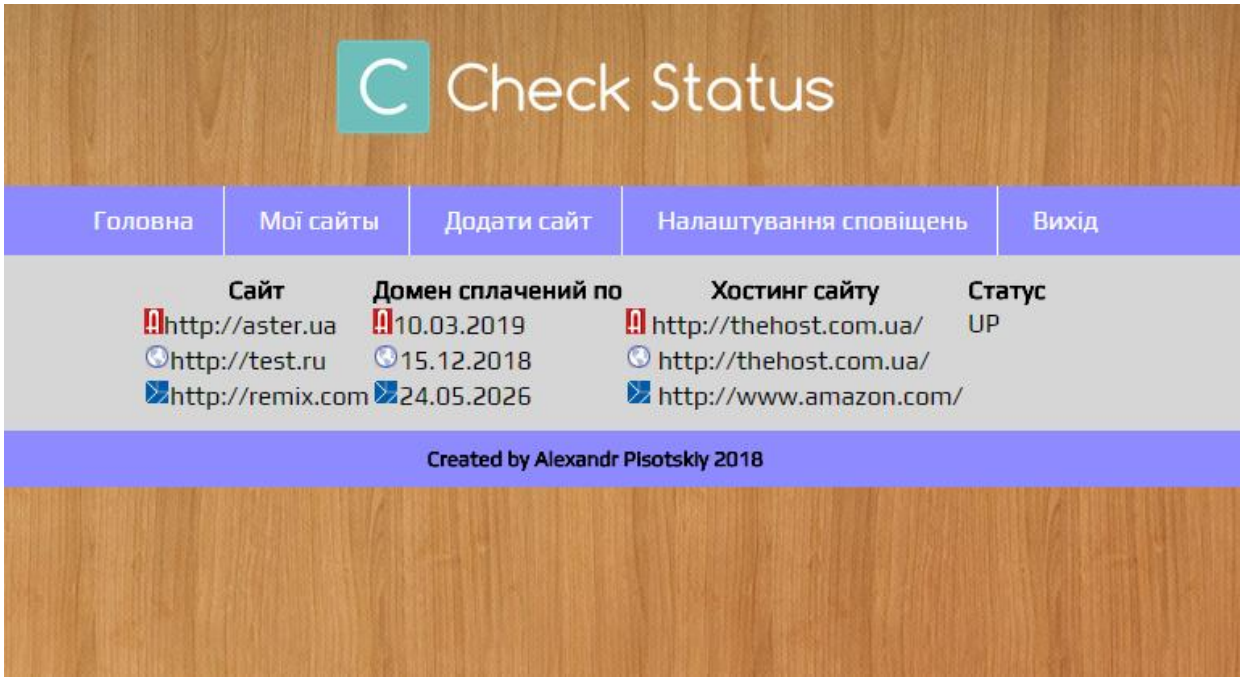


Рисунок 3.5 – Сайти користувача для моніторингу

Таблиця 3.5 – Тестовий випадок №5

Ідентифікатор тестового випадку		TC-5 ver 1.0	
Взаємозалежність тестових випадків		Відсутнє	
Мета тесту		Перевірити налаштування для сайту та збереження цих налаштувань в базі даних	
Методика тестування			
Крок	Дія	Очікуваний результат	Відмітка про виконання
1	Перейти на веб-сайт	Відображається головне вікно сайту	+
2	Користувач входить в систему	Користувач увійшов в систему	+

3	Користувач натискає вкладку «Мої сайти»	Потрапляє на сторінку сайтів які були додані раніше	+
4	Вибирає який сайт треба налаштувати	Потрапляє на сторінку налаштувань сайту (Рисунок 3.6)	+
5	Оновлює необхідні дані та натискає кнопку «Зберегти»	Отримує повідомлення про успішне зберігання налаштувань	+

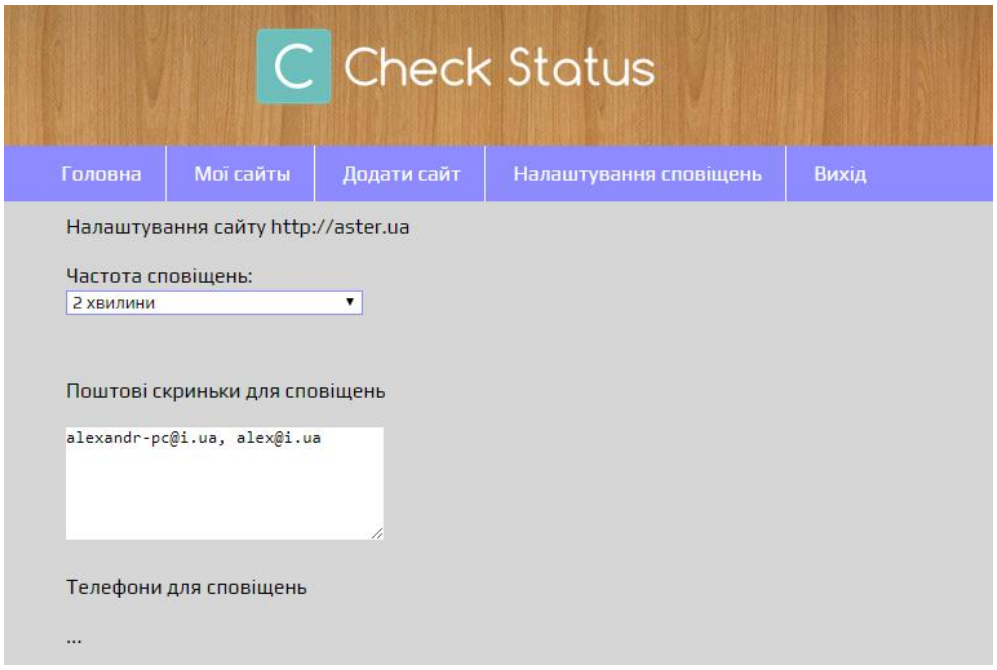


Рисунок 3.6 – Сторінка налаштувань для домену

У клієнтських системах брандмауер є надзвичайно корисним інструментом для захисту даних. Брандмауер можна використовувати як програмне забезпечення або обладнання для створення бар'єру між клієнтською мережею та трафіком з Інтернету. Аналізуючи вхідні трафік на основі правил попереднього налаштування, брандмауери можуть фільтрувати трафік, щоб блокувати зловмисний трафік, наприклад шкідливе програмне забезпечення і запобігти доступу хакерів до точки входу. Крім того, антивірусне програмне забезпечення також є ефективним рішенням питання конфіденційності та безпеки. Антивірусне програмне забезпечення може виявляти та автоматично видаляти віруси та шкідливі програми такі як «троянські коні», «хробаки», програми-шпигуни та програми-вимагачі. Скануючи клієнтську мережу в реальному часі, він може запобігти можливій уразливості. Окрім завантаження антивірусного програмного забезпечення та брандмауера, підтримка програмного забезпечення в актуальному стані також є життєво важливою справою. Зараз хакери розробляють складні програми, й важливо оновити програмне забезпечення для боротьби з ними.

У серверних системах віртуальна приватна мережа (VPN) є ефективним інструментом для онлайн-бізнесу. Він здатний зміцнювати безпеки шляхом захисту та шифрування даних. Він також забезпечує приватний перегляд і необмежений доступ до вмісту. За застосовуючи VPN, трафік, який проходить через протокол Інтернет-сервісу (ISP), буде зашифровано протоколами VPN.

Тим часом зловмисники не можуть переглядати та контролювати їх. Хоча в клієнт-серверному рішенні, рішення на основі шифрування шифруватиме інформацію між веб-хостами. Pretty Good Privacy (PGP) — це протокол на основі шифрування, який визначає, як шифрувати та дешифрувати тексти, файли з електронних листів. PGP використовує симетричний і асиметричні ключі для шифрування даних. Тим часом, відкритий ключ буде наданий

одержувачем тому, хто хоче щоб надіслати повідомлення. Тоді як одержувачам, які мають закритий ключ, можна дозволити лише розшифрувати повідомлення.

Більше того, розумне мислення також є набагато важливішим, оскільки це може запобігти крадіжці особистих даних, яка набуває особистого характеру інформація без підтвердження користувачів. Потрібно усвідомлювати і не довіряти людям, які є невідомі та ділитися особистою інформацією в облікових записах соціальних мереж. Немає такого поняття, як безкоштовний обід, подумайте розумний і оцінюючи веб-сайт, який стверджує, що пропонує переваги. Крім того, посиліть міцність точки доступу щоб уникнути проблем безпеки MITM. Використання шифрування WPA2 разом із алгоритмом AES може запобігти грубій обробці силові атаки.

Також важливо змінити назву пристрою та пароль точки доступу, оскільки зловмисники можуть змінити їх сервер і, на жаль, навіть знищити систему.

ВИСНОВКИ

На підставі результатів попередніх розділів можна зробити висновок про важливість конфіденційності та посилення безпеки і вдосконалення бездротових мереж.

Під час дослідження встановлено, що проблеми з безпекою та конфіденційністю бездротових мереж постійно актуальні та зростають. Зазначена проблема більшістю суспільства сприймається не серйозно. Суспільство продовжує рухатися до епохи технологій, яка постійно змінюється, особливо, коли конфіденційність і безпека пристроїв і мереж потрібні як ніколи. Дії та зусилля по забезпеченню захисту інформації повинні посилюватись, оскільки кібер-зловмисників стає все більше.

У роботі досліджені існуючи сервіси моніторингу працездатності сайтів користувачів, основні функції та категорії вибору сервісу, встановлені вимоги до моніторингу працездатності сайтів, етапи технології розробки сервісу моніторингу працездатності сайтів користувачів, обґрунтовано вибір локального сервісу, мови програмування, фреймворку та середовища програмування, проведено функціональне тестування веб-додатку.

Проблеми безпеки все ще виникають у кіберсвіті, і їхня кількість продовжує зростати через брак організаційних і технічних рішень та ігнорування населенням щодо цих поточних питань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Технології для бизнесу / www.itfb.com.ua
2) Official Ping-admin Database Site. URL: https://ping-admin.ru/
3) Official Host-tracker Database Site. URL: https://www.host-tracker.com/ua
4) Official Monitis Database Site. URL: https://www.monitis.com/
5) Official Site24x7Database Site. URL: https://www.site24x7.com/
6) Official Uptimerobot Database Site. URL: https://uptimerobot.com/
7) https://blogwork.ru/chto-takoe-lokalnyj-server/
8) Official Denwer Database Site. URL: http://www.denwer.ru/
9) Official Winginx Database Site. URL: https://winginx.com/ru/
10) Лаврищева Є.М., Грищенко В.М. / Складальне програмування. Основи промисловості програмних товарів / 2-вид. Доповнене та перероблене. - Київ: Наук. думка, 2009.– 372с.
11) Шнайдер Р. / Microsoft SQL Server 6.5. Проектування високопродуктивних баз даних. – К.: Лорі, 2010. – 361 с.
12) Official MySQL Database Site. URL: https://www.mysql.com/
13) Official Oracle Database Site. URL: https://www.oracle.com/index.html
14) Грофф Дж., Вайнберг П. SQL. К: BHV, 2005. – 608 с.
15) Official Jetbrains Database Site. URL: https://www.jetbrains.com/ru-ru/phpstorm/
16) Шубінский І.Б. Функціональна надійність інформаційних систем. Методи аналізу – К.: «Журнал Надійність», 2012, – 296 с.
17) IEEE 1012-2004 Standard for Software Verification and Validation. IEEE, 2005.
18) ISO/IEC 12207 Systems and software engineering - Software life cycle processes. Geneva, Switzerland: ISO, 2008.

19) S.J., Hussain, M., Irfan, N.Z., Jhanjhi, et al. (2020). Performance Enhancement in Wireless Body Area Networks. with Secure Communication. Wireless Pers Commun (2020). https://doi.org/10.1007/s11277-020-07702-7

20) K. Ramesh Rao, "Wireless Communication Security and Privacy issues and Challenges", Academia.edu, 2017. [Online]. Available: https://www.academia.edu/34148630/Wireless_Communication_Security_and_Privacy_issues_and_Challenges. [Accessed: 04- Jul- 2020].

21) Alferidah, D.K. and Jhanjhi, N.Z. (2020). A Review on Security and Privacy Issues and Challenges in Internet of Things. IJCSNS International Journal of Computer Science and Network Security, VOL.20 No.4, April 2020.

22) M. Almulhim and N. Zaman, "Proposing secure and lightweight authentication scheme for IoT based E-health applications," *2018 20th International Conference on Advanced Communication Technology (ICACT)*, Chuncheon-si Gangwon-do, Korea (South), 2018, pp. 481-487, doi: 10.23919/ICACT.2018.8323802.

23) K. Hussain, S. J. Hussain, N. Jhanjhi and M. Humayun, "SYN Flood Attack Detection based on Bayes Estimator (SFADBE) For MANET," *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2019, pp. 1-4, doi: 10.1109/ICCISci.2019.8716416.

24) B. Franklin, "Wireless Networking Security", Cs.bham.ac.uk, 2007. [Online]. Available: https://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS7/Wireless%20Networking%20Security. htm. [Accessed: 04- Jul- 2020].

25) Seungjin, L., Abdullah, A. Jhanjhi, N.Z. (2020). A Review on Honeypot-based Botnet Detection Models for Smart Factory. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 6, 2020.

ДОДАТКИ

PRACTICAL APPLICATION OF THE DEVELOPED COMPLEX METHOD

The requirements verification process is an integral part of the entire development process. Verification is closely related to the design, development and maintenance of the site.

Verification is the process of making sure that the task has been performed in full compliance with the requirements set out in the specifications. In parallel with this, new defects are also recorded. Verification allows you to guarantee that the software system is implemented without unforeseen functionality, meets the requirements, specifications and standards.

Verification verifies the compliance of some artifacts created during the development and maintenance of software with others previously created or used as source data, as well as the compliance of these artifacts and their development processes with rules and standards [11]. In particular, the verification checks the conformity between the norms of the standards, the description of the requirements (technical task) for the software, the design decisions, the source code, the documentation intended for the user and the functioning of the software itself. In addition, it is checked that the requirements, design solutions, documentation and code are designed in accordance with the norms and standards adopted in the given country, industry and organization during the development of the software, as well as - that during their creation, all the operations specified in the standards were performed in the required sequence . Errors and defects detected during verification are discrepancies or contradictions between several of the listed documents, between documents and the actual work of the program, between the norms of the standards and the actual processes of software development and maintenance. At the same time, deciding which document is to be corrected (maybe both) is a separate task [12].

Validation verifies the compliance of any artifacts created or used during the development and maintenance of the software with the needs and requirements of the users and customers of this software, taking into account the laws of the subject area and the limitations of the context of use of the software. These needs and requirements are most often not recorded in documents - when recorded, they turn into a description of requirements, one of the artifacts of the software development process [13]. Therefore, validation is a less formalized activity than verification. It is always conducted with the participation of representatives of customers, users, business analysts or experts in the subject area - those whose opinion can be considered a fairly good expression of the real needs and wants of users, customers and other interested parties. The methods of its implementation often use specific techniques for identifying the knowledge and real needs of the participants.

Testing is a process that consists in checking the compliance of a software product or website with the stated characteristics and requirements.

Next, the stages of site testing are described in detail:

1) Everything begins with preparatory work - the tester studies the received documentation (analyzes the functionality according to the technical task, studies the final layouts of the site and makes a test plan for further testing).

2) Functional testing is the longest stage of resource verification. The essence of this process is to check all the described functionality:

- checking the operation of all mandatory functions of the site;

– testing the functionality of the forms intended for the user on the site;

– checks of search performance (including relevance of results);

– checking hyperlinks, searching for non-working links;

– checks for uploading files to the server;

- checking the functionality of the counters on the website pages;

- review for compliance of the content of the site pages of the original content provided by the customer.

3) Layout testing - when checking the layout, first of all, the tester checks the location of the elements, the compliance of their positions with priority layouts, and also checks the optimization of images and graphics. The validity of the code is then checked. In the layout process, it is important to maintain the correct hierarchy of objects, and it is important to make sure of its validity upon completion of the work.

4) Usability testing – conducted to assess the ease of use of the product, based on the involvement of users as testers and the analysis of the results obtained.

Despite the fact that the study of the ease of use of the resource is carried out in the process of drawing up the technical task, developing layouts, there are situations when the obtained result is not optimal. Although this happens quite rarely, the optimal solution in this case is to make changes to the sold product.

5) Security testing - at this stage of testing, the specialist checks whether users do not have access to service / closed pages, and also checks the protection of all critical pages (for example, the site administration section) from external influences.

6) Site performance testing – performed to determine the speed of the site or its part under a certain load. Performance testing includes the following types of testing:
– load testing is the simplest form of performance testing. Load testing is usually done in order to evaluate the behavior of a site (or application) under a given expected load. This load can be, for example, the expected number of simultaneously working users on the site, making the specified number of transactions per time interval. This type of testing usually allows you to get the response time of all the most important business functions; - speed testing - checking the site's loading speed to determine the speed of execution of scripts, loading of images and content. This test is conducted in order to optimize the process of loading the site, as well as to determine the optimality of the server settings. In this section, we will use functional testing of a web application. The task of functional testing is to verify the compliance of the program with its specifications. With this approach, the text of the program is not available, and the program is considered as a "black box". The most common types of functional testing

are methods of random testing, equivalent breakdown and analysis of boundary conditions. Functional testing is intended to verify the compliance of the software with its specification (task) or standard, i.e. checking whether the software system or its modules perform the relevant functions and set requirements. At the same time, the tester checks the functions performed, and not the implementation of the software, by analyzing the input and output data. In the scientific literature, this method is also called the "black box" testing method. To implement this method, the functions of software programs must be known. The results of functional testing provide answers to the following questions: how the functions of software programs are performed; how input data is perceived; Functional testing is applied only when the software has already been created, that is, at the last stages of the software life cycle. If functional testing reveals a poor quality of software creation, then it is necessary to return to the previous stages of development, which causes both financial losses and time costs. Testing in the "requirements" perspective uses the specification of functional requirements for the system as a basis for the design of test cases (Test Cases). In this case, it is necessary to make a list of what will be tested and what will not be tested, prioritize requirements based on risks (if this is not done in the requirements document), and based on this, prioritize test scenarios (test cases). This will allow you to focus and not miss the most important functionality during testing. Test cases are presented in tables 3.1 - 3.8. On client systems, a firewall is an extremely useful tool for protecting data. A firewall can be used as software or hardware to create a barrier between the client network and traffic from the Internet. By analyzing incoming traffic based on pre-configured rules, firewalls can filter traffic to block malicious traffic such as malware and prevent hackers from accessing the entry point. In addition, antivirus software is also an effective solution to the issue of privacy and security. Antivirus software can detect and automatically remove viruses and malware such as Trojan horses, worms, spyware, and ransomware. By scanning the client network in real-time, it can prevent potential vulnerabilities. In addition to

downloading antivirus software and a firewall, keeping your software up-to-date is also vital. Hackers are now developing sophisticated programs, and it's important to update your software to combat them. In server systems, Virtual Private Network (VPN) is an effective tool for online business. It is able to strengthen security by protecting and encrypting data. It also provides private browsing and unlimited content access. By using a VPN, the traffic that goes through the Internet Service Protocol (ISP) will be encrypted by the VPN protocols. Meanwhile, attackers cannot view and control them. While in a client-server solution, an encryption-based solution will encrypt information between web hosts. Pretty Good Privacy (PGP) is an encryption-based protocol that defines how to encrypt and decrypt text, files from emails. PGP uses symmetric and asymmetric keys to encrypt data. Meanwhile, the public key will be provided by the receiver to whoever wants to send the message. Whereas recipients who have the private key can only be allowed to decrypt the message. Moreover, smart thinking is also much more important because it can prevent identity theft, which takes the form of personal information without users' authentication. You need to be aware and not trust people who are unknown and share personal information on social media accounts. There is no such thing as a free lunch, think smart and appreciate a website that claims to offer benefits. Also, strengthen access point strength to avoid MITM security issues. Using WPA2 encryption together with the AES algorithm can prevent a brute force attack.

It is also important to change the name of the device and the password of the access point, because attackers can change their server and, unfortunately, even destroy the system.